

Inria

RESEARCH CENTER

FIELD

**Networks, Systems and Services,
Distributed Computing**

Activity Report 2019

Section Software

Edition: 2020-03-21

DISTRIBUTED SYSTEMS AND MIDDLEWARE

1. COAST Project-Team 5
2. CTRL-A Project-Team 6
3. DELYS Project-Team (section vide) 7
4. MIMOVE Project-Team 8
5. Myriads Project-Team 10
6. SPIRALS Project-Team 12
7. STACK Project-Team 14
8. WHISPER Project-Team 18
9. WIDE Project-Team 19

DISTRIBUTED AND HIGH PERFORMANCE COMPUTING

10. ALPINES Project-Team 21
11. AVALON Project-Team 23
12. DATAMOVE Project-Team 27
13. HIEPACS Project-Team 29
14. KERDATA Project-Team 36
15. POLARIS Project-Team 39
16. ROMA Project-Team 42
17. STORM Project-Team 43
18. TADAAM Project-Team 50

DISTRIBUTED PROGRAMMING AND SOFTWARE ENGINEERING

19. DIVERSE Project-Team 55
20. EASE Project-Team 60
21. FOCUS Project-Team 63
22. INDES Project-Team 67
23. RMOD Project-Team 70

NETWORKS AND TELECOMMUNICATIONS

24. AGORA Project-Team 71
25. COATI Project-Team 73
26. DANTE Project-Team 75
27. DIANA Project-Team 77
28. DIONYSOS Project-Team 81
29. DYOGENE Project-Team 83
30. EVA Project-Team 84
31. FUN Project-Team 86
32. GANG Project-Team 88
33. GANG Project-Team 89
34. MARACAS Team 90
35. NEO Project-Team 92
36. RESIST Team 93
37. SOCRATE Project-Team 95

38. TRIBE Project-Team 97

COAST Project-Team

6. New Software and Platforms

6.1. MUTE

Multi-User Text Editor

KEYWORD: Collaborative systems

SCIENTIFIC DESCRIPTION: MUTE is a peer 2 peer collaborative editing platform that is used to evaluate replication algorithms in editing situations regarding their performances and to understand how it affects user experience.

FUNCTIONAL DESCRIPTION: Existing collaborative systems generally rely on a service provider that stores and has control over user data which is a threat for privacy. MUTE (Multi-User Text Editor) is a web-based real-time collaborative editor that overcomes this limitation by using a peer-to-peer architecture relying on WebRTC. Several users may edit in real-time a shared document and their modifications are immediately sent to the other users without transiting through a central server. Our editor offers support for working offline while still being able to reconnect at a later time, which gives it a unique feature. Data synchronisation is achieved by using the LogootSplit algorithm developed by team Coast.

NEWS OF THE YEAR: In 2019 we implemented a new algorithm, dotted logoot-split. We integrated a group key management algorithm to evaluate a secure version of the algorithm in dynamic situation. We also incorporated probes to evaluate collaboration situation.

- Participants: Claudia Ignat, François Charoy, Gérald Oster, Luc André, Matthieu Nicolas and Victorien Elvinger
- Contact: Gérald Oster
- URL: <https://github.com/coast-team/mute>

CTRL-A Project-Team

6. New Software and Platforms

6.1. Heptagon

KEYWORDS: Compilers - Synchronous Language - Controller synthesis

FUNCTIONAL DESCRIPTION: Heptagon is an experimental language for the implementation of embedded real-time reactive systems. It is developed inside the Synchronics large-scale initiative, in collaboration with Inria Rhones-Alpes. It is essentially a subset of Lucid Synchrone, without type inference, type polymorphism and higher-order. It is thus a Lustre-like language extended with hierarchical automata in a form very close to SCADE 6. The intention for making this new language and compiler is to develop new aggressive optimization techniques for sequential C code and compilation methods for generating parallel code for different platforms. This explains much of the simplifications we have made in order to ease the development of compilation techniques.

The current version of the compiler includes the following features: - Inclusion of discrete controller synthesis within the compilation: the language is equipped with a behavioral contract mechanisms, where assumptions can be described, as well as an "enforce" property part. The semantics of this latter is that the property should be enforced by controlling the behaviour of the node equipped with the contract. This property will be enforced by an automatically built controller, which will act on free controllable variables given by the programmer. This extension has been named BZR in previous works. - Expression and compilation of array values with modular memory optimization. The language allows the expression and operations on arrays (access, modification, iterators). With the use of location annotations, the programmer can avoid unnecessary array copies.

- Participants: Adrien Guatto, Brice Gelineau, Cédric Pasteur, Eric Rutten, Gwenaël Delaval, Léonard Gérard and Marc Pouzet
- Partners: UGA - ENS Paris - Inria - LIG
- Contact: Gwenaël Delaval
- URL: <http://heptagon.gforge.inria.fr>

DELYS Project-Team (section vide)

MIMOVE Project-Team

6. New Software and Platforms

6.1. SocialBus

Universal Social Network Bus

KEYWORDS: Middleware - Interoperability - Social networks - Software Oriented Service (SOA)

FUNCTIONAL DESCRIPTION: Online social network services (OSNSs) have become an integral part of our daily lives. At the same time, the aggressive market competition has led to the emergence of multiple competing siloed OSNSs that cannot interoperate. As a consequence, people face the burden of creating and managing multiple OSNS accounts and learning how to use them, to stay connected. The goal of the Universal Social Network Bus (USNB) is to relieve users from such a burden, letting them use their favorite applications to communicate.

- Authors: Rafael Angarita Arocha, Nikolaos Georgantas and Valérie Issarny
- Contact: Valérie Issarny
- URL: <https://gitlab.inria.fr/usnb/universal-social-network-bus>

6.2. WeBrowse

KEYWORDS: Web Usage Mining - Content analysis - Recommendation systems

FUNCTIONAL DESCRIPTION: The amount of information available on the web today, and the fast rate with which new information appears, overwhelm most users. The goal of our research is to assist Web users in discovering content. One of the most powerful means today to help people discover new web content is sharing between members of online communities. In the case of communities of a place (e.g., people who live, study, or work together) people share common interests, but often fail to actively share content. To address this problem, we have developed WeBrowse, a passive crowdsourced content discovery system for communities of a place.

WeBrowse leverages the passive observation of web-clicks (i.e., the URLs users intentionally visit) as an indication of users' interest in a piece of content. Intuitively, the more users click on a URL, the higher the interest in the content on the corresponding page. Our approach is then to leverage the collective clicks in a community to automatically discover relevant content to promote to users of the community.

To implement passive crowdsourcing, one must be in a position to observe the aggregated web-clicks of the community. Luckily, in many communities of a place, users will connect to the Internet from the same network, such as, e.g., the campus/enterprise network or the network of a residential Internet Service Provider (ISP) in a neighborhood. WeBrowse (i) observes web packets flowing through a network link, (ii) passively extracts HTTP logs (i.e., streams recording the headers of HTTP requests), and (iii) detects and decides on-the-fly the set of URLs to show to users.

- Contact: Renata Cruz Teixeira
- URL: <https://team.inria.fr/muse/webbrowse-info-page/>

6.3. VSB

eVolution Service Bus

KEYWORDS: Service and Thing choreographies - Middleware protocol interoperability - Enterprise service bus

FUNCTIONAL DESCRIPTION: VSB is a development and runtime environment dedicated to complex distributed applications of the Future Internet. Such applications are open, dynamic choreographies of extremely heterogeneous services and Things, including lightweight embedded systems (e.g., sensors, actuators and networks of them), mobile systems (e.g., smartphone applications), and resource-rich IT systems (e.g., systems hosted on enterprise servers and Cloud infrastructures). VSB's objective is to seamlessly interconnect, inside choreographies, services and Things that employ heterogeneous interaction protocols at the middleware level, e.g., SOAP Web services, REST Web services, Things using CoAP. This is based on runtime conversions between such protocols, with respect to their primitives and data type systems, while properly mapping between their semantics. This also includes mapping between the public interfaces of services/Things, regarding their operations and data, from the viewpoint of the middleware: the latter means that operations and data are converted based on their middleware-level semantics, while their business semantics remains transparent to the conversion. VSB follows the well-known Enterprise Service Bus (ESB) paradigm. We propose a generic interface description, which we call GIDL, for application components that employ VSB. Based on GIDL, we enable automated synthesis of binding components for connecting heterogeneous services and Things onto VSB.

- Participants: Georgios Bouloukakis, Nikolaos Georgantas and Patient Ntumba
- Contact: Nikolaos Georgantas
- URL: <https://gitlab.ow2.org/chorevolution/evolution-service-bus>

6.4. Service traceroute

KEYWORDS: Network monitoring - Network diagnosis

FUNCTIONAL DESCRIPTION: Traceroute is often used to help diagnose when users experience issues with Internet applications or services. Unfortunately, probes issued by classic traceroute tools differ from application traffic and hence can be treated differently by middleboxes within the network. We propose a new traceroute tool, called Service traceroute. Service traceroute leverages the idea from paratrace, which passively listens to application traffic to then issue traceroute probes that pretend to be part of the application flow. We extend this idea to work for modern Internet services with support for identifying the flows to probe automatically, for tracing of multiple concurrent flows, and for UDP flows. We implement command-line and library versions of Service traceroute, which we release as open source.

- Partner: Princeton University
- Contact: Renata Cruz Teixeira
- URL: <https://github.com/wontoniii/service-traceroute>

6.5. Network Microscope

KEYWORDS: Quality of Experience - Network monitoring - Video analysis

FUNCTIONAL DESCRIPTION: A system that accurately infers video streaming quality metrics in real time, such as startup delay or video resolution, by using just a handful of features extracted from passive traffic measurement. Network Microscope passively collects a corpus of network features about the traffic flows of interest in the network and directs those to a real-time analytics framework that can perform more complex inference tasks. Network Microscope enables network operators to determine degradations in application quality as they happen, even when the traffic is encrypted.

- Participants: Francesco Bronzino and Renata Cruz Teixeira
- Contact: Renata Cruz Teixeira
- URL: <https://netmicroscope.com/>

6.6. HostView Mobile

KEYWORDS: Quality of Experience - Network monitoring

FUNCTIONAL DESCRIPTION: HostView for mobile runs on Android devices to monitor user system and network performance together with user feedback on Internet experience.

- Contact: Giulio Grassi

Myriads Project-Team

6. New Software and Platforms

6.1. PaaSage Adapter

KEYWORDS: Cloud computing - Dynamic adaptation - Cloud applications management

FUNCTIONAL DESCRIPTION: The purpose of the Adapter is to transform the current configuration of a cloud application into a target configuration in an efficient and safe way. The Adapter is part of PaaSage, an open-source platform for modeling, deploying and executing applications on different clouds in an optimal manner. The Adapter has the following responsibilities: (1) validating reconfiguration plans, (2) applying the plans to the running system, and (3) maintaining an up-to-date representation of the current system state.

- Contact: Nikolaos Parlavantzas
- URL: <https://team.inria.fr/myriads/software-and-platforms/paasage-adapter/>

6.2. SAIDS

self-adaptable intrusion detection system

KEYWORDS: Cloud - Security

FUNCTIONAL DESCRIPTION: SAIDS is a self-adaptable intrusion detection system for IaaS clouds. To maintain an effective level of intrusion detection, SAIDS monitors changes in the virtual infrastructure of a Cloud environment and reconfigures its components (security probes) accordingly. SAIDS can also reconfigure probes in the case of a change in the list of running services.

- Authors: Anna Giannakou and Jean-Léon Cusinato
- Contact: Christine Morin

6.3. SimGrid

KEYWORDS: Large-scale Emulators - Grid Computing - Distributed Applications

SCIENTIFIC DESCRIPTION: SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The simulation engine uses algorithmic and implementation techniques toward the fast simulation of large systems on a single machine. The models are theoretically grounded and experimentally validated. The results are reproducible, enabling better scientific practices.

Its models of networks, cpus and disks are adapted to (Data)Grids, P2P, Clouds, Clusters and HPC, allowing multi-domain studies. It can be used either to simulate algorithms and prototypes of applications, or to emulate real MPI applications through the virtualization of their communication, or to formally assess algorithms and applications that can run in the framework.

The formal verification module explores all possible message interleavings in the application, searching for states violating the provided properties. We recently added the ability to assess liveness properties over arbitrary and legacy codes, thanks to a system-level introspection tool that provides a finely detailed view of the running application to the model checker. This can for example be leveraged to verify both safety or liveness properties, on arbitrary MPI code written in C/C++/Fortran.

NEWS OF THE YEAR: There were 3 major releases in 2019: Python bindings were introduced, SMPI now partially supports some of the MPI/IO functions, a new model for Wifi networks was proposed, and the API for the simulation of storage resources was completely revisited. We also pursued our efforts to improve the documentation of the software, simplified the web site, and made a lot of bug fixing and code refactoring.

- Participants: Adrien Lèbre, Arnaud Legrand, Augustin Degomme, Florence Perronnin, Frédéric Suter, Jean-Marc Vincent, Jonathan Pastor, Luka Stanisic and Martin Quinson
- Partners: CNRS - ENS Rennes
- Contact: Martin Quinson
- URL: <https://simgrid.org/>

6.4. DiFFuSE

Distributed framework for cloud-based epidemic simulations

KEYWORDS: Simulation - Cloud

FUNCTIONAL DESCRIPTION: The DiFFuSE framework enables simulations of epidemics to take full advantage of cloud environments. The framework provides design support, reusable code, and tools for building and executing epidemic simulations. Notably, the framework automatically handles failures and supports elastic allocation of resources from multiple clouds.

- Authors: Yvon Jégou, Manh Linh Pham, Nikolaos Parlavantzas and Christine Morin
- Contact: Nikolaos Parlavantzas
- Publication: [hal-01612979/](https://hal.archives-ouvertes.fr/hal-01612979/)
- URL: <https://team.inria.fr/myriads/software-and-platforms/diffuse/>

6.5. GinFlow

KEYWORDS: Workflow - Distributed computing - Distributed - Distributed Applications - Dynamic adaptation - Framework

FUNCTIONAL DESCRIPTION: GinFlow decentralizes the coordination of the execution of workflow-based applications. GinFlow relies on an architecture where multiple service agents (SA) coordinate each others through a shared space containing the workflow description and current status. GinFlow allows the user to define several variants of a workflow and to switch from one to the other during run time.

- Participants: Cédric Tedeschi, Hector Fernandez, Javier Rojas Balderrama, Matthieu Simonin and Thierry Priol
- Partner: Université de Rennes 1
- Contact: Cédric Tedeschi
- URL: <http://ginflow.inria.fr>

6.6. libcvss

KEYWORD: Cybersecurity

FUNCTIONAL DESCRIPTION: libcvss is a Rust implementation of the CVSS specification. The supported versions of CVSS are 2.0, 3.0 and 3.1.

The official CVSS website describes CVSS this way: "The Common Vulnerability Scoring System (CVSS) provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity. The numerical score can then be translated into a qualitative representation (such as low, medium, high, and critical) to help organizations properly assess and prioritize their vulnerability management processes."

libcvss provides Rust users with a native way to manipulate CVSS-formatted vulnerability data. Rust is leveraged to provide a CVSS implementation focused on both performance and correctness.

- Participant: Clement El Baz
- Contact: Clement El Baz
- URL: <https://crates.io/crates/libcvss>

SPIRALS Project-Team

6. New Software and Platforms

6.1. APISENSE

KEYWORDS: Mobile sensing - Crowd-sensing - Mobile application - Crowd-sourcing - Android

FUNCTIONAL DESCRIPTION: APISENSE platform is a software solution to collect various contextual information from Android devices (client application) and automatically upload collected data to a server (deployed as a SaaS). APISENSE is based on a Cloud computing infrastructure to facilitate datasets collection from significant populations of mobile users for research purposes.

- Participants: Antoine Veuille, Christophe Ribeiro, Julien Duribreux, Nicolas Haderer, Romain Rouvoy, Romain Sommerard and Lakhdar Meftah
- Partner: Université de Lille
- Contact: Romain Rouvoy
- URL: <https://apisense.io>

6.2. PowerAPI

KEYWORDS: Energy efficiency - Energy management

FUNCTIONAL DESCRIPTION: PowerAPI is a library for monitoring the energy consumption of software systems.

PowerAPI differs from existing energy process-level monitoring tool in its software orientation, with a fully customizable and modular solution that let the user to precisely define what he/she wants to monitor. PowerAPI is based on a modular and asynchronous event-driven architecture using the Akka library. PowerAPI offers an API which can be used to define requests about energy spent by a process, following its hardware resource utilization (in term of CPU, memory, disk, network, etc.).

- Participants: Adel Noureddine, Loïc Huertas, Maxime Colmant, Romain Rouvoy, Mohammed Chakib Belgaid and Arthur D'azemar
- Contact: Romain Rouvoy
- URL: <http://powerapi.org>

6.3. Saloon

KEYWORDS: Feature Model - Software Product Line - Cloud computing - Model-driven engineering - Ontologies

FUNCTIONAL DESCRIPTION: Saloon is a framework for the selection and configuration of Cloud providers according to application requirements. The framework enables the specification of such requirements by defining ontologies. Each ontology provides a unified vision of provider offers in terms of frameworks, databases, languages, application servers and computational resources (i.e., memory, storage and CPU frequency). Furthermore, each provider is related to a Feature Model (FM) with attributes and cardinalities, which captures its capabilities. By combining the ontology and FMs, the framework is able to match application requirements with provider capabilities and select a suitable one. Specific scripts to the selected provider are generated in order to enable its configuration.

- Participants: Clément Quinton, Daniel Romero Acero, Laurence Duchien, Lionel Seinturier and Romain Rouvoy
- Partner: Université de Lille
- Contact: Clément Quinton
- URL: <https://gitlab.irisa.fr/drome00A/saloon>

6.4. SPOON

KEYWORDS: Java - Code analysis

FUNCTIONAL DESCRIPTION: Spoon is an open-source library that enables you to transform (see below) and analyze Java source code (see example). Spoon provides a complete and fine-grained Java metamodel where any program element (classes, methods, fields, statements, expressions...) can be accessed both for reading and modification. Spoon takes as input source code and produces transformed source code ready to be compiled.

- Participants: Gérard Paligot, Lionel Seinturier, Martin Monperrus, Nicolas Petitprez and Simon Urli
- Contact: Martin Monperrus
- URL: <http://spoon.gforge.inria.fr>

STACK Project-Team

6. New Software and Platforms

6.1. MAD

Madeus Application Deployer

KEYWORDS: Automatic deployment - Distributed Software - Component models - Cloud computing

SCIENTIFIC DESCRIPTION: MAD is a Python implementation of the Madeus deployment model for multi-component distributed software. Precisely, it allows to: 1. describe the deployment process and the dependencies of distributed software components in accordance with the Madeus model, 2. describe an assembly of components, resulting in a functional distributed software, 3. automatically deploy the component assembly of distributed software following the operational semantics of Madeus.

RELEASE FUNCTIONAL DESCRIPTION: Initial submission with basic functionalities of MAD

NEWS OF THE YEAR: Operational prototype.

- Participants: Christian Pérez, Dimitri Pertin, H el ene Coullon and Maverick Chardet
- Partners: IMT Atlantique - LS2N - LIP
- Contact: H el ene Coullon
- Publications: [Madeus: A formal deployment model - Behavioral interfaces for reconfiguration of component models](#)

6.2. Nitro

KEYWORDS: Cloud storage - Virtual Machine Image - Geo-distribution

SCIENTIFIC DESCRIPTION: Nitro is a storage system that is designed to work in geo-distributed cloud environments (i.e., over WAN) to efficiently manage Virtual Machine Images (VMIs).

Nitro employs fixed-size deduplication to store VMIs. This technique contributes to minimizing the network cost. Also, Nitro incorporates a network-aware scheduling algorithm (based on max flow algorithm) to determine which chunks should be pulled from which site in order to reconstruct the corresponding image on the destination site, with minimal (provisioning) time.

FUNCTIONAL DESCRIPTION: Geo-distributed Storage System to optimize Images (VM, containers, ...) management, in terms of cost and time, in geographically distributed cloud environment (i.e. data centers are connected over WAN).

- Authors: Jad Darrous, Shadi Ibrahim and Christian P erez
- Contact: Shadi Ibrahim
- URL: <https://gitlab.inria.fr/jdarrous/nitro>

6.3. VMPlaces

KEYWORDS: Simulation - Virtualization - Scheduling

FUNCTIONAL DESCRIPTION: VMPlaces is a dedicated framework to evaluate and compare VM placement algorithms. This framework is composed of two major components: the injector and the VM placement algorithm. The injector is the generic part of the framework (i.e. the one you can directly use) while the VM placement algorithm is the part you want to study (or compare with available algorithms). Currently, the VMPlaceS is released with three algorithms:

Entropy, a centralized approach using a constraint programming approach to solve the placement/reconfiguration VM problem

Snooze, a hierarchical approach where each manager of a group invokes Entropy to solve the placement/reconfiguration VM problem. Note that in the original implementation of Snooze, it is using a specific heuristic to solve the placement/reconfiguration VM problem. As the sake of simplicity, we have simply reused the entropy scheduling code.

DVMS, a distributed approach that dynamically partitions the system and invokes Entropy on each partition.

- Participants: Adrien Lèbre, Jonathan Pastor and Mario Südholt
- Contact: Adrien Lèbre
- URL: <http://beyondtheclouds.github.io/VMPlaceS/>

6.4. ENOS

Experimental eNvironment for OpenStack

KEYWORDS: OpenStack - Experimentation - Reproducibility

FUNCTIONAL DESCRIPTION: Enos workflow :

A typical experiment using Enos is the sequence of several phases:

- enos up : Enos will read the configuration file, get machines from the resource provider and will prepare the next phase - enos os : Enos will deploy OpenStack on the machines. This phase rely highly on Kolla deployment. - enos init-os : Enos will bootstrap the OpenStack installation (default quotas, security rules, ...) - enos bench : Enos will run a list of benchmarks. Enos support Rally and Shaker benchmarks. - enos backup : Enos will backup metrics gathered, logs and configuration files from the experiment.

- Partner: Orange Labs
- Contact: Adrien Lèbre
- URL: <http://enos.readthedocs.io/en/stable/>

6.5. EnOSlib

EnOSlib is a library to help you with your experiments

KEYWORDS: Distributed Applications - Distributed systems - Evaluation - Grid Computing - Cloud computing - Experimentation - Reproducibility - Linux - Virtualization

FUNCTIONAL DESCRIPTION: EnOSlib is a library to help you with your distributed application experiments. The main parts of your experiment logic is made reusable by the following EnOSlib building blocks:

- Reusable infrastructure configuration: The provider abstraction allows you to run your experiment on different environments (locally with Vagrant, Grid'5000, Chameleon and more) - Reusable software provisioning: In order to configure your nodes, EnOSlib exposes different APIs with different level of expressivity - Reusable experiment facilities: Tasks help you to organize your experimentation workflow.

EnOSlib is designed for experimentation purpose: benchmark in a controlled environment, academic validation ...

- Contact: Matthieu Simonin
- Publications: [Toward a Holistic Framework for Conducting Scientific Evaluations of OpenStack - EnosStack: A LAMP-like stack for the experimenter](#)
- URL: <https://discovery.gitlabpages.inria.fr/enoslib/>

6.6. Concerto

KEYWORDS: Reconfiguration - Distributed Software - Component models - Dynamic software architecture

FUNCTIONAL DESCRIPTION: Concerto is an implementation of the formal model Concerto written in Python. Concerto allows to : 1. describe the life-cycle and the dependencies of software components, 2. describe a components assembly that forms the overall life-cycle of a distributed software, 3. automatically reconfigure a Concerto assembly of components by using a set of reconfiguration instructions as well as a formal operational semantics.

- Partners: IMT Atlantique - LS2N - LIP
- Contact: Maverick Chardet
- URL: <https://gitlab.inria.fr/VeRDi-project/concerto>

6.7. Platforms

6.7.1. OpenStack

OpenStack is the de facto open-source management system to operate and use Cloud Computing infrastructures. Started in 2012, the OpenStack foundation gathers 500 organizations including groups such as Intel, AT&T, RedHat, etc. The software platform relies on tens of services with a 6-month development cycle. It is composed of more than 2 millions of lines of code, mainly in Python, just for the core services. While these aspects make the whole ecosystem quite swift, they are also good signs of maturity of this community. We created and animated between 2016 and 2018 the Fog/Edge/Massively Distributed (FEMDC) Special Interest Group⁰ and have been contributing to the Performance working group since 2015. The former investigates how OpenStack can address Fog/Edge Computing use cases whereas the latter addresses scalability, reactivity and high-availability challenges. In addition to releasing white papers and guidelines [96], the major result from the academic view point is the aforementioned EnOS solution, a holistic framework to conduct performance evaluations of OpenStack (control and data plane). In May 2018, the FEMDC SiG turned into a larger group under the control of the OpenStack foundation. This group gathers large companies such as Verizon, ATT, etc.

6.7.2. Grid'5000

Grid'5000 is a large-scale and versatile testbed for experiment-driven research in all areas of computer science, with a focus on parallel and distributed computing including Cloud, HPC and Big Data. It provides access to a large amount of resources: 12000 cores, 800 compute-nodes grouped in homogeneous clusters, and featuring various technologies (GPU, SSD, NVMe, 10G and 25G Ethernet, Infiniband, Omni-Path) and advanced monitoring and measurement features for traces collection of networking and power consumption, providing a deep understanding of experiments. It is highly reconfigurable and controllable. Researchers can experiment with a fully customized software stack thanks to bare-metal deployment features, and can isolate their experiment at the networking layer advanced monitoring and measurement features for traces collection of networking and power consumption, providing a deep understanding of experiments designed to support Open Science and reproducible research, with full traceability of infrastructure and software changes on the testbed. STACK members are strongly involved into the management and the supervision of the testbed, notably through the steering committee or the SeDuCe testbed described hereafter.

6.7.3. SeDuCe

The SeDuCe Project aims to deliver a research testbed dedicated to holistic research studies on energetical aspects of datacenters. Part of the Grid'5000 Nantes' site, this infrastructure is composed of probes that measure the power consumption of each server, each switch and each cooling system, and also measure the temperature at the front and the back of each servers. These sensors enable research to cover a full spectrum of the energetical aspect of datacenters, such as cooling and power consumption depending of experimental conditions.

The testbed is connected to renewable energy sources (solar panels). This "green" datacenter will enable researchers to perform real experiment-driven studies on fields such as temperature based scheduling or "green" aware software (*i.e.*, software that take into account renewable energies and weather conditions).

⁰https://wiki.openstack.org/wiki/Fog_Edge_Massively_Distributed_Clouds

6.7.4. SILECS

STACK Members are involved in the definition and bootstrap of the SILECS infrastructure. This infrastructure can be seen as a merge of the Grid'5000 and FIT testbeds with the goal of providing a common platform for experimental computer Science (Next Generation Internet, Internet of things, clouds, HPC, big data, etc.).

WHISPER Project-Team

6. New Software and Platforms

6.1. Coccinelle

KEYWORDS: Code quality - Evolution - Infrastructure software

FUNCTIONAL DESCRIPTION: Coccinelle is a tool for code search and transformation for C programs. It has been extensively used for bug finding and evolutions in Linux kernel code.

- Participants: Gilles Muller, Julia Lawall, Nicolas Palix, Rene Rydhof Hansen and Thierry Martinez
- Partners: LIP6 - IRILL
- Contact: Julia Lawall
- URL: <http://coccinelle.lip6.fr>

6.2. Prequel

KEYWORDS: Code search - Git

SCIENTIFIC DESCRIPTION: The commit history of a code base such as the Linux kernel is a gold mine of information on how evolutions should be made, how bugs should be fixed, etc. Nevertheless, the high volume of commits available and the rudimentary filtering tools provided mean that it is often necessary to wade through a lot of irrelevant information before finding example commits that can help with a specific software development problem. To address this issue, we propose Prequel (Patch Query Language), which brings the descriptive power of code matching to the problem of querying a commit history.

FUNCTIONAL DESCRIPTION: Prequel is a tool for searching for complex patterns in the commits of software managed using git.

- Participants: Gilles Muller and Julia Lawall
- Partners: LIP6 - IRILL
- Contact: Julia Lawall
- URL: <http://prequel-pql.gforge.inria.fr/>

6.3. Usuba

KEYWORDS: Cryptography - Optimizing compiler - Synchronous Language

FUNCTIONAL DESCRIPTION: Usuba is a programming language for specifying block ciphers as well as a bitslicing compiler, for producing high-throughput and secure code.

- Contact: Pierre-Evariste Dagand
- Publication: [Usuba, Optimizing & Trustworthy Bitslicing Compiler](#)
- URL: <https://github.com/DadaIsCrazy/usuba/>

6.4. SchedDisplay

KEYWORDS: Linux kernel - Scheduling - Multicore

FUNCTIONAL DESCRIPTION: SchedDisplay is a visualization tool for SchedLog, a custom ring buffer collecting scheduling events in the Linux kernel. SchedDisplay allows kernel developers to analyze the behavior of the Linux scheduler while running a multicore application.

RELEASE FUNCTIONAL DESCRIPTION: First version released as part of a Demo made during the 10th PLOS workshop: <https://ess.cs.uni-osnabrueck.de/workshops/plos/2019/program.php>

- Partner: Oracle Labs
- Contact: Gilles Muller
- URL: <https://gitlab.inria.fr/gmuller/scheddisplay>

WIDE Project-Team

5. New Software and Platforms

5.1. WebGC

Web-based Gossip Communication

KEYWORDS: Epidemic protocols - Gossip protocols - Peer-to-peer - Web - Personalized systems - Decentralized architectures - Recommendation systems - WebRTC - Decentralized web

SCIENTIFIC DESCRIPTION: The library currently includes the implementation of two peer sampling protocols, Cyclon and the generic peer-sampling protocol from, as well as a clustering protocol. All protocols implement a common GossipProtocol “interface”

FUNCTIONAL DESCRIPTION: WebGC consists of a WebRTC-based library that supports gossip-based communication between web browsers and enables them to operate with Node-JS applications. WebGC comprises the implementation of standard gossip protocols such as Peer Sampling or Clustering, and simplifies the development of new protocols. It comprises a decentralized signaling service that makes it easier to build completely decentralized browser-based applications.

- Participants: Anne-Marie Kermarrec, Davide Frey, Matthieu Simonin and Raziel Carvajal Gomez
- Contact: Davide Frey

5.2. YALPS

KEYWORDS: Traffic-shaping - Nat traversal - Experimentation - Peer-to-peer - Simulator - Deployment

FUNCTIONAL DESCRIPTION: YALPS is an open-source Java library designed to facilitate the development, deployment, and testing of distributed applications. Applications written using YALPS can be run both in simulation and in real-world mode without changing a line of code or even recompiling the sources. A simple change in a configuration file will load the application in the proper environment. A number of features make YALPS useful both for the design and evaluation of research prototypes and for the development of applications to be released to the public. Specifically, YALPS makes it possible to run the same application as a simulation or in a real deployment. Applications communicate by means of application-defined messages which are then routed either through UDP/TCP or through YALPS’s simulation infrastructure. In both cases, YALPS’s communication layer offers features for testing and evaluating distributed protocols and applications. Communication channels can be tuned to incorporate message losses or to constrain their outgoing bandwidth. Finally, YALPS includes facilities to support operation in the presence of NATs and firewalls using relaying and NAT-traversal techniques. The implementation of YALPS includes approximately 16K lines of code, and is used in several projects by ASAP, including HEAP, AllYours-P2P, and Behave.

- Participants: Anne Marie Kermarrec, Arnaud Jegou, Davide Frey, Heverson Borba Ribeiro and Maxime Monod
- Contact: Davide Frey
- URL: <http://yalps.gforge.inria.fr/>

5.3. KIFF

KIFF: An impressively fast and efficient JAVA library for KNN construction

KEYWORD: KNN

FUNCTIONAL DESCRIPTION: This package implements the KIFF algorithm reported in [1]. KIFF is a generic, fast and scalable K-Nearest-Neighbor graph construction algorithm. This algorithm connects each object to its k most similar counterparts, according to a given similarity metric. In term of comparison, this package implements also HYREC [2] and NN-DESCENT [3]. The standalone program implements cosine similarity only, however this library supports arbitrary similarity measures.

[1] Antoine Boutet, Anne-Marie Kermarrec, Nupur Mittal, Francois Taiani. Being prepared in a sparse world: the case of KNN graph construction. ICDE 2016, Finland.

- Partner: LIRIS
- Contact: Antoine Boutet

ALPINES Project-Team

6. New Software and Platforms

6.1. FreeFem++

FreeFem++

SCIENTIFIC DESCRIPTION: FreeFem++ is a partial differential equation solver. It has its own language. freefem scripts can solve multiphysics non linear systems in 2D and 3D.

Problems involving PDE (2d, 3d) from several branches of physics such as fluid-structure interactions require interpolations of data on several meshes and their manipulation within one program. FreeFem++ includes a fast 2d-tree-based interpolation algorithm and a language for the manipulation of data on multiple meshes (as a follow up of bamg (now a part of FreeFem++)).

FreeFem++ is written in C++ and the FreeFem++ language is a C++ idiom. It runs on Macs, Windows, Unix machines. FreeFem++ replaces the older freefem and freefem+.

FUNCTIONAL DESCRIPTION: FreeFem++ is a PDE (partial differential equation) solver based on a flexible language that allows a large number of problems to be expressed (elasticity, fluids, etc) with different finite element approximations on different meshes.

- Partner: UPMC
- Contact: Frederic Hecht
- URL: <http://www.freefem.org/ff++/>

6.2. HPDDM

SCIENTIFIC DESCRIPTION: HPDDM is an efficient implementation of various domain decomposition methods (DDM) such as one- and two-level Restricted Additive Schwarz methods, the Finite Element Tearing and Interconnecting (FETI) method, and the Balancing Domain Decomposition (BDD) method. This code has been proven to be efficient for solving various elliptic problems such as scalar diffusion equations, the system of linear elasticity, but also frequency domain problems like the Helmholtz equation. A comparison with modern multigrid methods can be found in the thesis of Pierre Jolivet.

FUNCTIONAL DESCRIPTION: HPDDM is an efficient implementation of various domain decomposition methods (DDM) such as one- and two-level Restricted Additive Schwarz methods, the Finite Element Tearing and Interconnecting (FETI) method, and the Balancing Domain Decomposition (BDD) method.

- Participants: Frédéric Nataf and Pierre Jolivet
- Contact: Pierre Jolivet
- URL: <https://github.com/hpddm>

6.3. LORASC

LORASC preconditioner

KEYWORD: Preconditioner

- Participants: Laura Grigori and Rémi Lacroix
- Contact: Laura Grigori

6.4. Platforms

6.4.1. *preAlps*

KEYWORD: Preconditioned enlarged Krylov subspace methods

FUNCTIONAL DESCRIPTION: Contains enlarged Conjugate Gradient Krylov subspace method and Loras preconditioner.

- Partners: Inria
- Contact: Simplic Donfack, Laura Grigori, Olivier Tissot
- URL: <https://github.com/NLAFET/preAlps>

6.4.2. *BemTool*

KEYWORD: Boundary Element Method

FUNCTIONAL DESCRIPTION: BemTool is a C++ header-only library implementing the boundary element method for the discretisation of the Laplace, Helmholtz and Maxwell equations, in 2D and 3D. Its main purpose is the assembly of classic boundary element matrices, which can be compressed and inverted through its interface with HTOOL.

- Partners: UPMC - ANR NonlocalDD
- Contact: Xavier Claeys
- URL: <https://github.com/xclaeys/BemTool>

6.4.3. *HTool*

KEYWORD: Hierarchical Matrices

FUNCTIONAL DESCRIPTION: HTOOL is a C++ header-only library implementing compression techniques (e.g. Adaptive Cross Approximation) using hierarchical matrices. The library uses MPI and OpenMP for parallelism, and is interfaced with HPDDM for the solution of linear systems.

- Partners: CNRS - UPMC - ANR NonlocalDD
- Contact: Pierre Marchand
- URL: <https://github.com/PierreMarchand20/htool>

AVALON Project-Team

5. New Software and Platforms

5.1. DIET

Distributed Interactive Engineering Toolbox

KEYWORDS: Scheduling - Clusters - Grid - Cloud - HPC - Middleware - Data management.

FUNCTIONAL DESCRIPTION: Middleware for grids and clouds. Toolbox for the use and porting of intensive computing applications on heterogeneous architectures.

RELEASE FUNCTIONAL DESCRIPTION: - Native Google Drive Support for the data manager - Standardization of internal integer types. - New types (see Changelog for more information)

NEWS OF THE YEAR: New DIET release (DIET 2.11) is available since may 2019 Batch's GENCI support to use the GENCI resources

- Participants: Joel Faubert, Hadrien Croubois, Abdelkader Amar, Arnaud Lefray, Aurélien Bouteiller, Benjamin Isnard, Daniel Balouek, Eddy Caron, Eric Bois, Frédéric Desprez, Frédéric Lombart, Gaël Le Mahec, Guillaume Verger, Huaxi Zhang, Jean-Marc Nicod, Jonathan Rouzaud-Cornabas, Lamiel Toch, Maurice Faye, Peter Frauenkron, Philippe Combes, Philippe Laurent, Raphaël Bolze, Yves Caniou and Cyril Seguin
- Partners: CNRS - ENS Lyon - UCBL Lyon 1 - Sysfera
- Contact: Eddy Caron
- URL: <http://graal.ens-lyon.fr/diet/>

5.2. SimGrid

KEYWORDS: Large-scale Emulators - Grid Computing - Distributed Applications

SCIENTIFIC DESCRIPTION: SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The simulation engine uses algorithmic and implementation techniques toward the fast simulation of large systems on a single machine. The models are theoretically grounded and experimentally validated. The results are reproducible, enabling better scientific practices.

Its models of networks, cpus and disks are adapted to (Data)Grids, P2P, Clouds, Clusters and HPC, allowing multi-domain studies. It can be used either to simulate algorithms and prototypes of applications, or to emulate real MPI applications through the virtualization of their communication, or to formally assess algorithms and applications that can run in the framework.

The formal verification module explores all possible message interleavings in the application, searching for states violating the provided properties. We recently added the ability to assess liveness properties over arbitrary and legacy codes, thanks to a system-level introspection tool that provides a finely detailed view of the running application to the model checker. This can for example be leveraged to verify both safety or liveness properties, on arbitrary MPI code written in C/C++/Fortran.

NEWS OF THE YEAR: There were 3 major releases in 2019: Python bindings were introduced, SMPI now partially supports some of the MPI/IO functions, a new model for Wifi networks was proposed, and the API for the simulation of storage resources was completely revisited. We also pursued our efforts to improve the documentation of the software, simplified the web site, and made a lot of bug fixing and code refactoring.

- Participants: Adrien Lèbre, Arnaud Legrand, Augustin Degomme, Florence Perronnin, Frédéric Suter, Jean-Marc Vincent, Jonathan Pastor, Luka Stanic and Martin Quinson
- Partners: CNRS - ENS Rennes
- Contact: Martin Quinson
- URL: <https://simgrid.org/>

5.3. SeeDep

Seed based Deployment

KEYWORDS: Reproducibility - Deployment - Cloud

SCIENTIFIC DESCRIPTION: SeeDep aims at devising a new way where researchers can communicate in a comprehensive and accurate way the experimentation set-up used in their work. It lies on two components: (i) a public algorithm that generates experimentation networks, and (ii) a generation key (i.e. a seed) that can be shared which specifies the said network. Therefore, researchers only need to share (in their paper for instance) the “generation key” that corresponds to their experimentation network. With such key, any other researcher/professional will be able to re-generate a comprehensive and accurate model of the same network.

FUNCTIONAL DESCRIPTION: SeeDep is a framework aiming at generating, reproducing and deploying experiments set-up on different Cloud platforms.

- Participants: Cyril Seguin and Eddy Caron
- Partner: Nokia Bell Labs
- Contact: Eddy Caron

5.4. libkomp

Runtime system libkomp

KEYWORDS: HPC - Multicore - OpenMP

FUNCTIONAL DESCRIPTION: libKOMP is a runtime support for OpenMP compatible with different compiler: GNU gcc/gfortran, Intel icc/ifort or clang/llvm. It is based on source code initially developed by Intel for its own OpenMP runtime, with extensions from Kaapi software (task representation, task scheduling). Moreover it contains an OMPT module for recording trace of execution.

RELEASE FUNCTIONAL DESCRIPTION: Initial version

- Contact: Thierry Gautier
- URL: <http://gitlab.inria.fr/openmp/libkomp>

5.5. XKBLAS

KEYWORDS: BLAS - Dense linear algebra - GPU

FUNCTIONAL DESCRIPTION: XKBLAS is yet another BLAS library (Basic Linear Algebra Subroutines) that targets multi-GPUs architecture thanks to the XKaapi runtime and with block algorithms from PLASMA library. The library offers a wrapper library able to capture calls to BLAS (C or Fortran). The internal API is based on asynchronous invocations in order to enable overlapping between communication by computation and also to better composed sequences of calls to BLAS.

This current version of XKBlas is the first public version and contains only BLAS level 3 algorithms, including XGEMMT:

XGEMM XGEMMT: see MKL GEMMT interface XTRSM XTRMM XSYMM XSYRK XSYR2K XHEMM XHERK XHER2K

For classical precision Z, C, D, S.

RELEASE FUNCTIONAL DESCRIPTION: XKBlas has following limitations:

0.1 versions: calls to BLAS kernels must be initiated by the same thread that initializes the XKBlas library.

- Participants: Thierry Gautier and João Vicente Ferreira Lima
- Contact: Thierry Gautier
- URL: <https://gitlab.inria.fr/xkblas/versions>

5.6. Concerto

KEYWORDS: Reconfiguration - Distributed Software - Component models - Dynamic software architecture

FUNCTIONAL DESCRIPTION: Concerto is an implementation of the formal model Concerto written in Python. Concerto allows to : 1. describe the life-cycle and the dependencies of software components, 2. describe a components assembly that forms the overall life-cycle of a distributed software, 3. automatically reconfigure a Concerto assembly of components by using a set of reconfiguration instructions as well as a formal operational semantics.

- Partners: IMT Atlantique - LS2N - LIP
- Contact: Maverick Chardet
- URL: <https://gitlab.inria.fr/VeRDi-project/concerto>

5.7. Kwapi

KiloWatt API

KEYWORD: Power monitoring

FUNCTIONAL DESCRIPTION: Kwapi is a software framework dealing with energy monitoring of large scale infrastructures through heterogeneous energy sensors. Kwapi has been designed inside the FSN XLCloud project for Openstack infrastructures. Through the support of Hemera Inria project, kwapi has been extended and deployed in production mode to support easy and large scale energy profiling of the Grid5000 resources. Kwapi now supports high frequency powermeters of the Grid5000 Lyon platform.

RELEASE FUNCTIONAL DESCRIPTION: - many bugfixes - multiprocessing instead of multithreading - many optimizations

- Participants: François Rossigneux, Jean-Patrick Gelas, Laurent Lefèvre, Laurent Pouilloux, Simon Delamare and Matthieu Imbert
- Contact: Laurent Lefèvre
- URL: <https://launchpad.net/kwapi>

5.8. execo

KEYWORDS: Toolbox - Deployment - Orchestration - Python

FUNCTIONAL DESCRIPTION: Execo offers a Python API for asynchronous control of local or remote, standalone or parallel, unix processes. It is especially well suited for quickly and easily scripting workflows of parallel/distributed operations on local or remote hosts: automate a scientific workflow, conduct computer science experiments, perform automated tests, etc. The core python package is execo. The execo_g5k package provides a set of tools and extensions for the Grid5000 testbed. The execo_engine package provides tools to ease the development of computer sciences experiments.

RELEASE FUNCTIONAL DESCRIPTION: - misc python3 support fixes - basic documentation for wheezy compatible package build - remove some debug outputs - fix crash in processes conductor in some situations - improve/fix process stdout/stderr handlers - fix get_cluster_network equipments - add a FAQ

- Participants: Florent Chuffart, Laurent Pouilloux and Matthieu Imbert
- Contact: Matthieu Imbert
- URL: <http://execo.gforge.inria.fr>

5.9. Platforms

5.9.1. Platform: Grid'5000

Participants: Laurent Lefèvre, Simon Delamare, David Loup, Christian Perez.

FUNCTIONAL DESCRIPTION

The Grid'5000 experimental platform is a scientific instrument to support computer science research related to distributed systems, including parallel processing, high performance computing, cloud computing, operating systems, peer-to-peer systems and networks. It is distributed on 10 sites in France and Luxembourg, including Lyon. Grid'5000 is a unique platform as it offers to researchers many and varied hardware resources and a complete software stack to conduct complex experiments, ensure reproducibility and ease understanding of results. In 2018, a new generation of high speed wattmeters has been deployed on the Lyon site. They allow energy monitoring with up to 50 measurements per second. In parallel, a new version of kwapi (software stack for energy monitoring) has been proposed and redesigned.

- Contact: Laurent Lefèvre
- URL: <https://www.grid5000.fr/>

5.9.2. Platform: Leco

Participants: Thierry Gautier, Laurent Lefèvre, Christian Perez.

FUNCTIONAL DESCRIPTION

The LECO experimental platform is a new medium size scientific instrument funded by DRRT to investigate research related to BigData and HPC. It is located in Grenoble as part of the the HPCDA computer managed by UMS GRICAD. The platform has been deployed in 2018 and was available for experiment since the summer. All the nodes of the platform are instrumented to capture the energy consumption and data are available through the Kwapi software.

- Contact: Thierry Gautier

5.9.3. Platform: SILECS

Participants: Laurent Lefèvre, Simon Delamare, Christian Perez.

FUNCTIONAL DESCRIPTION

The SILECS infrastructure (IR ministère) aims at providing an experimental platform for experimental computer Science (Internet of things, clouds, HPC, big data, *etc.*). This new infrastructure is based on two existing infrastructures, Grid'5000 and FIT.

- Contact: Christian Perez
- URL: <https://www.silecs.net/>

DATAMOVE Project-Team

6. New Software and Platforms

6.1. FlowVR

SCIENTIFIC DESCRIPTION: FlowVR adopts the "data-flow" paradigm, where your application is divided as a set of components exchanging messages (think of it as a directed graph). FlowVR enables to encapsulate existing codes in components, interconnect them through data channels, and deploy them on distributed computing resources. FlowVR takes care of all the heavy lifting such as application deployment and message exchange.

The base entity, called a module or component, is an autonomous process, potentially multi-threaded with tools like OpenMP, TBB, or deferring computations to a GPU or Xeon Phi. This module processes data coming from input ports and write data on output ports. A module has no global insight on where the data comes from or goes to. The programming interface is designed to limit code refactoring, easing turning an existing code into a FlowVR component. The three main functions are:

wait(): Blocking function call that waits for the availability of new messages on input ports. get(): Retrieve a handle to access the message received at the previous wait() call on a given input port. put(): Notify FlowVR that a new message on a given output port is ready for dispatch. FlowVR manages data transfers. Intra-node communications between two components take place through a shared memory segment, avoiding copies. Once the sender has prepared the data in a shared memory segment, it simply handles a pointer to the destination that can directly access them. Inter-node communications extend this mechanism, FlowVR taking care of packing and transferring the data from the source shared memory segment to the destination shared memory segment.

Assembling components to build an application consists in writing a Python script, instanciate it according to the target machine. FlowVR will process it and prepare everything so that in one command line you can deploy and start your application.

FUNCTIONAL DESCRIPTION: FlowVR adopts the "data-flow" paradigm, where your application is divided as a set of components exchanging messages (think of it as a directed graph). FlowVR enables to encapsulate existing codes in components, interconnect them through data channels, and deploy them on distributed computing resources. FlowVR takes care of all the heavy lifting such as application deployment and message exchange.

- Participants: Bruno Raffin, Clément Ménier, Emmanuel Melin, Jean Denis Lesage, Jérémie Allard, Jérémy Jaussaud, Matthieu Dreher, Sébastien Limet, Sophie Robert and Valérie Gourantou
- Contact: Bruno Raffin
- URL: <http://flowvr.sf.net>

6.2. OAR

KEYWORDS: HPC - Cloud - Clusters - Resource manager - Light grid

SCIENTIFIC DESCRIPTION: This batch system is based on a database (PostgreSQL (preferred) or MySQL), a script language (Perl) and an optional scalable administrative tool (e.g. Taktuk). It is composed of modules which interact mainly via the database and are executed as independent programs. Therefore, formally, there is no API, the system interaction is completely defined by the database schema. This approach eases the development of specific modules. Indeed, each module (such as schedulers) may be developed in any language having a database access library.

FUNCTIONAL DESCRIPTION: OAR is a versatile resource and task manager (also called a batch scheduler) for HPC clusters, and other computing infrastructures (like distributed computing experimental testbeds where versatility is a key).

- Participants: Bruno Bzeznik, Olivier Richard and Pierre Neyron
- Partners: LIG - CNRS - Grid'5000 - CIMENT
- Contact: Olivier Richard
- URL: <http://oar.imag.fr>

6.3. MELISSA

Modular External Library for In Situ Statistical Analysis

KEYWORD: Sensitivity Analysis

FUNCTIONAL DESCRIPTION: Melissa is an in situ solution for sensitivity analysis. It implements iterative algorithms to compute spatio-temporal statistic fields over results of large scale sensitivity studies. Melissa relies on a client/server architecture, composed of three main modules:

Melissa Server: an independent parallel executable. It receives data from the simulations, updates iterative statistics as soon as possible, then throw data away. Melissa API: a shared library to be linked within the simulation code. It mainly transmit simulation data to Melissa Server at each timestep. The simulations of the sensitivity analysis become the clients of Melissa Server. Melissa Launcher: A Python script in charge of generating and managing the whole global sensitivity analysis.

- Authors: Theophile Terraz, Bruno Raffin, Alejandro Ribes and Bertrand Iooss
- Partner: Edf
- Contact: Bruno Raffin
- Publications: [In Situ Statistical Analysis for Parametric Studies - Melissa: Large Scale In Transit Sensitivity Analysis Avoiding Intermediate Files](#)
- URL: <https://melissa-sa.github.io>

6.4. Platforms

6.4.1. *Grid'5000 (<https://www.grid5000.fr/>) and Meso Center Ciment (<http://ciment.univ-grenoble-alpes.fr/>)*

We have been very active in promoting the factorization of compute resources at a regional and national level. We have a three level implication, locally to maintain a pool of very flexible experimental machines (hundreds of cores), regionally through the CIMENT meso center (Equipex Grant), and nationally by contributing to the Grid'5000 platform, our local resources being included in this platform. Olivier Richard is member of Grid'5000 scientific committee and Pierre Neyron is member of the technical committee. The OAR scheduler in particular is deployed on both infrastructures. We are currently preparing proposals for the next generation machines within the context of the new university association (Univ. Grenoble-Alpes).

HIEPACS Project-Team

6. New Software and Platforms

6.1. AVCI

Adaptive vibrational configuration interaction

KEYWORDS: Vibrational spectra - Eigen value

FUNCTIONAL DESCRIPTION: A-VCI is a theoretical vibrational spectroscopy algorithm developed to effectively reduce the number of vibrational states used in the configuration-interaction (CI) process. It constructs a nested basis for the discretization of the Hamiltonian operator inside a large CI approximation space and uses an a-posteriori error estimator (residue) to select the most relevant directions to expand the discretization space.

The Hamiltonian operator consists of 3 operators: a harmonic oscillator sum, the potential energy surface operator and the Coriolis operators. In addition, the code can compute the intensity of eigenvectors.

The code can handle molecules up to 10 atoms, which corresponds to solving an eigenvalue problem in a 24-dimensional space.

- Partner: IPREM
- Contact: Olivier Coulaud

6.2. Chameleon

KEYWORDS: Runtime system - Task-based algorithm - Dense linear algebra - HPC - Task scheduling

SCIENTIFIC DESCRIPTION: Chameleon is part of the MORSE (Matrices Over Runtime Systems @ Exascale) project. The overall objective is to develop robust linear algebra libraries relying on innovative runtime systems that can fully benefit from the potential of those future large-scale complex machines.

We expect advances in three directions based first on strong and closed interactions between the runtime and numerical linear algebra communities. This initial activity will then naturally expand to more focused but still joint research in both fields.

1. Fine interaction between linear algebra and runtime systems. On parallel machines, HPC applications need to take care of data movement and consistency, which can be either explicitly managed at the level of the application itself or delegated to a runtime system. We adopt the latter approach in order to better keep up with hardware trends whose complexity is growing exponentially. One major task in this project is to define a proper interface between HPC applications and runtime systems in order to maximize productivity and expressivity. As mentioned in the next section, a widely used approach consists in abstracting the application as a DAG that the runtime system is in charge of scheduling. Scheduling such a DAG over a set of heterogeneous processing units introduces a lot of new challenges, such as predicting accurately the execution time of each type of task over each kind of unit, minimizing data transfers between memory banks, performing data prefetching, etc. Expected advances: In a nutshell, a new runtime system API will be designed to allow applications to provide scheduling hints to the runtime system and to get real-time feedback about the consequences of scheduling decisions.

2. Runtime systems. A runtime environment is an intermediate layer between the system and the application. It provides low-level functionality not provided by the system (such as scheduling or management of the heterogeneity) and high-level features (such as performance portability). In the framework of this proposal, we will work on the scalability of runtime environment. To achieve scalability it is required to avoid all centralization. Here, the main problem is the scheduling of the tasks. In many task-based runtime environments the scheduler is centralized and becomes a bottleneck as soon as too many cores are involved. It is therefore required to distribute the scheduling decision or to compute a data distribution that impose the mapping of task using, for instance the so-called “owner-compute” rule. Expected advances: We will design runtime systems that enable an efficient and scalable use of thousands of distributed multicore nodes enhanced with accelerators.

3. Linear algebra. Because of its central position in HPC and of the well understood structure of its algorithms, dense linear algebra has often pioneered new challenges that HPC had to face. Again, dense linear algebra has been in the vanguard of the new era of petascale computing with the design of new algorithms that can efficiently run on a multicore node with GPU accelerators. These algorithms are called “communication-avoiding” since they have been redesigned to limit the amount of communication between processing units (and between the different levels of memory hierarchy). They are expressed through Direct Acyclic Graphs (DAG) of fine-grained tasks that are dynamically scheduled. Expected advances: First, we plan to investigate the impact of these principles in the case of sparse applications (whose algorithms are slightly more complicated but often rely on dense kernels). Furthermore, both in the dense and sparse cases, the scalability on thousands of nodes is still limited, new numerical approaches need to be found. We will specifically design sparse hybrid direct/iterative methods that represent a promising approach.

Overall end point. The overall goal of the MORSE associate team is to enable advanced numerical algorithms to be executed on a scalable unified runtime system for exploiting the full potential of future exascale machines.

FUNCTIONAL DESCRIPTION: Chameleon is a dense linear algebra software relying on sequential task-based algorithms where sub-tasks of the overall algorithms are submitted to a Runtime system. A Runtime system such as StarPU is able to manage automatically data transfers between not shared memory area (CPUs-GPUs, distributed nodes). This kind of implementation paradigm allows to design high performing linear algebra algorithms on very different type of architecture: laptop, many-core nodes, CPUs-GPUs, multiple nodes. For example, Chameleon is able to perform a Cholesky factorization (double-precision) at 80 TFlop/s on a dense matrix of order 400 000 (i.e. 4 min 30 s).

RELEASE FUNCTIONAL DESCRIPTION: Chameleon includes the following features:

- BLAS 3, LAPACK one-sided and LAPACK norms tile algorithms - Support QUARK and StarPU runtime systems and PaRSEC since 2018 - Exploitation of homogeneous and heterogeneous platforms through the use of BLAS/LAPACK CPU kernels and cuBLAS/MAGMA CUDA kernels - Exploitation of clusters of interconnected nodes with distributed memory (using OpenMPI)

- Participants: Cédric Castagnede, Samuel Thibault, Emmanuel Agullo, Florent Pruvost and Mathieu Faverge
- Partners: Innovative Computing Laboratory (ICL) - King Abdullha University of Science and Technology - University of Colorado Denver
- Contact: Emmanuel Agullo
- URL: <https://gitlab.inria.fr/solverstack/chameleon>

6.3. Diodon

KEYWORDS: Dimensionality reduction - Data analysis

FUNCTIONAL DESCRIPTION: Most of dimension reduction methods inherited from Multivariate Data Analysis, and currently implemented as element in statistical learning for handling very large datasets (the dimension of spaces is the number of features) rely on a chain of pretreatments, a core with a SVD for low rank approximation of a given matrix, and a post-treatment for interpreting results. The costly part in computations is the SVD, which is in cubic complexity. Diodon is a list of functions and drivers which implement (i) pre-treatments, SVD and post-treatments on a large diversity of methods, (ii) random projection methods for running the SVD which permits to bypass the time limit in computing the SVD, and (iii) an implementation in C++ of the SVD with random projection at prescribed rank or precision, connected to MDS.

- Contact: Alain Franc

6.4. DPLASMA

Distributed Parallel Linear Algebra Software for Multicore Architectures

FUNCTIONAL DESCRIPTION: DPLASMA is the leading implementation of a dense linear algebra package for distributed heterogeneous systems. It is designed to deliver sustained performance for distributed systems where each node featuring multiple sockets of multicore processors, and if available, accelerators like GPUs or Intel Xeon Phi. DPLASMA achieves this objective through the state of the art ParSEC runtime, porting the PLASMA algorithms to the distributed memory realm.

- Contact: Mathieu Faverge
- URL: <http://icl.cs.utk.edu/parsec/index.html>

6.5. Fabulous

Fast Accurate Block Linear krylOv Solver

KEYWORDS: Numerical algorithm - Block Krylov solver

SCIENTIFIC DESCRIPTION: Versatile and flexible numerical library that implements Block Krylov iterative schemes for the solution of linear systems of equations with multiple right-hand sides

FUNCTIONAL DESCRIPTION: Versatile and flexible numerical library that implements Block Krylov iterative schemes for the solution of linear systems of equations with multiple right-hand sides. The library implements block variants of minimal norm residual variants with partial convergence management and spectral information recycling. The package already implements regular block-GMRES (BGMRES), Inexact Breakdown BGMRES (IB-BMGRES), Inexact Breakdown BGMRES with Deflated Restarting (IB-BGMRES-DR), Block Generalized Conjugate Residual with partial convergence management. The C++ library relies on callback mechanisms to implement the calculations (matrix-vector, dot-product, ...) that depend on the parallel data distribution selected by the user.

- Participants: Emmanuel Agullo, Luc Giraud, Gilles Marait and Cyrille Piacibello
- Contact: Luc Giraud
- Publication: [Block GMRES method with inexact breakdowns and deflated restarting](#)
- URL: <https://gitlab.inria.fr/solverstack/fabulous/>

6.6. MAPHYS

Massively Parallel Hybrid Solver

KEYWORD: Parallel hybrid direct/iterative solution of large linear systems

FUNCTIONAL DESCRIPTION: MaPHyS is a software package that implements a parallel linear solver coupling direct and iterative approaches. The underlying idea is to apply to general unstructured linear systems domain decomposition ideas developed for the solution of linear systems arising from PDEs. The interface problem, associated with the so called Schur complement system, is solved using a block preconditioner with overlap between the blocks that is referred to as Algebraic Additive Schwarz. A fully algebraic coarse space is available for symmetric positive definite problems, that insures the numerical scalability of the preconditioner.

The parallel implementation is based on MPI+thread. Maphys relies on state-of-the art sparse and dense direct solvers.

MaPHyS is essentially a preconditioner that can be used to speed-up the convergence of any Krylov subspace method and is coupled with the ones implemented in the Fabulous package.

- Participants: Emmanuel Agullo, Luc Giraud, Matthieu Kuhn, Gilles Marait and Louis Poirel
- Contact: Emmanuel Agullo
- Publications: [Hierarchical hybrid sparse linear solver for multicore platforms - Robust coarse spaces for Abstract Schwarz preconditioners via generalized eigenproblems](#)
- URL: <https://gitlab.inria.fr/solverstack/maphys>

6.7. MetaPart

KEYWORDS: High performance computing - HPC - Parallel computing - Graph algorithmics - Graph - Hypergraph

FUNCTIONAL DESCRIPTION: MetaPart is a framework for graph or hypergraph manipulation that addresses different problems, like partitioning, repartitioning, or co-partitioning, ... MetaPart is made up of several projects, such as StarPart, LibGraph or CoPart. StarPart is the core of the MetaPart framework. It offers a wide variety of graph partitioning methods (Metis, Scotch, Zoltan, Patoh, ParMetis, Kahip, ...), which makes it easy to compare these different methods and to better adjust the parameters of these methods. It is built upon the LibGraph library, that provides basic graph and hypergraph routines. The Copart project is a library used on top of StarPart, that provides co-partitioning algorithms for the load-balancing of parallel coupled simulations.

- Participant: Aurélien Esnard
- Contact: Aurélien Esnard
- URL: <https://gitlab.inria.fr/metapart>

6.8. MPICPL

MPI CouPLing

KEYWORDS: MPI - Coupling software

FUNCTIONAL DESCRIPTION: MPICPL is a software library dedicated to the coupling of parallel legacy codes, that are based on the well-known MPI standard. It proposes a lightweight and comprehensive programming interface that simplifies the coupling of several MPI codes (2, 3 or more). MPICPL facilitates the deployment of these codes thanks to the mpicplrun tool and it interconnects them automatically through standard MPI inter-communicators. Moreover, it generates the universe communicator, that merges the world communicators of all coupled-codes. The coupling infrastructure is described by a simple XML file, that is just loaded by the mpicplrun tool.

- Participant: Aurélien Esnard
- Contact: Aurélien Esnard
- URL: <https://gitlab.inria.fr/esnard/mpicpl>

6.9. OptiDis

KEYWORDS: Dislocation dynamics simulation - Fast multipole method - Large scale - Collision

FUNCTIONAL DESCRIPTION: OptiDis is a new code for large scale dislocation dynamics simulations. Its purpose is to simulate real life dislocation densities (up to 5.1022 dislocations/m²) in order to understand plastic deformation and study strain hardening. The main application is to observe and understand plastic deformation of irradiated zirconium. Zirconium alloys are the first containment barrier against the dissemination of radioactive elements. More precisely, with neutron irradiated zirconium alloys we are talking about channeling mechanism, which means to stick with the reality, more than tens of thousands of induced loops, i. e. 100 million degrees of freedom in the simulation. The code is based on Numodis code developed at CEA Saclay and the ScalFMM library developed in HiePACS project. The code is written in C++ language and using the last features of C++11/14. One of the main aspects is the hybrid parallelism MPI/OpenMP that gives the software the ability to scale on large cluster while the computation load rises. In order to achieve that, we use different levels of parallelism. First of all, the simulation box is distributed over MPI processes, then we use a thinner level for threads, dividing the domain by an Octree representation. All these parts are controlled by the ScalFMM library. On the last level, our data are stored in an adaptive structure that absorbs the dynamics of this type of simulation and manages the parallelism of tasks..

- Participant: Olivier Coulaud
- Contact: Olivier Coulaud
- URL: <http://optidis.gforge.inria.fr/>

6.10. PaStiX

Parallel Sparse matrix package

KEYWORDS: Linear algebra - High-performance calculation - Sparse Matrices - Linear Systems Solver - Low-Rank compression

SCIENTIFIC DESCRIPTION: PaStiX is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million of unknowns. The mapping and scheduling algorithm handle a combination of 1D and 2D block distributions. A dynamic scheduling can also be applied to take care of NUMA architectures while taking into account very precisely the computational costs of the BLAS 3 primitives, the communication costs and the cost of local aggregations.

FUNCTIONAL DESCRIPTION: PaStiX is a scientific library that provides a high performance parallel solver for very large sparse linear systems based on block direct and block ILU(k) methods. It can handle low-rank compression techniques to reduce the computation and the memory complexity. Numerical algorithms are implemented in single or double precision (real or complex) for LLt, LDLt and LU factorization with static pivoting (for non symmetric matrices having a symmetric pattern). The PaStiX library uses the graph partitioning and sparse matrix block ordering packages Scotch or Metis.

The PaStiX solver is suitable for any heterogeneous parallel/distributed architecture when its performance is predictable, such as clusters of multicore nodes with GPU accelerators or KNL processors. In particular, we provide a high-performance version with a low memory overhead for multicore node architectures, which fully exploits the advantage of shared memory by using an hybrid MPI-thread implementation.

The solver also provides some low-rank compression methods to reduce the memory footprint and/or the time-to-solution.

- Participants: Tony Delarue, Grégoire Pichon, Mathieu Faverge, EsragÜl Korkmaz and Pierre Ramet
- Partners: Université Bordeaux 1 - INP Bordeaux
- Contact: Pierre Ramet
- URL: <https://gitlab.inria.fr/solverstack/pastix>

6.11. pmtool

KEYWORDS: Scheduling - Task scheduling - StarPU - Heterogeneity - GPGPU - Performance analysis

FUNCTIONAL DESCRIPTION: Analyse post-mortem the behavior of StarPU applications. Provide lower bounds on makespan. Study the performance of different schedulers in a simple context. Provide implementations of many scheduling algorithms from the literature

NEWS OF THE YEAR: Included many new algorithms, in particular online algorithms Better integration with StarPU by accepting .rec files as input

- Participant: Lionel Eyraud-Dubois
- Contact: Lionel Eyraud-Dubois
- Publications: [Approximation Proofs of a Fast and Efficient List Scheduling Algorithm for Task-Based Runtime Systems on Multicores and GPUs](#) - [Fast Approximation Algorithms for Task-Based Runtime Systems](#)
- URL: <https://gitlab.inria.fr/eyrauddu/pmtool>

6.12. rotor

Re-materializing Optimally with pyTORch

KEYWORDS: Deep learning - Optimization - Python - GPU - Automatic differentiation

FUNCTIONAL DESCRIPTION: Allows to train very large convolutional networks on limited memory by optimally selecting which activations should be kept and which should be recomputed. This code is meant to replace the checkpoint.py utility available in pytorch, by providing more efficient rematerialization strategies. The algorithm is easier to tune: the only required parameter is the available memory, instead of the number of segments.

- Contact: Lionel Eyraud-Dubois

6.13. ScalFMM

Scalable Fast Multipole Method

KEYWORDS: N-body - Fast multipole method - Parallelism - MPI - OpenMP

SCIENTIFIC DESCRIPTION: ScalFMM is a software library to simulate N-body interactions using the Fast Multipole Method. The library offers two methods to compute interactions between bodies when the potential decays like $1/r$. The first method is the classical FMM based on spherical harmonic expansions and the second is the Black-Box method which is an independent kernel formulation (introduced by E. Darve @ Stanford). With this method, we can now easily add new non oscillatory kernels in our library. For the classical method, two approaches are used to decrease the complexity of the operators. We consider either matrix formulation that allows us to use BLAS routines or rotation matrix to speed up the M2L operator.

ScalFMM intends to offer all the functionalities needed to perform large parallel simulations while enabling an easy customization of the simulation components: kernels, particles and cells. It works in parallel in a shared/distributed memory model using OpenMP and MPI. The software architecture has been designed with two major objectives: being easy to maintain and easy to understand. There is two main parts:

the management of the octree and the parallelization of the method the kernels. This new architecture allow us to easily add new FMM algorithm or kernels and new paradigm of parallelization.

FUNCTIONAL DESCRIPTION: Compute N-body interactions using the Fast Multipole Method for large number of objects

- Participants: Bramas Bérenger, Olivier Coulaud and Pierre Estérie
- Contact: Olivier Coulaud
- URL: <https://gitlab.inria.fr/solverstack/ScalFMM>

6.14. VITE

Visual Trace Explorer

KEYWORDS: Visualization - Execution trace

FUNCTIONAL DESCRIPTION: ViTE is a trace explorer. It is a tool made to visualize execution traces of large parallel programs. It supports Pajé, a trace format created by Inria Grenoble, and OTF and OTF2 formats, developed by the University of Dresden and allows the programmer a simpler way to analyse, debug and/or profile large parallel applications.

- Participant: Mathieu Faverge
- Contact: Mathieu Faverge
- URL: <http://vite.gforge.inria.fr/>

6.15. PlaFRIM

Plateforme Fédérative pour la Recherche en Informatique et Mathématiques

KEYWORDS: High-Performance Computing - Hardware platform

FUNCTIONAL DESCRIPTION: PlaFRIM is an experimental platform for research in modeling, simulations and high performance computing. This platform has been set up from 2009 under the leadership of Inria Bordeaux Sud-Ouest in collaboration with computer science and mathematics laboratories, respectively Labri and IMB with a strong support in the region Aquitaine.

It aggregates different kinds of computational resources for research and development purposes. The latest technologies in terms of processors, memories and architecture are added when they are available on the market. It is now more than 1,000 cores (excluding GPU and Xeon Phi) that are available for all research teams of Inria Bordeaux, Labri and IMB. This computer is in particular used by all the engineers who work in HiePACS and are advised by F. Rue from the SED.

- Contact: Olivier Coulaud
- URL: <https://www.plafrim.fr/en/home/>

KERDATA Project-Team

6. New Software and Platforms

6.1. Damaris

KEYWORDS: Visualization - I/O - HPC - Exascale - High performance computing

SCIENTIFIC DESCRIPTION: Damaris is a middleware for I/O and data management targeting large-scale, MPI-based HPC simulations. It initially proposed to dedicate cores for asynchronous I/O in multicore nodes of recent HPC platforms, with an emphasis on ease of integration in existing simulations, efficient resource usage (with the use of shared memory) and simplicity of extension through plug-ins. Over the years, Damaris has evolved into a more elaborate system, providing the possibility to use dedicated cores or dedicated nodes to in situ data processing and visualization. It proposes a seamless connection to the VisIt visualization framework to enable in situ visualization with minimum impact on run time. Damaris provides an extremely simple API and can be easily integrated into the existing large-scale simulations.

Damaris was at the core of the PhD thesis of Matthieu Dorier, who received an Accessit to the Gilles Kahn Ph.D. Thesis Award of the SIF and the Academy of Science in 2015. Developed in the framework of our collaboration with the JLESC – Joint Laboratory for Extreme-Scale Computing, Damaris was the first software resulted from this joint lab validated in 2011 for integration to the Blue Waters supercomputer project. It scaled up to 16,000 cores on Oak Ridge’s leadership supercomputer Titan (first in the Top500 supercomputer list in 2013) before being validated on other top supercomputers. Active development is currently continuing within the KerData team at Inria, where it is at the center of several collaborations with industry as well as with national and international academic partners.

FUNCTIONAL DESCRIPTION: Damaris is a middleware for data management and in-situ visualization targeting large-scale HPC simulations: - In situ data analysis by some dedicated cores/nodes of the simulation platform - Asynchronous and fast data transfer from HPC simulations to Damaris - Semantic-aware dataset processing through Damaris plug-ins - Writing aggregated data (by hdf5 format) or visualizing them either by VisIt or ParaView

- Participants: Gabriel Antoniu, Lokman Rahmani, Luc Bougé, Matthieu Dorier, Orçun Yildiz and Hadi Salimi
- Partner: ENS Rennes
- Contact: Matthieu Dorier
- URL: <https://project.inria.fr/damaris/>

6.2. OverFlow

FUNCTIONAL DESCRIPTION: OverFlow is a uniform data management system for scientific workflows running across geographically distributed sites, aiming to reap economic benefits from this geo-diversity. The software is environment-aware, as it monitors and models the global cloud infrastructure, offering high and predictable data handling performance for transfer cost and time, within and across sites. OverFlow proposes a set of pluggable services, grouped in a data-scientist cloud kit. They provide the applications with the possibility to monitor the underlying infrastructure, to exploit smart data compression, deduplication and geo-replication, to evaluate data management costs, to set a tradeoff between money and time, and optimize the transfer strategy accordingly.

Currently, OverFlow is used for data transfers by the Microsoft Research ATLE Munich team as well as for synthetic benchmarks at the Politehnica University of Bucharest.

- Participants: Alexandru Costan, Gabriel Antoniu and Radu Marius Tudoran
- Contact: Alexandru Costan

6.3. Pufferbench

KEYWORDS: Distributed Storage Systems - Elasticity - Benchmarking

SCIENTIFIC DESCRIPTION: Pufferbench is a benchmark for evaluating how fast one can scale up and down a distributed storage system on a given infrastructure and, thereby, how viably can one implement storage malleability on it. Besides, it can serve to quickly prototype and evaluate mechanisms for malleability in existing distributed storage systems.

FUNCTIONAL DESCRIPTION: Pufferbench is a benchmark to designed to evaluate whether to use malleable distributed storage systems on a given platform. - It measures the duration of commission and decommission operations. - Its modularity allows to quickly change and adapt each component to the needs of the user. - It can serve as a baseline when implementing commission and decommission mechanisms in a distributed storage system.

RELEASE FUNCTIONAL DESCRIPTION: This is the first release of Pufferbench.

It includes default components for each of the customisable components: - storage: in memory, on drive with file system cache, and on drive without file system cache - network: MPI network - IODispatcher: basic, and with acknowledgements - DataTransferScheduler: basic - DataDistributionGenerator: uniform, and random - MetadataGenerator: Files of same size The diversity of available components enables Pufferbench to fit to multiple use cases.

- Participants: Nathanaël Cherièr, Matthieu Dorier and Gabriel Antoniu
- Partner: ENS Rennes
- Contact: Nathanaël Cherièr
- Publication: [hal-01886351](#)
- URL: <https://gitlab.inria.fr/Puffertools/Pufferbench/wikis/home>

6.4. Tyr

KEYWORDS: Cloud storage - Distributed Storage Systems - Big data

FUNCTIONAL DESCRIPTION: Tyr is the first blob storage system to provide built-in, multiblob transactions, while retaining sequential consistency and high throughput under heavy access concurrency. Tyr offers fine-grained random write access to data and in-place atomic operations.

- Partner: Universidad Politécnica de Madrid
- Contact: Gabriel Antoniu

6.5. Planner

KEYWORDS: Edge elements - Cloud computing - Scheduling

FUNCTIONAL DESCRIPTION: Planner is a middleware for uniform and transparent stream processing across Edge and Cloud. Planner automatically selects which parts of the executiongraph will be executed at the Edge in order to minimize thenetwork cost.

- Partner: ENS Cachan
- Contact: Gabriel Antoniu
- URL: <https://team.inria.fr/kerdata/>

6.6. KerA

KerAnalytics

KEYWORD: Distributed Storage Systems

FUNCTIONAL DESCRIPTION: A unified architecture for stream ingestion and storage which can lead to the optimization of the processing of Big Data applications. This approach minimizes data movement within the analytics architecture, finally leading to better utilized resources.

- Contact: Gabriel Antoniu

6.7. TailWind

KEYWORDS: Fault-tolerance - Data management. - Distributed Data Management

FUNCTIONAL DESCRIPTION: Replication is essential for fault-tolerance. However, in in-memory systems, it is a source of high overhead. Remote direct memory access (RDMA) is attractive to create redundant copies of data, since it is low-latency and has no CPU overhead at the target. However, existing approaches still result in redundant data copying and active receivers. To ensure atomic data transfers, receivers check and apply only fully received messages. Tailwind is a zero-copy recovery-log replication protocol for scale-out in-memory databases. Tailwind is the first replication protocol that eliminates *all* CPU-driven data copying and fully bypasses target server CPUs, thus leaving backups idle. Tailwind ensures all writes are atomic by leveraging a protocol that detects incomplete RDMA transfers. Tailwind substantially improves replication throughput and response latency compared with conventional RPC-based replication. In symmetric systems where servers both serve requests and act as replicas, Tailwind also improves normal-case throughput by freeing server CPU resources for request processing. We implemented and evaluated Tailwind on RAMCloud, a low-latency in-memory storage system. Experiments show Tailwind improves RAMCloud's normal-case request processing throughput by $1.7\times$. It also cuts down writes median and 99th percentile latencies by 2x and 3x respectively.

- Contact: Gabriel Antoniu

POLARIS Project-Team

6. New Software and Platforms

6.1. Framesoc

FUNCTIONAL DESCRIPTION: Framesoc is the core software infrastructure of the SoC-Trace project. It provides a graphical user environment for execution-trace analysis, featuring interactive analysis views as Gantt charts or statistics views. It provides also a software library to store generic trace data, play with them, and build other analysis tools (e.g., Ocelotl).

- Participants: Arnaud Legrand and Jean-Marc Vincent
- Contact: Guillaume Huard
- URL: <http://soctrace-inria.github.io/framesoc/>

6.2. GameSeer

FUNCTIONAL DESCRIPTION: GameSeer is a tool for students and researchers in game theory that uses Mathematica to generate phase portraits for normal form games under a variety of (user-customizable) evolutionary dynamics. The whole point behind GameSeer is to provide a dynamic graphical interface that allows the user to employ Mathematica's vast numerical capabilities from a simple and intuitive front-end. So, even if you've never used Mathematica before, you should be able to generate fully editable and customizable portraits quickly and painlessly.

- Contact: Panayotis Mertikopoulos
- URL: <http://mescal.imag.fr/membres/panayotis.mertikopoulos/>

6.3. marmoteCore

Markov Modeling Tools and Environments - the Core

KEYWORDS: Modeling - Stochastic models - Markov model

FUNCTIONAL DESCRIPTION: marmoteCore is a C++ environment for modeling with Markov chains. It consists in a reduced set of high-level abstractions for constructing state spaces, transition structures and Markov chains (discrete-time and continuous-time). It provides the ability of constructing hierarchies of Markov models, from the most general to the particular, and equip each level with specifically optimized solution methods.

This software is developed within the ANR MARMOTE project: ANR-12-MONU-00019.

- Participants: Alain Jean-Marie, Hlib Mykhailenko, Benjamin Briot, Franck Quessette, Issam Rabhi, Jean-Marc Vincent and Jean-Michel Fourneau
- Partner: UVSQ
- Contact: Alain Jean-Marie
- Publications: [marmoteCore: a Markov Modeling Platform](#) - [marmoteCore: a software platform for Markov modeling](#)
- URL: <http://marmotecore.gforge.inria.fr/>

6.4. Moca

Memory Organisation Cartography and Analysis

KEYWORDS: High-Performance Computing - Performance analysis

- Contact: David Beniamine
- URL: <https://github.com/dbeniamine/MOCA>

6.5. Ocelotl

Multidimensional Overviews for Huge Trace Analysis

FUNCTIONAL DESCRIPTION: Ocelotl is an innovative visualization tool, which provides overviews for execution trace analysis by using a data aggregation technique. This technique enables to find anomalies in huge traces containing up to several billions of events, while keeping a fast computation time and providing a simple representation that does not overload the user.

- Participants: Arnaud Legrand and Jean-Marc Vincent
- Contact: Jean-Marc Vincent
- URL: <http://soctrace-inria.github.io/ocelotl/>

6.6. PSI

Perfect Simulator

FUNCTIONAL DESCRIPTION: Perfect simulator is a simulation software of markovian models. It is able to simulate discrete and continuous time models to provide a perfect sampling of the stationary distribution or directly a sampling of functional of this distribution by using coupling from the past. The simulation kernel is based on the CFTP algorithm, and the internal simulation of transitions on the Aliasing method.

- Contact: Jean-Marc Vincent
- URL: <http://psi.gforge.inria.fr/>

6.7. SimGrid

KEYWORDS: Large-scale Emulators - Grid Computing - Distributed Applications

SCIENTIFIC DESCRIPTION: SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The simulation engine uses algorithmic and implementation techniques toward the fast simulation of large systems on a single machine. The models are theoretically grounded and experimentally validated. The results are reproducible, enabling better scientific practices.

Its models of networks, cpus and disks are adapted to (Data)Grids, P2P, Clouds, Clusters and HPC, allowing multi-domain studies. It can be used either to simulate algorithms and prototypes of applications, or to emulate real MPI applications through the virtualization of their communication, or to formally assess algorithms and applications that can run in the framework.

The formal verification module explores all possible message interleavings in the application, searching for states violating the provided properties. We recently added the ability to assess liveness properties over arbitrary and legacy codes, thanks to a system-level introspection tool that provides a finely detailed view of the running application to the model checker. This can for example be leveraged to verify both safety or liveness properties, on arbitrary MPI code written in C/C++/Fortran.

NEWS OF THE YEAR: There were 3 major releases in 2019: Python bindings were introduced, SMPI now partially supports some of the MPI/IO functions, a new model for Wifi networks was proposed, and the API for the simulation of storage resources was completely revisited. We also pursued our efforts to improve the documentation of the software, simplified the web site, and made a lot of bug fixing and code refactoring.

- Participants: Adrien Lèbre, Arnaud Legrand, Augustin Degomme, Florence Perronnin, Frédéric Suter, Jean-Marc Vincent, Jonathan Pastor, Luka Stanic and Martin Quinson
- Partners: CNRS - ENS Rennes
- Contact: Martin Quinson
- URL: <https://simgrid.org/>

6.8. Tabarnac

Tool for Analyzing the Behavior of Applications Running on NUMA Architecture

KEYWORDS: High-Performance Computing - Performance analysis - NUMA

- Contact: David Beniamine
- URL: <https://dbeniamine.github.io/Tabarnac/>

ROMA Project-Team

6. New Software and Platforms

6.1. MUMPS

A Multifrontal Massively Parallel Solver

KEYWORDS: High-Performance Computing - Direct solvers - Finite element modelling

FUNCTIONAL DESCRIPTION: MUMPS is a software library to solve large sparse linear systems ($AX=B$) on sequential and parallel distributed memory computers. It implements a sparse direct method called the multifrontal method. It is used worldwide in academic and industrial codes, in the context numerical modeling of physical phenomena with finite elements. Its main characteristics are its numerical stability, its large number of features, its high performance and its constant evolution through research and feedback from its community of users. Examples of application fields include structural mechanics, electromagnetism, geophysics, acoustics, computational fluid dynamics. MUMPS has been developed by INPT(ENSEEIH)-IRIT, Inria, CERFACS, University of Bordeaux, CNRS and ENS Lyon. Since January 2019, it is developed and licensed by Mumps Technologies SAS.

NEWS OF THE YEAR: In June 2019, a new version of MUMPS, MUMPS 5.2.1, was released by Mumps Technologies.

- Participants: Gilles Moreau, Abdou Guermouche, Alfredo Buttari, Aurélia Fevre, Bora Uçar, Chiara Puglisi, Clément Weisbecker, Emmanuel Agullo, François-Henry Rouet, Guillaume Joslin, Jacko Koster, Jean-Yves L'Excellent, Marie Durand, Maurice Brémond, Mohamed Sid-Lakhdar, Patrick Amestoy, Philippe Combes, Stéphane Pralet, Theo Mary and Tzvetomila Slavova
- Partners: Université de Bordeaux - CNRS - CERFACS - ENS Lyon - INPT - IRIT - Université de Lyon - Université de Toulouse - LIP - Mumps Technologies SAS
- Contact: Jean-Yves L'Excellent
- URL: <http://mumps-solver.org/>

STORM Project-Team

6. New Software and Platforms

6.1. Chameleon

KEYWORDS: Runtime system - Task-based algorithm - Dense linear algebra - HPC - Task scheduling

SCIENTIFIC DESCRIPTION: Chameleon is part of the MORSE (Matrices Over Runtime Systems @ Exascale) project. The overall objective is to develop robust linear algebra libraries relying on innovative runtime systems that can fully benefit from the potential of those future large-scale complex machines.

We expect advances in three directions based first on strong and closed interactions between the runtime and numerical linear algebra communities. This initial activity will then naturally expand to more focused but still joint research in both fields.

1. Fine interaction between linear algebra and runtime systems. On parallel machines, HPC applications need to take care of data movement and consistency, which can be either explicitly managed at the level of the application itself or delegated to a runtime system. We adopt the latter approach in order to better keep up with hardware trends whose complexity is growing exponentially. One major task in this project is to define a proper interface between HPC applications and runtime systems in order to maximize productivity and expressivity. As mentioned in the next section, a widely used approach consists in abstracting the application as a DAG that the runtime system is in charge of scheduling. Scheduling such a DAG over a set of heterogeneous processing units introduces a lot of new challenges, such as predicting accurately the execution time of each type of task over each kind of unit, minimizing data transfers between memory banks, performing data prefetching, etc. Expected advances: In a nutshell, a new runtime system API will be designed to allow applications to provide scheduling hints to the runtime system and to get real-time feedback about the consequences of scheduling decisions.

2. Runtime systems. A runtime environment is an intermediate layer between the system and the application. It provides low-level functionality not provided by the system (such as scheduling or management of the heterogeneity) and high-level features (such as performance portability). In the framework of this proposal, we will work on the scalability of runtime environment. To achieve scalability it is required to avoid all centralization. Here, the main problem is the scheduling of the tasks. In many task-based runtime environments the scheduler is centralized and becomes a bottleneck as soon as too many cores are involved. It is therefore required to distribute the scheduling decision or to compute a data distribution that impose the mapping of task using, for instance the so-called “owner-compute” rule. Expected advances: We will design runtime systems that enable an efficient and scalable use of thousands of distributed multicore nodes enhanced with accelerators.

3. Linear algebra. Because of its central position in HPC and of the well understood structure of its algorithms, dense linear algebra has often pioneered new challenges that HPC had to face. Again, dense linear algebra has been in the vanguard of the new era of petascale computing with the design of new algorithms that can efficiently run on a multicore node with GPU accelerators. These algorithms are called “communication-avoiding” since they have been redesigned to limit the amount of communication between processing units (and between the different levels of memory hierarchy). They are expressed through Direct Acyclic Graphs (DAG) of fine-grained tasks that are dynamically scheduled. Expected advances: First, we plan to investigate the impact of these principles in the case of sparse applications (whose algorithms are slightly more complicated but often rely on dense kernels). Furthermore, both in the dense and sparse cases, the scalability on thousands of nodes is still limited, new numerical approaches need to be found. We will specifically design sparse hybrid direct/iterative methods that represent a promising approach.

Overall end point. The overall goal of the MORSE associate team is to enable advanced numerical algorithms to be executed on a scalable unified runtime system for exploiting the full potential of future exascale machines.

FUNCTIONAL DESCRIPTION: Chameleon is a dense linear algebra software relying on sequential task-based algorithms where sub-tasks of the overall algorithms are submitted to a Runtime system. A Runtime system such as StarPU is able to manage automatically data transfers between not shared memory area (CPUs-GPUs, distributed nodes). This kind of implementation paradigm allows to design high performing linear algebra algorithms on very different type of architecture: laptop, many-core nodes, CPUs-GPUs, multiple nodes. For example, Chameleon is able to perform a Cholesky factorization (double-precision) at 80 TFlop/s on a dense matrix of order 400 000 (i.e. 4 min 30 s).

RELEASE FUNCTIONAL DESCRIPTION: Chameleon includes the following features:

- BLAS 3, LAPACK one-sided and LAPACK norms tile algorithms - Support QUARK and StarPU runtime systems and PaRSEC since 2018 - Exploitation of homogeneous and heterogeneous platforms through the use of BLAS/LAPACK CPU kernels and cuBLAS/MAGMA CUDA kernels - Exploitation of clusters of interconnected nodes with distributed memory (using OpenMPI)

- Participants: Cédric Castagnede, Samuel Thibault, Emmanuel Agullo, Florent Pruvost and Mathieu Faverge
- Partners: Innovative Computing Laboratory (ICL) - King Abdulla University of Science and Technology - University of Colorado Denver
- Contact: Emmanuel Agullo
- URL: <https://gitlab.inria.fr/solverstack/chameleon>

6.2. hwloc

Hardware Locality

KEYWORDS: NUMA - Multicore - GPU - Affinities - Open MPI - Topology - HPC - Locality

FUNCTIONAL DESCRIPTION: Hardware Locality (hwloc) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices. It builds a widely-portable abstraction of these resources and exposes it to applications so as to help them adapt their behavior to the hardware characteristics. They may consult the hierarchy of resources, their attributes, and bind task or memory on them.

hwloc targets many types of high-performance computing applications, from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers, and multiple other parallel libraries already use hwloc.

NEWS OF THE YEAR: hwloc 2.1 brought support for modern multi-die processors and memory-side caches. It also enhanced memory locality in heterogeneous memory architecture (e.g. with non-volatile memory DIMMs). The visualization of many-core platforms was also improved by factorizing objects when many of them are identical.

- Participants: Brice Goglin and Valentin Hoyet
- Partners: Open MPI consortium - Intel - AMD - IBM
- Contact: Brice Goglin
- Publications: [hwloc: a Generic Framework for Managing Hardware Affinities in HPC Applications](#) - [Managing the Topology of Heterogeneous Cluster Nodes with Hardware Locality \(hwloc\)](#) - [A Topology-Aware Performance Monitoring Tool for Shared Resource Management in Multicore Systems](#) - [Exposing the Locality of Heterogeneous Memory Architectures to HPC Applications](#) - [Towards the Structural Modeling of the Topology of next-generation heterogeneous cluster Nodes with hwloc](#) - [On the Overhead of Topology Discovery for Locality-aware Scheduling in HPC](#) - [Memory Footprint of Locality Information on Many-Core Platforms](#) - [M&MMs: Navigating Complex Memory Spaces with hwloc](#)
- URL: <http://www.open-mpi.org/projects/hwloc/>

6.3. KaStORS

The KaStORS OpenMP Benchmark Suite

KEYWORDS: Benchmarking - HPC - Task-based algorithm - Task scheduling - OpenMP - Data parallelism

FUNCTIONAL DESCRIPTION: The KaStORS benchmarks suite has been designed to evaluate implementations of the OpenMP dependent task paradigm, introduced as part of the OpenMP 4.0 specification.

- Participants: François Broquedis, Nathalie Furmento, Olivier Aumage, Philippe Virouleau, Pierrick Brunet, Samuel Thibault and Thierry Gautier
- Contact: Thierry Gautier
- URL: <http://kastors.gforge.inria.fr/#!/index.md>

6.4. KStar

The KStar OpenMP Compiler

KEYWORDS: Source-to-source compiler - OpenMP - Task scheduling - Compilers - Data parallelism

FUNCTIONAL DESCRIPTION: The KStar software is a source-to-source OpenMP compiler for languages C and C++. The KStar compiler translates OpenMP directives and constructs into API calls from the StarPU runtime system or the XKaapi runtime system. The KStar compiler is virtually fully compliant with OpenMP 3.0 constructs. The KStar compiler supports OpenMP 4.0 dependent tasks and accelerated targets.

- Participants: Nathalie Furmento, Olivier Aumage, Philippe Virouleau and Samuel Thibault
- Contact: Olivier Aumage
- Publications: [Bridging the gap between OpenMP and task-based runtime systems for the fast multipole method](#) - [Bridging the gap between OpenMP 4.0 and native runtime systems for the fast multipole method](#) - [Evaluation of OpenMP Dependent Tasks with the KASTORS Benchmark Suite](#)
- URL: <http://kstar.gforge.inria.fr/#!/index.md>

6.5. AFF3CT

A Fast Forward Error Correction Toolbox

KEYWORDS: High-Performance Computing - Signal processing - Error Correction Code

FUNCTIONAL DESCRIPTION: AFF3CT proposes high performance Error Correction algorithms for Polar, Turbo, LDPC, RSC (Recursive Systematic Convolutional), Repetition and RA (Repeat and Accumulate) codes. These signal processing codes can be parameterized in order to optimize some given metrics, such as Bit Error Rate, Bandwidth, Latency, ...using simulation. For the designers of such signal processing chain, AFF3CT proposes also high performance building blocks so to develop new algorithms. AFF3CT compiles with many compilers and runs on Windows, Mac OS X, Linux environments and has been optimized for x86 (SSE, AVX instruction sets) and ARM architectures (NEON instruction set).

- Authors: Adrien Cassagne, Bertrand Le Gal, Camille Leroux, Denis Barthou and Olivier Aumage
- Partner: IMS
- Contact: Adrien Cassagne
- Publications: [AFF3CT: A Fast Forward Error Correction Toolbox!](#) - [AFF3CT : Un environnement de simulation pour le codage de canal](#) - [Toward High-Performance Implementation of 5G SCMA Algorithms](#) - [An Efficient, Portable and Generic Library for Successive Cancellation Decoding of Polar Codes](#) - [Beyond Gbps Turbo Decoder on Multi-Core CPUs](#) - [Energy Consumption Analysis of Software Polar Decoders on Low Power Processors](#) - [Fast and Flexible Software Polar List Decoders](#) - [Fast Simulation and Prototyping with AFF3CT](#)
- URL: <https://aff3ct.github.io/>

6.6. MORSE

KEYWORDS: High performance computing - Matrix calculation - Fast multipole method - Runtime system

FUNCTIONAL DESCRIPTION: MORSE (Matrices Over Runtime Systems @ Exascale) is a scientific project, its objectives are to solve matrix problems on complex architectures, using runtime systems. More specifically, the goal is to write codes that reach a high level of performance for all architectures. The algorithms are written independently of the architecture, and the runtime system dispatches the different computational parts to the different computing units. This methodology has been validated on three classes of problems: dense linear algebra, sparse and dense, and fast multipole methods. The corresponding codes have been incorporated into several softwares, MAGMA, Pastix and ScalFMM.

- Contact: Emmanuel Agullo
- URL: <http://icl.cs.utk.edu/morse/>

6.7. SwLoc

Software Contexts for Locality

KEYWORDS: HPC - Locality - Contexts - Multicore - GPU

FUNCTIONAL DESCRIPTION: SwLoc is a library for flexible and generic partitioning of computing resources (processors, accelerators) to be able to co-execute confined parallel regions which can rely on different runtime systems (e.g. OpenMP, Intel TBB, StarPU, etc.). With all different hypervisor strategies, It is possible to adapt dynamically the computing resources of each context, in order to match each parallel region's need as closely as possible.

- Contact: Corentin Salingue
- URL: <http://swloc.gforge.inria.fr/>

6.8. VITE

Visual Trace Explorer

KEYWORDS: Visualization - Execution trace

FUNCTIONAL DESCRIPTION: ViTE is a trace explorer. It is a tool made to visualize execution traces of large parallel programs. It supports Pajé, a trace format created by Inria Grenoble, and OTF and OTF2 formats, developed by the University of Dresden and allows the programmer a simpler way to analyse, debug and/or profile large parallel applications.

- Participant: Mathieu Faverge
- Contact: Mathieu Faverge
- URL: <http://vite.gforge.inria.fr/>

6.9. PARCOACH

PARallel Control flow Anomaly CHecker

KEYWORDS: High-Performance Computing - Program verification - Debug - MPI - OpenMP - Compilation

SCIENTIFIC DESCRIPTION: PARCOACH verifies programs in two steps. First, it statically verifies applications with a data- and control-flow analysis and outlines execution paths leading to potential deadlocks. The code is then instrumented, displaying an error and synchronously interrupting all processes if the actual scheduling leads to a deadlock situation.

FUNCTIONAL DESCRIPTION: Supercomputing plays an important role in several innovative fields, speeding up prototyping or validating scientific theories. However, supercomputers are evolving rapidly with now millions of processing units, posing the questions of their programmability. Despite the emergence of more widespread and functional parallel programming models, developing correct and effective parallel applications still remains a complex task. As current scientific applications mainly rely on the Message Passing Interface (MPI) parallel programming model, new hardware designed for Exascale with higher node-level parallelism clearly advocate for an MPI+X solutions with X a thread-based model such as OpenMP. But integrating two different programming models inside the same application can be error-prone leading to complex bugs - mostly detected unfortunately at runtime. PARallel COnTrol flow Anomaly CHecker aims at helping developers in their debugging phase.

- Participants: Emmanuelle Saillard, Denis Barthou and Pierre Huchant
- Partner: CEA
- Contact: Emmanuelle Saillard
- Publications: [Combining Static and Dynamic Validation of MPI Collective Communication - PARCOACH: Combining static and dynamic validation of MPI collective communications - Static Validation of Barriers and Worksharing Constructs in OpenMP Applications - Correctness Analysis of MPI-3 Non-Blocking Communications in PARCOACH - Static/Dynamic Validation of MPI Collective Communications in Multi-threaded Context - MPI Thread-Level Checking for MPI+OpenMP Applications - PARCOACH Extension for Hybrid Applications with Interprocedural Analysis - PARCOACH Extension for a Full-Interprocedural Collectives Verification - Multi-Valued Expression Analysis for Collective Checking](#)
- URL: <https://esaillar.github.io/PARCOACH/>

6.10. StarPU

The StarPU Runtime System

KEYWORDS: Multicore - GPU - Scheduling - HPC - Performance

SCIENTIFIC DESCRIPTION: Traditional processors have reached architectural limits which heterogeneous multicore designs and hardware specialization (eg. coprocessors, accelerators, ...) intend to address. However, exploiting such machines introduces numerous challenging issues at all levels, ranging from programming models and compilers to the design of scalable hardware solutions. The design of efficient runtime systems for these architectures is a critical issue. StarPU typically makes it much easier for high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Cell processors: rather than handling low-level issues, programmers may concentrate on algorithmic concerns. Portability is obtained by the means of a unified abstraction of the machine. StarPU offers a unified offloadable task abstraction named "codelet". Rather than rewriting the entire code, programmers can encapsulate existing functions within codelets. In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architectures (eg. one function for CUDA and one function for CPUs). StarPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine. In order to relieve programmers from the burden of explicit data transfers, a high-level data management library enforces memory coherency over the machine: before a codelet starts (eg. on an accelerator), all its data are transparently made available on the compute resource. Given its expressive interface and portable scheduling policies, StarPU obtains portable performances by efficiently (and easily) using all computing resources at the same time. StarPU also takes advantage of the heterogeneous nature of a machine, for instance by using scheduling strategies based on auto-tuned performance models.

StarPU is a task programming library for hybrid architectures

The application provides algorithms and constraints: - CPU/GPU implementations of tasks - A graph of tasks, using either the StarPU's high level GCC plugin pragmas or StarPU's rich C API

StarPU handles run-time concerns - Task dependencies - Optimized heterogeneous scheduling - Optimized data transfers and replication between main memory and discrete memories - Optimized cluster communications

Rather than handling low-level scheduling and optimizing issues, programmers can concentrate on algorithmic concerns!

FUNCTIONAL DESCRIPTION: StarPU is a runtime system that offers support for heterogeneous multicore machines. While many efforts are devoted to design efficient computation kernels for those architectures (e.g. to implement BLAS kernels on GPUs), StarPU not only takes care of offloading such kernels (and implementing data coherency across the machine), but it also makes sure the kernels are executed as efficiently as possible.

- Participants: Corentin Salingue, Andra Hugo, Benoît Lize, Cédric Augonnet, Cyril Roelandt, François Tessier, Jérôme Clet-Ortega, Ludovic Courtès, Ludovic Stordeur, Marc Sergent, Mehdi Juhoor, Nathalie Furmento, Nicolas Collin, Olivier Aumage, Pierre-André Wacrenier, Raymond Namyst, Samuel Thibault, Simon Archipoff, Xavier Lacoste, Terry Cojean, Yanis Khorsi, Philippe Virouleau, Loïc Jouans and Leo Villeveygoux
- Contact: Olivier Aumage
- Publications: [Asynchronous Task-Based Execution of the Reverse Time Migration for the Oil and Gas Industry](#) - [A Compiler Algorithm to Guide Runtime Scheduling](#) - [Achieving high-performance with a sparse direct solver on Intel KNL](#) - [Modeling Irregular Kernels of Task-based codes: Illustration with the Fast Multipole Method](#) - [Scheduling of Dense Linear Algebra Kernels on Heterogeneous Resources](#) - [Critical resources management and scheduling under StarPU](#) - [Achieving High Performance on Supercomputers with a Sequential Task-based Programming Model](#) - [Programmation of heterogeneous architectures using moldable tasks](#) - [The StarPU Runtime System at Exascale ? : Scheduling and Programming over Upcoming Machines.](#) - [A Visual Performance Analysis Framework for Task-based Parallel Applications running on Hybrid Clusters](#) - [Analyzing Dynamic Task-Based Applications on Hybrid Platforms: An Agile Scripting Approach](#) - [Detecção de Anomalias de Desempenho em Aplicações de Alto Desempenho baseadas em Tarefas em Clusters Híbridos](#) - [Resource aggregation for task-based Cholesky Factorization on top of heterogeneous machines](#) - [On Runtime Systems for Task-based Programming on Heterogeneous Platforms](#) - [Resource aggregation in task-based applications over accelerator-based multicore machines](#) - [Controlling the Memory Subscription of Distributed Applications with a Task-Based Runtime System](#) - [Exploiting Two-Level Parallelism by Aggregating Computing Resources in Task-Based Applications Over Accelerator-Based Machines](#) - [Exploiting Two-Level Parallelism by Aggregating Computing Resources in Task-Based Applications Over Accelerator-Based Machines](#) - [Achieving High Performance on Supercomputers with a Sequential Task-based Programming Model](#) - [Bridging the gap between OpenMP 4.0 and native runtime systems for the fast multipole method](#) - [Scalability of a task-based runtime system for dense linear algebra applications](#) - [Faithful Performance Prediction of a Dynamic Task-Based Runtime System for Heterogeneous Multi-Core Architectures](#) - [Towards seismic wave modeling on heterogeneous many-core architectures using task-based runtime system](#) - [Bridging the Gap between Performance and Bounds of Cholesky Factorization on Heterogeneous Platforms](#) - [Composing multiple StarPU applications over heterogeneous machines: A supervised approach](#) - [Evaluation of OpenMP Dependent Tasks with the KASTORS Benchmark Suite](#) - [A runtime approach to dynamic resource allocation for sparse direct solvers](#) - [Modeling and Simulation of a Dynamic Task-Based Runtime System for Heterogeneous Multi-Core Architectures](#) - [Toward OpenCL Automatic Multi-Device Support](#) - [Harnessing clusters of hybrid nodes with a sequential task-based programming model](#) - [Taking advantage of hybrid systems for sparse direct solvers via task-based runtimes](#) - [Modulariser les ordonnanceurs de tâches : une approche structurelle](#) - [Overview of Distributed Linear Algebra on Hybrid Nodes over the StarPU Runtime](#) - [StarPU-MPI: Task Programming over Clusters of Machines Enhanced with Accelerators](#) - [Modeling and Simulation of a Dynamic Task-Based Runtime System for Heterogeneous Multi-Core Architectures](#) -

Taking advantage of hybrid systems for sparse direct solvers via task-based runtimes - Adaptive Task Size Control on High Level Programming for GPU/CPU Work Sharing - Composing multiple StarPU applications over heterogeneous machines: a supervised approach - Implementation of FEM Application on GPU with StarPU - Le problème de la composition parallèle : une approche supervisée - Support exécutif scalable pour les architectures hybrides distribuées - SOCL: An OpenCL Implementation with Automatic Multi-Device Adaptation Support - C Language Extensions for Hybrid CPU/GPU Programming with StarPU - Programming Models and Runtime Systems for Heterogeneous Architectures - Programmation unifiée multi-accélateur OpenCL - StarPU-MPI: Task Programming over Clusters of Machines Enhanced with Accelerators - Parallelization on Heterogeneous Multicore and Multi-GPU Systems of the Fast Multipole Method for the Helmholtz Equation Using a Runtime System - High-Level Support for Pipeline Parallelism on Many-Core Architectures - Programmability and Performance Portability Aspects of Heterogeneous Multi-/Manycore Systems - Programmation des architectures hétérogènes à l'aide de tâches divisibles - StarPU: a unified platform for task scheduling on heterogeneous multicore architectures - PEPPER: Efficient and Productive Usage of Hybrid Computing Systems - The PEPPER Approach to Programmability and Performance Portability for Heterogeneous many-core Architectures - Flexible runtime support for efficient skeleton programming on hybrid systems - LU Factorization for Accelerator-based Systems - QR Factorization on a Multicore Node Enhanced with Multiple GPU Accelerators - Programmation multi-accélateurs unifiée en OpenCL - Détection optimale des coins et contours dans des bases d'images volumineuses sur architectures multicœurs hétérogènes - Association de modèles de programmation pour l'exploitation de clusters de GPUs dans le calcul intensif - Programming heterogeneous, accelerator-based multicore machines: current situation and main challenges - Scheduling Tasks over Multicore machines enhanced with accelerators: a Runtime System's Perspective - Composabilité de codes parallèles sur architectures hétérogènes - Data-Aware Task Scheduling on Multi-Accelerator based Platforms - Dynamically scheduled Cholesky factorization on multicore architectures with GPU accelerators. - StarPU: a Runtime System for Scheduling Tasks over Accelerator-Based Multicore Machines - StarPU : un support exécutif unifié pour les architectures multicœurs hétérogènes - Automatic Calibration of Performance Models on Heterogeneous Multicore Architectures - StarPU: A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures - Exploiting the Cell/BE architecture with the StarPU unified runtime system - Bridging the gap between OpenMP and task-based runtime systems for the fast multipole method - Composability of parallel codes on heterogeneous architectures - Are Static Schedules so Bad ? A Case Study on Cholesky Factorization - Scheduling of Linear Algebra Kernels on Multiple Heterogeneous Resources - Approximation Proofs of a Fast and Efficient List Scheduling Algorithm for Task-Based Runtime Systems on Multicores and GPUs - Resource aggregation for task-based Cholesky Factorization on top of modern architectures - Visual Performance Analysis of Memory Behavior in a Task-Based Runtime on Hybrid Platforms - Tolérance aux pannes dans l'exécution distribuée de graphes de tâches

- URL: <http://starpu.gforge.inria.fr/>

TADAAM Project-Team

6. New Software and Platforms

6.1. Hsplit

Hardware communicators split

KEYWORDS: MPI communication - Topology - Hardware platform

SCIENTIFIC DESCRIPTION: Hsplit is a library that implements an abstraction allowing the programmer using MPI in their parallel applications to access the underlying hardware structure through a hierarchy of communicators. Hsplit is based on the `MPI_Comm_split_type` routine and provides a new value for the `split_type` argument that specifically creates a hierarchy a subcommunicators where each new subcommunicator corresponds to a meaningful hardware level. The important point is that only the structure o the hardware is exploited and the number of levels or the levels names are not fixed so as to propose a solution independent from future hardware evolutions (such as new levels for instance). Another flavor of this `MPI_Comm_split_type` function is provided that creates a roots communicators at the same time a subcommunicator is produced, in order to ease the collective communication and/or synchronization among subcommunicators.

FUNCTIONAL DESCRIPTION: Hsplit implements an abstraction that allows the programmer using MPI in their parallel applications to access the underlying hardware structure through a hierarchy of communicators. Hsplit is based on the `MPI_Comm_split_type` routine and provides a new value for the `split_type` argument that specifically creates a hierarchy a subcommunicators where each new subcommunicator corresponds to a meaningful hardware level. The important point is that only the structure o the hardware is exploited and the number of levels or the levels names are not fixed so as to propose a solution independent from future hardware evolutions (such as new levels for instance). Another flavor of this `MPI_Comm_split_type` function is provided that creates a roots communicators at the same time a subcommunicator is produced, in order to ease the collective communication and/or synchronization among subcommunicators.

NEWS OF THE YEAR: Most of our proposal had been officially read in front of the MPI Forum at the last physical meeting in December at Albuquerque. This concerns hte guided and the unguided mode of the split function. This now has to pass two votes in the next physical meetings in 2020 to be part of the new version of the standard: MPI 4.0 that shall be ratified and released at the end of 2020. Since no other MPI library currently implements the unguided mode, Hsplit will be the only software that is currently able to provide it.

- Participants: Guillaume Mercier, Brice Goglin and Emmanuel Jeannot
- Contact: Guillaume Mercier
- Publications: [A hierarchical model to manage hardware topology in MPI applications - A Hierarchical Model to Manage Hardware Topology in MPI Applications](#)
- URL: <http://mpi-topology.gforge.inria.fr/>

6.2. hwloc

Hardware Locality

KEYWORDS: NUMA - Multicore - GPU - Affinities - Open MPI - Topology - HPC - Locality

FUNCTIONAL DESCRIPTION: Hardware Locality (hwloc) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices. It builds a widely-portable abstraction of these resources and exposes it to applications so as to help them adapt their behavior to the hardware characteristics. They may consult the hierarchy of resources, their attributes, and bind task or memory on them.

hwloc targets many types of high-performance computing applications, from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers, and multiple other parallel libraries already use hwloc.

NEWS OF THE YEAR: hwloc 2.1 brought support for modern multi-die processors and memory-side caches. It also enhanced memory locality in heterogeneous memory architecture (e.g. with non-volatile memory DIMMs). The visualization of many-core platforms was also improved by factorizing objects when many of them are identical.

- Participants: Brice Goglin and Valentin Hoyet
- Partners: Open MPI consortium - Intel - AMD - IBM
- Contact: Brice Goglin
- Publications: [hwloc: a Generic Framework for Managing Hardware Affinities in HPC Applications](#) - [Managing the Topology of Heterogeneous Cluster Nodes with Hardware Locality \(hwloc\)](#) - [A Topology-Aware Performance Monitoring Tool for Shared Resource Management in Multicore Systems](#) - [Exposing the Locality of Heterogeneous Memory Architectures to HPC Applications](#) - [Towards the Structural Modeling of the Topology of next-generation heterogeneous cluster Nodes with hwloc](#) - [On the Overhead of Topology Discovery for Locality-aware Scheduling in HPC](#) - [Memory Footprint of Locality Information on Many-Core Platforms](#) - [M&MMs: Navigating Complex Memory Spaces with hwloc](#)
- URL: <http://www.open-mpi.org/projects/hwloc/>

6.3. NetLoc

Network Locality

KEYWORDS: Topology - Locality - Distributed networks - HPC - Parallel computing - MPI communication

FUNCTIONAL DESCRIPTION: netloc (Network Locality) is a library that extends hwloc to network topology information by assembling hwloc knowledge of server internals within graphs of inter-node fabrics such as Infiniband, Intel OmniPath or Cray networks.

Netloc builds a software representation of the entire cluster so as to help applications properly place their tasks on the nodes. It may also help communication libraries optimize their strategies according to the wires and switches.

Netloc targets the same challenges as hwloc but focuses on a wider spectrum by enabling cluster-wide solutions such as process placement. It interoperates with the Scotch graph partitioner to do so.

Netloc is distributed within hwloc releases starting with hwloc 2.0.

- Participants: Brice Goglin, Clément Foyer and Cyril Bordage
- Contact: Brice Goglin
- Publications: [netloc: Towards a Comprehensive View of the HPC System Topology](#) - [Netloc: a Tool for Topology-Aware Process Mapping](#)
- URL: <http://www.open-mpi.org/projects/netloc/>

6.4. NewMadeleine

NewMadeleine: An Optimizing Communication Library for High-Performance Networks

KEYWORDS: High-performance calculation - MPI communication

FUNCTIONAL DESCRIPTION: NewMadeleine is the fourth incarnation of the Madeleine communication library. The new architecture aims at enabling the use of a much wider range of communication flow optimization techniques. Its design is entirely modular: drivers and optimization strategies are dynamically loadable software components, allowing experimentations with multiple approaches or on multiple issues with regard to processing communication flows.

The optimizing scheduler SchedOpt targets applications with irregular, multi-flow communication schemes such as found in the increasingly common application conglomerates made of multiple programming environments and coupled pieces of code, for instance. SchedOpt itself is easily extensible through the concepts of optimization strategies (what to optimize for, what the optimization goal is) expressed in terms of tactics (how to optimize to reach the optimization goal). Tactics themselves are made of basic communication flows operations such as packet merging or reordering.

The communication library is fully multi-threaded through its close integration with PIOMan. It manages concurrent communication operations from multiple libraries and from multiple threads. Its MPI implementation MadMPI fully supports the MPI_THREAD_MULTIPLE multi-threading level.

- Participants: Alexandre Denis, Clément Foyer, Nathalie Furmento, Raymond Namyst, Adrien Guibaud, Florian Reynier and Philippe Swartvagher
- Contact: Alexandre Denis
- Publications: [NewMadeleine: a Fast Communication Scheduling Engine for High Performance Networks](#) - [Ordonnancement et qualité de service pour réseaux rapides](#) - [Improving Reactivity and Communication Overlap in MPI using a Generic I/O Manager](#) - [PIOMan : un gestionnaire d'entrées-sorties générique](#) - [A multithreaded communication engine for multicore architectures](#) - [A multicore-enabled multirail communication engine](#) - [About the interactions between communication and thread scheduling in clusters of multicore machines](#) - [Scalability of the NewMadeleine Communication Library for Large Numbers of MPI Point-to-Point Requests](#) - [An analysis of the impact of multi-threading on communication performance](#) - [A scalable and generic task scheduling system for communication libraries](#) - [A Generic and High Performance Approach for Fault Tolerance in Communication Library](#) - [A High-Performance Superpipeline Protocol for InfiniBand](#) - [A sampling-based approach for communication libraries auto-tuning](#) - [High performance checksum computation for fault-tolerant MPI over InfiniBand](#) - [pioman: a Generic Framework for Asynchronous Progression and Multithreaded Communications](#) - [pioman: a pthread-based Multithreaded Communication Engine](#) - [Updating MadMPI to MPI-3: Remote Memory Access](#) - [Portage de StarPU sur la bibliothèque de communication NewMadeleine](#)
- URL: <http://pm2.gforge.inria.fr/newmadeleine/>

6.5. PaMPA

Parallel Mesh Partitioning and Adaptation

KEYWORDS: Dynamic load balancing - Unstructured heterogeneous meshes - Parallel remeshing - Subdomain decomposition - Parallel numerical solvers

SCIENTIFIC DESCRIPTION: PaMPA is a parallel library for handling, redistributing and remeshing unstructured meshes on distributed-memory architectures. PaMPA dramatically eases and speeds-up the development of parallel numerical solvers for compact schemes. It provides solver writers with a distributed mesh abstraction and an API to: - describe unstructured and possibly heterogeneous meshes, on the form of a graph of interconnected entities of different kinds (e.g. elements, faces, edges, nodes), - attach values to the mesh entities, - distribute such meshes across processing elements, with an overlap of variable width, - perform synchronous or asynchronous data exchanges of values across processing elements, - describe numerical schemes by means of iterators over mesh entities and their connected neighbors of a given kind, - redistribute meshes so as to balance computational load, - perform parallel dynamic remeshing, by applying adequately a user-provided sequential remeshing to relevant areas of the distributed mesh.

PaMPA runs concurrently multiple sequential remeshing tasks to perform dynamic parallel remeshing and redistribution of very large unstructured meshes. E.g., it can remesh a tetrahedral mesh from 43M elements to more than 1B elements on 280 Broadwell processors in 20 minutes.

FUNCTIONAL DESCRIPTION: Parallel library for handling, redistributing and remeshing unstructured, heterogeneous meshes on distributed-memory architectures. PaMPA dramatically eases and speeds-up the development of parallel numerical solvers for compact schemes.

NEWS OF THE YEAR: PaMPA has been used to remesh an industrial mesh of a helicopter turbine combustion chamber, up to more than 1 billion elements.

- Participants: Cécile Dobrzynski, Cedric Lachat and François Pellegrini
- Partners: Université de Bordeaux - CNRS - IPB
- Contact: François Pellegrini
- URL: <http://project.inria.fr/pampa/>

6.6. TopoMatch

KEYWORDS: Intensive parallel computing - High-Performance Computing - Hierarchical architecture - Placement

SCIENTIFIC DESCRIPTION: TreeMatch embeds a set of algorithms to map processors/cores in order to minimize the communication cost of the application.

Important features are : the number of processors can be greater than the number of applications processes , it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speeds) , it implements different placement algorithms that are switched according to the input size.

Some core algorithms are parallel to speed-up the execution. Optionally embeds scotch for fix-vertex mapping. enable exhaustive search if required. Several metric mapping are computed. Allow for oversubscribing of ressources. multithreaded.

TreeMatch is integrated into various software such as the Charm++ programming environment as well as in both major open-source MPI implementations: Open MPI and MPICH2.

FUNCTIONAL DESCRIPTION: TreeMatch is a library for performing process placement based on the topology of the machine and the communication pattern of the application.

- Participants: Adele Villiermet, Emmanuel Jeannot, François Tessier, Guillaume Mercier and Pierre Celor
- Partners: Université de Bordeaux - CNRS - IPB
- Contact: Emmanuel Jeannot
- URL: <http://treematch.gforge.inria.fr/>

6.7. SCOTCH

KEYWORDS: Mesh partitioning - Domain decomposition - Graph algorithmics - High-performance calculation - Sparse matrix ordering - Static mapping

FUNCTIONAL DESCRIPTION: Scotch is a graph partitioner. It helps optimise the division of a problem, by means of a graph, into a set of independent sub-problems of equivalent sizes. These sub-problems can also be solved in parallel.

RELEASE FUNCTIONAL DESCRIPTION: Version 6.0 offers many new features:

sequential graph repartitioning

sequential graph partitioning with fixed vertices

sequential graph repartitioning with fixed vertices

new, fast, direct k-way partitioning and mapping algorithms

multi-threaded, shared memory algorithms in the (formerly) sequential part of the library

exposure in the API of many centralized and distributed graph handling routines

embedded pseudo-random generator for improved reproducibility

and even more...

NEWS OF THE YEAR: In 2019, several versions of Scotch have been released, from v6.0.7 up to v6.0.9. While they are mostly bugfix updates, several new features and API routines have been added, to increase its use by third-party software, notably routines handling target topologies. Also, code quality has been improved by the addition of many tests in the continuous integration process. A new graphical system has been developed by Amaury Jacques (Inria intern, Feb.-May 2019) to display differences in result quality across versions and builds. This system has been adopted by other Inria projects.

- Participants: François Pellegrini, Sébastien Fourestier, Jun-Ho Her, Cédric Chevalier and Amaury Jacques
- Partners: Université de Bordeaux - IPB - CNRS - Region Aquitaine
- Contact: François Pellegrini
- Publications: [Process Mapping onto Complex Architectures and Partitions Thereof - Multi-criteria Graph Partitioning with Scotch](#) - [Adaptation au repartitionnement de graphes d'une méthode d'optimisation globale par diffusion](#) - [Contributions au partitionnement de graphes parallèle multi-niveaux](#) - [A parallelisable multi-level banded diffusion scheme for computing balanced partitions with smooth boundaries](#) - [PT-Scotch: A tool for efficient parallel graph ordering](#) - [Design and implementation of efficient tools for parallel partitioning and distribution of very large numerical problems](#) - [Improvement of the Efficiency of Genetic Algorithms for Scalable Parallel Graph Partitioning in a Multi-Level Framework](#) - [PT-Scotch : Un outil pour la renumérotation parallèle efficace de grands graphes dans un contexte multi-niveaux](#) - [PT-Scotch: A tool for efficient parallel graph ordering](#)
- URL: <http://www.labri.fr/~pelegrin/scotch/>

6.8. disk-revolve

KEYWORDS: Automatic differentiation - Gradients - Machine learning

FUNCTIONAL DESCRIPTION: This software provides several algorithms (Disk-Revolve, 1D-Revolve, Periodic-Disk-Revolve,...) computing the optimal checkpointing strategy when executing an adjoint chain with limited memory. The considered architecture has a level of limited memory that is free to access (writing and reading costs are negligible) and a level of unlimited memory with non-negligible access costs. The algorithms describe which data should be saved in the memory to minimize the number of re-computation during the execution.

- Authors: Guillaume Aupy and Julien Herrmann
- Contact: Julien Herrmann
- Publications: [H-Revolve: A Framework for Adjoint Computation on Synchronous Hierarchical Platforms](#) - [Periodicity in optimal hierarchical checkpointing schemes for adjoint computations](#) - [Optimal Multistage Algorithm for Adjoint Computation](#)
- URL: <https://gitlab.inria.fr/adjoint-computation/disk-revolve-public>

DIVERSE Project-Team

5. New Software and Platforms

5.1. amiunique

KEYWORDS: Privacy - Browser fingerprinting

SCIENTIFIC DESCRIPTION: The amiunique web site has been deployed in the context of the DiverSE's research activities on browser fingerprinting and how software diversity can be leveraged in order to mitigate the impact of fingerprinting on the privacy of users. The construction of a dataset of genuine fingerprints is essential to understand in detail how browser fingerprints can serve as unique identifiers and hence what should be modified in order to mitigate its impact privacy. This dataset also supports the large-scale investigation of the impact of web technology advances on fingerprinting. For example, we can analyze in detail the impact of the HTML5 canvas element or the behavior of fingerprinting on mobile devices.

The whole source code of amiunique is open source and is distributed under the terms of the MIT license.

- Panopticlick <https://panopticlick.eff.org/>
- BrowserSpy <http://browserspy.dk/>

Main innovative features:

- canvas fingerprinting
- WebGL fingerprinting
- advanced JS features (platform, DNT, etc.)

Impact: The website has been showcased in several professional forums in 2014 and 2015 (Open World Forum 2014, FOSSA'14, FIC'15, ICT'15) and it has been visited by more than 100,000 unique visitors in one year.

FUNCTIONAL DESCRIPTION: This web site aims at informing visitors about browser fingerprinting and possible tools to mitigate its effect, as well as at collecting data about the fingerprints that can be found on the web. It collects browser fingerprints with the explicit agreement of the users (they have to click on a button on the home page). Fingerprints are composed of 17 attributes, which include regular HTTP headers as well as the most recent state of the art techniques (canvas fingerprinting, WebGL information).

- Participants: Benoit Baudry and Pierre Laperdrix
- Partner: INSA Rennes
- Contact: Benoit Baudry
- URL: <https://amiunique.org/>

5.2. FAMILIAR

KEYWORDS: Software line product - Configators - Customisation

SCIENTIFIC DESCRIPTION: FAMILIAR (for FeAture Model scrIpt Language for manipulation and Automatic Reasoning) is a language for importing, exporting, composing, decomposing, editing, configuring, computing "diffs", refactoring, reverse engineering, testing, and reasoning about (multiple) feature models. All these operations can be combined to realize complex variability management tasks. A comprehensive environment is proposed as well as integration facilities with the Java ecosystem.

FUNCTIONAL DESCRIPTION: Familiar is an environment for large-scale product customisation. From a model of product features (options, parameters, etc.), Familiar can automatically generate several million variants. These variants can take many forms: software, a graphical interface, a video sequence or even a manufactured product (3D printing). Familiar is particularly well suited for developing web configurators (for ordering customised products online), for providing online comparison tools and also for engineering any family of embedded or software-based products.

- Participants: Aymeric Hervieu, Benoit Baudry, Didier Vojtisek, Edward Mauricio Alferez Salinas, Guillaume Bécan, Joao Bosco Ferreira-Filho, Julien Richard-Foy, Mathieu Acher, Olivier Barais and Sana Ben Nasr
- Contact: Mathieu Acher
- URL: <http://familiar-project.github.com>

5.3. GEMOC Studio

KEYWORDS: DSL - Language workbench - Model debugging

SCIENTIFIC DESCRIPTION: The language workbench put together the following tools seamlessly integrated to the Eclipse Modeling Framework (EMF):

- Melange, a tool-supported meta-language to modularly define executable modeling languages with execution functions and data, and to extend (EMF-based) existing modeling languages.
- MoCCML, a tool-supported meta-language dedicated to the specification of a Model of Concurrency and Communication (MoCC) and its mapping to a specific abstract syntax and associated execution functions of a modeling language.
- GEL, a tool-supported meta-language dedicated to the specification of the protocol between the execution functions and the MoCC to support the feedback of the data as well as the callback of other expected execution functions.
- BCOoL, a tool-supported meta-language dedicated to the specification of language coordination patterns to automatically coordinates the execution of, possibly heterogeneous, models.
- Sirius Animator, an extension to the model editor designer Sirius to create graphical animators for executable modeling languages.

FUNCTIONAL DESCRIPTION: The GEMOC Studio is an eclipse package that contains components supporting the GEMOC methodology for building and composing executable Domain-Specific Modeling Languages (DSMLs). It includes the two workbenches: The GEMOC Language Workbench: intended to be used by language designers (aka domain experts), it allows to build and compose new executable DSMLs. The GEMOC Modeling Workbench: intended to be used by domain designersto create, execute and coordinate models conforming to executable DSMLs. The different concerns of a DSML, as defined with the tools of the language workbench, are automatically deployed into the modeling workbench. They parametrize a generic execution framework that provide various generic services such as graphical animation, debugging tools, trace and event managers, timeline, etc.

- Participants: Didier Vojtisek, Dorian Leroy, Erwan Bousse, Fabien Coulon and Julien DeAntoni
- Partners: IRIT - ENSTA - I3S - OBEO - Thales TRT
- Contact: Benoît Combemale
- URL: <http://gemoc.org/studio.html>

5.4. Kevoree

KEYWORDS: M2M - Dynamic components - Iot - Heterogeneity - Smart home - Cloud - Software architecture - Dynamic deployment

SCIENTIFIC DESCRIPTION: Kevoree is an open-source models@runtime platform (<http://www.kevoree.org>) to properly support the dynamic adaptation of distributed systems. Models@runtime basically pushes the idea of reflection [132] one step further by considering the reflection layer as a real model that can be uncoupled from the running architecture (e.g. for reasoning, validation, and simulation purposes) and later automatically resynchronized with its running instance.

Kevoree has been influenced by previous work that we carried out in the DiVA project [132] and the Entimid project [135]. With Kevoree we push our vision of models@runtime [131] farther. In particular, Kevoree provides a proper support for distributed models@runtime. To this aim we introduced the Node concept to model the infrastructure topology and the Group concept to model semantics of inter node communication during synchronization of the reflection model among nodes. Kevoree includes a Channel concept to allow for multiple communication semantics between remoteComponents deployed on heterogeneous nodes. All Kevoree concepts (Component, Channel, Node, Group) obey the object type design pattern to separate deployment artifacts from running artifacts. Kevoree supports multiple kinds of very different execution node technology (e.g. Java, Android, MiniCloud, FreeBSD, Arduino, ...).

Kevoree is distributed under the terms of the LGPL open source license.

Main competitors:

- the Fractal/Frascati eco-system (<http://frascati.ow2.org/doc/1.4/frascati-userguide.html>).
- SpringSource Dynamic Module (<http://spring.io/>)
- GCM-Proactive (<http://proactive.inria.fr/>)
- OSGi (<http://www.osgi.org>)
- Chef
- Vagran (<http://vagrantup.com/>)

Main innovative features:

- distributed models@runtime platform (with a distributed reflection model and an extensible models@runtime dissemination set of strategies).
- Support for heterogeneous node type (from Cyber Physical System with few resources until cloud computing infrastructure).
- Fully automated provisioning model to correctly deploy software modules and their dependencies.
- Communication and concurrency access between software modules expressed at the model level (not in the module implementation).

FUNCTIONAL DESCRIPTION: Kevoree is an open-source models@runtime platform to properly support the dynamic adaptation of distributed systems. Models@runtime basically pushes the idea of reflection one step further by considering the reflection layer as a real model that can be uncoupled from the running architecture (e.g. for reasoning, validation, and simulation purposes) and later automatically resynchronized with its running instance.

- Participants: Aymeric Hervieu, Benoit Baudry, Francisco-Javier Acosta Padilla, Inti Gonzalez Herrera, Ivan Paez Anaya, Jacky Bourgeois, Jean Emile Dartois, Johann Bourcier, Manuel Leduc, Maxime Tricoire, Mohamed Boussaa, Noël Plouzeau and Olivier Barais
- Contact: Olivier Barais
- URL: <http://kevoree.org/>

5.5. Melange

KEYWORDS: Model-driven engineering - Meta model - MDE - DSL - Model-driven software engineering - Dedicated langage - Language workbench - Meta-modélisation - Modeling language - Meta-modeling

SCIENTIFIC DESCRIPTION: Melange is a follow-up of the executable metamodeling language Kermeta, which provides a tool-supported dedicated meta-language to safely assemble language modules, customize them and produce new DSMLs. Melange provides specific constructs to assemble together various abstract syntax and operational semantics artifacts into a DSML. DSMLs can then be used as first class entities to be reused, extended, restricted or adapted into other DSMLs. Melange relies on a particular model-oriented type system that provides model polymorphism and language substitutability, i.e. the possibility to manipulate a model through different interfaces and to define generic transformations that can be invoked on models written using different DSLs. Newly produced DSMLs are correct by construction, ready for production (i.e., the result can be deployed and used as-is), and reusable in a new assembly.

Melange is tightly integrated with the Eclipse Modeling Framework ecosystem and relies on the meta-language Ecore for the definition of the abstract syntax of DSLs. Executable meta-modeling is supported by weaving operational semantics defined with Xtend. Designers can thus easily design an interpreter for their DSL in a non-intrusive way. Melange is bundled as a set of Eclipse plug-ins.

FUNCTIONAL DESCRIPTION: Melange is a language workbench which helps language engineers to mashup their various language concerns as language design choices, to manage their variability, and support their reuse. It provides a modular and reusable approach for customizing, assembling and integrating DSMLs specifications and implementations.

- Participants: Arnaud Blouin, Benoît Combemale, David Mendez Acuna, Didier Vojtisek, Dorian Leroy, Erwan Bousse, Fabien Coulon, Jean-Marc Jézéquel, Olivier Barais and Thomas Degueule
- Contact: Benoît Combemale
- URL: <http://melange-lang.org>

5.6. DSpot

KEYWORDS: Software testing - Test amplification

FUNCTIONAL DESCRIPTION: DSpot is a tool that generates missing assertions in JUnit tests. DSpot takes as input a Java project with an existing test suite. As output, DSpot outputs new test cases on console. DSpot supports Java projects built with Maven and Gradle

- Participants: Benoit Baudry, Martin Monperrus and Benjamin Danglot
- Partner: KTH Royal Institute of Technology
- Contact: Benjamin Danglot
- URL: <https://github.com/STAMP-project/dspot>

5.7. ALE

Action Language for Ecore

KEYWORDS: Meta-modeling - Executable DSML

FUNCTIONAL DESCRIPTION: Main features of ALE include:

- Executable metamodeling: Re-open existing EClasses to insert new methods with their implementations
- Metamodel extension: The very same mechanism can be used to extend existing Ecore metamodels and insert new features (eg. attributes) in a non-intrusive way
- Interpreted: No need to deploy Eclipse plugins, just run the behavior on a model directly in your modeling environment
- Extensible: If ALE doesn't fit your needs, register Java classes as services and invoke them inside your implementations of EOperations.
- Partner: OBEO
- Contact: Benoît Combemale
- URL: <http://gemoc.org/ale-lang/>

5.8. InspectorGidget

KEYWORDS: Static analysis - Software testing - User Interfaces

FUNCTIONAL DESCRIPTION: InspectorGidget is a static code analysing tool. InspectorGidget analyses UI (user interface/interaction) code of a software system to extract high level information and metrics. InspectorGidget also finds bad UI coding practices, such as Blob listener instances. InspectorGidget analyses Java code.

- Participants: Arnaud Blouin and Benoit Baudry
- Contact: Arnaud Blouin
- Publications: [hal-01499106v5](#) - [hal-01308625v2](#)
- URL: <https://github.com/diverse-project/InspectorGidget>

5.9. Descartes

KEYWORDS: Software testing - Mutation analysis

FUNCTIONAL DESCRIPTION: Descartes evaluates the capability of your test suite to detect bugs using extreme mutation testing.

Descartes is a mutation engine plugin for PIT which implements extreme mutation operators as proposed in the paper *Will my tests tell me if I break this code?*.

- Participants: Oscar Luis Vera Perez, Benjamin Danglot, Benoit Baudry and Martin Monperrus
- Partner: KTH Royal Institute of Technology
- Contact: Benoit Baudry
- Publications: [Descartes: a PITest engine to detect pseudo-tested methods - Tool Demonstration - A Comprehensive Study of Pseudo-tested Methods](#)
- URL: <https://github.com/STAMP-project/pitest-descartes>

5.10. PitMP

PIT for Multi-module Project

KEYWORDS: Mutation analysis - Mutation testing - Java - JUnit - Maven

FUNCTIONAL DESCRIPTION: PIT and Descartes are mutation testing systems for Java applications, which allows you to verify if your test suites can detect possible bugs, and so to evaluate the quality of your test suites. They evaluate the capability of your test suite to detect bugs using mutation testing (PIT) or extreme mutation testing (Descartes). Mutation testing does it by introducing small changes or faults into the original program. These modified versions are called mutants. A good test suite should be able to kill or detect a mutant. Traditional mutation testing works at the instruction level, e.g., replacing ">" by "<=", so the number of generated mutants is huge, as the time required to check the entire test suite. That's why Extreme Mutation strategy appeared. In Extreme Mutation testing, the whole body of a method under test is removed. Descartes is a mutation engine plugin for PIT which implements extreme mutation operators. Both provide reports combining, line coverage, mutation score and list of weaknesses in the source.

- Partners: CSQE - KTH Royal Institute of Technology - ENGINEERING
- Contact: Caroline Landry
- URL: <https://github.com/STAMP-project/pitmp-maven-plugin>

EASE Project-Team

5. New Software and Platforms

5.1. THEGAME

KEYWORD: Contextual service

SCIENTIFIC DESCRIPTION: Context-aware applications have to sense the environment in order to adapt themselves and provide with contextual services. This is the case of Smart Homes equipped with sensors and augmented appliances. However, sensors can be numerous, heterogeneous and unreliable. Thus the data fusion is complex and requires a solid theory to handle those problems. The aim of the data fusion, in our case, is to compute small pieces of context we call context attributes. Those context attributes are diverse and could be for example the presence in a room, the number of people in a room or even that someone may be sleeping in a room. For this purpose, we developed an implementation of the belief functions theory (BFT). THE GAME (THEory of Evidence in a lanGuage Adapted for Many Embedded systems) is made of a set of C-Libraries. It provides the basics of belief functions theory, computations are optimized for an embedded environment (binary representation of sets, conditional compilation and diverse algorithmic optimizations).

THE GAME is published under apache licence (<https://github.com/bpietropaoli/THEGAME/>). It is maintained and experimented by Aurélien Richez within a sensor network platform developed by TACOMA since June 2013.

FUNCTIONAL DESCRIPTION: THEGAME is a set of software services for detecting different types of situation in a building (presence in a room, activity level, etc.) based on a set of raw data sourced from all sorts of sensors. Written in C or Java, it can be integrated in an embedded computer: tablet, smartphone, box, etc., and can be connected to different sensor networks. It can be used to implement context-aware services: for example, to alert the user if s/he forgets to close a window when leaving the building, or to turn off the heating in an empty room, etc.

- Participants: Aurélien Richez and Bastien Pietropaoli
- Contact: Frédéric Weis
- URL: <https://github.com/bpietropaoli/THEGAME/>

5.2. Platform Pervasive_RFID

SCIENTIFIC DESCRIPTION

The RFID experiment testbed has been designed and deployed in collaboration with IETR (see Figure 2). This system allows both interactive testing as well as long running experiments of RFID reading protocols. It comprises a software platform allowing fine control over all dynamic aspects influencing RFID readings: movements for target and antenna, RFID reader configuration, and smart antenna configuration (diversity and power control).

KEYWORDS: Composite objects - RFID

- Participants: Paul Couderc and Alexis Girard (Univ. Rennes 1)
- Partner: Univ. Rennes 1 (IETR - lab bringing together researchers in the electronics and telecommunications)
- Contact: Paul Couderc

5.3. ISO/IEC 15118-2 Open source Implementation

SCIENTIFIC DESCRIPTION

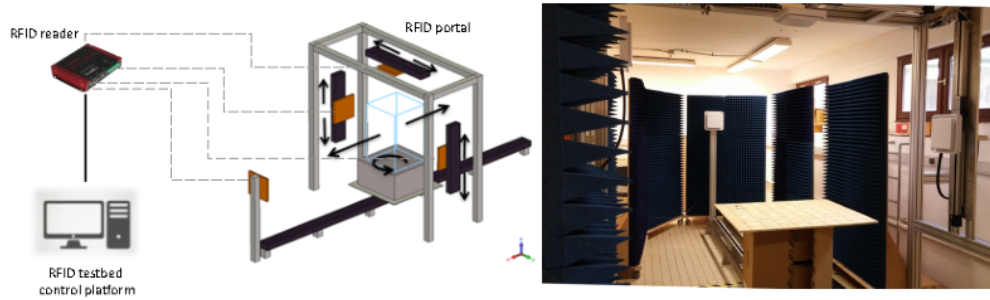


Figure 2. RFID testbed

The ISO/IEC 15118 standard, named "Road vehicles – Vehicle-to-Grid Communication Interface", defines how an electric vehicle and a charging station should communicate. It enables the Smart Charging of electric vehicles by allowing them to plan their charging sessions. As we want to be able to manage the charge of electric vehicles in our micro Smart Grid systems, we decided to implement the protocol defined by this standard. The goal is also to participate actively in the design of the new version of this protocol. During a charging session the charging station provides the vehicle with the status of the electric power grid. The vehicle is then able to plan its charging session accordingly. It sends back its charge plan to the charging station, so that the Smart Grid is aware of it. The protocol also provides security and authentication features.

This software platform was implemented onto small PCs, and was used to control the charge in a small and portable demonstration platform, to demonstrate how it is possible to interconnect this high level decision and communication software with low level components, such as a Battery Management System (BMS), and a battery charger. In 2016, in the context of the Greenfeed project our software has been demonstrated to control the charge of the electric vehicle during the final demonstration of the project. The integration work has been done in collaboration with VeDeCom⁰.

KEYWORDS: Smart Grid - Intelligent Transport System

- Partner: IMT Atlantique
- Contact: Jean-Marie Bonnin

⁰<http://www.vedecom.fr/>

FOCUS Project-Team

6. New Software and Platforms

6.1. HoCA

Higher-Order Complexity Analysis

KEYWORDS: Ocaml - Verification - Runtime Complexity Analysis

SCIENTIFIC DESCRIPTION: Over the last decade, various tools for the static analysis of resource properties of programs have emerged. In particular, the rewriting community has recently developed several tools for the time complexity analysis of term rewrite systems. These tools have matured and are nowadays able to treat non-trivial programs, in a fully automatic setting. However, none of these automatic complexity analysers can deal with higher-order functions, a pervasive feature of functional programs. HoCA (Higher-Order Complexity Analyser) overcomes this limitation by translating higher-order programs – in the form of side-effect free OCaml programs - into equivalent first-order rewrite systems. At the heart of our tool lies Reynold’s defunctionalization technique. Defunctionalization however is not enough. Resulting programs have a recursive structure too complicated to be analysed automatically in all but trivial cases. To overcome this issue, HoCA integrates a handful of well established program transformation techniques, noteworthy dead-code elimination, inlining, instantiation and uncurrying. A complexity bound on the resulting first-order program can be relayed back reliably to the higher-order program of interest. A detailed description of HoCA is available on <http://arxiv.org/abs/1506.05043>.

FUNCTIONAL DESCRIPTION: HoCA is an abbreviation for Higher-Order Complexity Analysis, and is meant as a laboratory for the automated complexity analysis of higher-order functional programs. Currently, HoCA consists of one executable `pcf2trs` which translates a pure subset of OCaml to term rewrite systems, in a complexity reflecting manner. As a first step, HoCA desugars the given program to a variation of Plotkin’s PCF with data-constructors. Via Reynold’s defunctionalization, the PCF program is turned into an applicative term rewrite system (ATRS for short), call-by-value reductions of the PCF program are simulated by the ATRS step-by-step, on the ATRS, and various complexity reflecting transformations are performed: inlining, dead-code-elimination, instantiation of higher-order variables through a call-flow-analysis and finally uncurrying. This results finally in a first-order rewrite system, whose runtime-complexity reflects the complexity of the initial program, asymptotically.

- Participants: Martin Avanzini and Ugo Dal Lago
- Contact: Ugo Dal Lago
- URL: <http://cbr.uibk.ac.at/tools/hoca/>

6.2. JOLIE

Java Orchestration Language Interpreter Engine

KEYWORD: Microservices

SCIENTIFIC DESCRIPTION: Jolie enforces a strict separation of concerns between behaviour, describing the logic of the application, and deployment, describing the communication capabilities. The behaviour is defined using the typical constructs of structured sequential programming, communication primitives, and operators to deal with concurrency (parallel composition and input choice). Jolie communication primitives comprise two modalities of interaction typical of Service-Oriented Architectures (SOAs), namely one-way (sends an asynchronous message) and request-response (sends a message and waits for an answer). A main feature of the Jolie language is that it allows one to switch among many communication media and data protocols in a simple, uniform way. Since it targets the field of SOAs, Jolie supports the main communication media (TCP/IP sockets, Bluetooth L2CAP, Java RMI, and Unix local sockets) and data protocols (HTTP, JSON-RPC, XML-RPC, SOAP and their respective SSL versions) from this area.

FUNCTIONAL DESCRIPTION: Jolie is a language for programming service-oriented and microservice applications. It directly supports service-oriented abstractions such as service, port, and session. Jolie allows to program a service behaviour, possibly obtained by composing existing services, and supports the main communication protocols and data formats used in service-oriented architectures. Differently from other service-oriented programming languages such as WS-BPEL, Jolie is based on a user-friendly Java-like syntax (more readable than the verbose XML syntax of WS-BPEL). Moreover, the kernel of Jolie is equipped with a formal operational semantics. Jolie is used to provide proof of concepts around Focus activities.

RELEASE FUNCTIONAL DESCRIPTION: There are many fixes to the HTTP extension, improvements to the embedding engine for Javascript programs, and improvements to the support tools `jolie2java` and `wsdl2jolie`.

NEWS OF THE YEAR: During 2019 the Jolie project saw three major actions.

The first action regards the build system used for the development of the language, which has been transitioned to Maven, the main build automation tool used for Java projects. The move to Maven is dictated by two needs. The first is to streamline the development and release processes of Jolie, as Maven greatly helps in obtaining, updating, and managing library dependencies. The second necessity addressed by Maven is helping in partitioning the many sub-projects that constitute the Jolie codebase, reducing development and testing times. Having Jolie as a Maven project also helps in providing Jolie sub-components (as Maven libraries) to other projects. Finally, the move to Maven is set within a larger effort to expedite the inclusion in the main Jolie development branch of contributions by new members of its growing community.

The second action regards the transition to Netty as a common framework to support communication protocols and data formats in Jolie. Netty is a widely-adopted Java framework for the development of network applications, and it was used in 2018 to successfully support several IoT communication protocols and data formats in a Jolie spin-off project, called JIoT. The work in 2019 integrated into the Jolie codebase the protocols and data format developed within the JIoT project and pushed towards the integration of the Netty development branch into the main branch of the Jolie project (i.e., re-implementing using Netty the many protocol and data-formats already supported by Jolie). The Netty development branch is currently in a beta phase and it is subject to thorough in-production tests, to ensure consistent behaviour with the previous implementation.

The third action regards the development and support for a new official IDE for Jolie. Hence, along with the ones already existing for the Atom and Sublime Text text editors, Jolie developers can use the Jolie plugin (based on the Language Server Protocol) for the Visual Studio Code text editor to obtain syntax highlighting, documentation aids, file navigation, syntax checking, semantic checking, and quick-run shortcuts for their Jolie programs.

In addition to the above actions, in 2019 Jolie transitioned through three minor releases and a major one, from 1.7.1 to 1.8.2. The minor releases mainly fixed bugs, improved performance, and included new protocol/data-format functionalities. The major release included a slim-down of the notation for the composition of statements, types definitions, and tree structures, for a terser codebase. Upgrades to 1.8.2 also introduced: timeouts for solicit-response invocations to handle the interruption of long-standing requests, more user-friendly messages from the Jolie interpreter, including easier-to-parse errors and the pretty-printing of data structures, for a more effective development and debugging experience.

In 2019 Jolie also saw the development of a new Jolie library, called TQuery, which is a query framework integrated into the Jolie language for the data handling/querying of Jolie trees. Tquery is based on a tree-based instantiation (language and semantics) of MQuery, a sound variant of the Aggregation Framework, the query language of the most popular document-oriented database: MongoDB. Usage scenarios for Tquery are (but not limited to) eHealth, the Internet-of-Things, and Edge Computing, where data should be handled in an ephemeral way, i.e., in a real-time manner but with the constraint that data shall not persist in the system.

- Participants: Claudio Guidi, Fabrizio Montesi, Maurizio Gabbrielli, Saverio Giallorenzo and Ivan Lanese
- Contact: Fabrizio Montesi
- URL: <http://www.jolie-lang.org/>

6.3. NightSplitter

KEYWORD: Constraint-based programming

FUNCTIONAL DESCRIPTION: Nightsplitter deals with the group preference optimization problem. We propose to split users into subgroups trying to optimize members' satisfaction as much as possible. In a large city with a huge volume of activity information, designing subgroup activities and avoiding time conflict is a challenging task. Currently, the Demo is available only for restaurant and movie activities in the city of Paris.

- Contact: Tong Liu
- URL: <http://cs.unibo.it/t.liu/nightsplitter/>

6.4. AIOCJ

Adaptive Interaction-Oriented Choreographies in Jolie

KEYWORD: Dynamic adaptation

SCIENTIFIC DESCRIPTION: AIOCJ is an open-source choreographic programming language for developing adaptive systems. It allows one to describe a full distributed system as a unique choreographic program and to generate code for each role avoiding by construction errors such as deadlocks. Furthermore, it supports dynamic adaptation of the distributed system via adaptation rules.

FUNCTIONAL DESCRIPTION: AIOCJ is a framework for programming adaptive distributed systems based on message passing. AIOCJ comes as a plugin for Eclipse, AIOCJ-ecl, allowing to edit descriptions of distributed systems written as adaptive interaction-oriented choreographies (AIOC). From interaction-oriented choreographies the description of single participants can be automatically derived. Adaptation is specified by rules allowing one to replace predetermined parts of the AIOC with a new behaviour. A suitable protocol ensures that all the participants are updated in a coordinated way. As a result, the distributed system follows the specification given by the AIOC under all changing sets of adaptation rules and environment conditions. In particular, the system is always deadlock free. AIOCJ can interact with external services, seen as functions, by specifying their URL and the protocol they support (HTTP, SOAP, ...). Deadlock-freedom guarantees of the application are preserved provided that those services do not block.

NEWS OF THE YEAR: In 2019 we performed a major upgrade to AIOCJ: the possibility to introduce new roles, absent from a running choreography, within a given adaptation rule. The inclusion of new roles is supported by a slight, incremental change in the AIOCJ syntax and by a new component of the AIOCJ runtime environment.

- Participants: Ivan Lanese, Jacopo Mauro, Maurizio Gabbrielli, Mila Dalla Preda and Saverio Giallorenzo
- Contact: Saverio Giallorenzo
- URL: <http://www.cs.unibo.it/projects/jolie/aioj.html>

6.5. CauDEr

Causal-consistent Debugger for Erlang

KEYWORDS: Debug - Reversible computing

SCIENTIFIC DESCRIPTION: The CauDEr reversible debugger is based on the theory of causal-consistent reversibility, which states that any action can be undone provided that its consequences, if any, are undone beforehand. This theory relies on a causal semantic for the target language, and can be used even if different processes have different notions of time

FUNCTIONAL DESCRIPTION: CauDEr is a debugger allowing one to explore the execution of concurrent Erlang programs both forward and backward. Notably, when going backward, any action can be undone provided that its consequences, if any, are undone beforehand. The debugger also provides commands to automatically find relevant past actions (e.g., send of a given message) and undo them, including their consequences. Forward computation can be driven by a log taken from a computation in the standard Erlang/OTP environment. An action in the log can be selected and replayed together with all and only its causes. The debugger enables one to find a bug by following the causality links from the visible misbehaviour to the bug. The debugger takes an Erlang program but debugging is done on its translation into Core Erlang.

NEWS OF THE YEAR: Work in 2019 consisted in maintenance, bug fixing and some minor refinements, in particular on the logging part.

- Partner: Universitat Politècnica de València
- Contact: Ivan Lanese
- URL: <https://github.com/mistupv/cauder>

6.6. SUNNY-AS

SUNNY FOR ALGORITHM SELECTION

KEYWORDS: Optimisation - Machine learning

FUNCTIONAL DESCRIPTION: SUNNY-AS is a portfolio solver derived from SUNNY-CP for Algorithm Selection Problems (ASLIB). The goal of SUNNY-AS is to provide a flexible, configurable, and usable portfolio solver that can be set up and executed just like a regular individual solver.

- Contact: Tong Liu
- URL: <https://github.com/lteu/oasc>

6.7. eco-imp

Expected Cost Analysis for Imperative Programs

KEYWORDS: Software Verification - Automation - Runtime Complexity Analysis - Randomized algorithms

FUNCTIONAL DESCRIPTION: Eco-imp is a cost analyser for probabilistic and non-deterministic imperative programs. Particularly, it features dedicated support for sampling from distributions, and can thereby accurately reason about the average case complexity of randomized algorithms, in a fully automatic fashion. The tool is based on an adaption of the ert-calculus of Kaminski et al., extended to the more general setting of cost analysis where the programmer is free to specify a (non-uniform) cost measure on programs. The main distinctive feature of eco-imp, though, is the combination of this calculus with an expected value analysis. This provides the glue to analyse program components in complete independence, that is, the analysis is modular and thus scalable. As a consequence, confirmed by our experiments, eco-imp runs on average orders of magnitude faster than comparable tools: execution times of several seconds become milliseconds.

- Contact: Martin Avanzini
- URL: <http://www-sop.inria.fr/members/Martin.Avanzini/software/eco-imp/>

INDES Project-Team

5. New Software and Platforms

5.1. Bigloo

KEYWORD: Compilers

FUNCTIONAL DESCRIPTION: Bigloo is a Scheme implementation devoted to one goal: enabling Scheme based programming style where C(++) is usually required. Bigloo attempts to make Scheme practical by offering features usually presented by traditional programming languages but not offered by Scheme and functional programming. Bigloo compiles Scheme modules. It delivers small and fast stand alone binary executables. Bigloo enables full connections between Scheme and C programs, between Scheme and Java programs.

RELEASE FUNCTIONAL DESCRIPTION: modification of the object system (language design and implementation), new APIs (alsa, flac, mpg123, avahi, csv parsing), new library functions (UDP support), new regular expressions support, new garbage collector (Boehm's collection 7.3alpha1).

- Participant: Manuel Serrano
- Contact: Manuel Serrano
- URL: <http://www-sop.inria.fr/teams/indes/fp/Bigloo/>

5.2. Hop

KEYWORDS: Programming language - Multimedia - Iot - Web 2.0 - Functional programming

SCIENTIFIC DESCRIPTION: The Hop programming environment consists in a web broker that intuitively combines in a single architecture a web server and a web proxy. The broker embeds a Hop interpreter for executing server-side code and a Hop client-side compiler for generating the code that will get executed by the client.

An important effort is devoted to providing Hop with a realistic and efficient implementation. The Hop implementation is validated against web applications that are used on a daily-basis. In particular, we have developed Hop applications for authoring and projecting slides, editing calendars, reading RSS streams, or managing blogs.

FUNCTIONAL DESCRIPTION: Multitier web programming language and runtime environment.

- Participant: Manuel Serrano
- Contact: Manuel Serrano
- URL: <http://hop.inria.fr>

5.3. IFJS

Information Flow monitor inlining for JavaScript

KEYWORD: Cybersecurity

FUNCTIONAL DESCRIPTION: The IFJS compiler is applied to JavaScript code. The compiler generates JavaScript code instrumented with checks to secure code. The compiler takes into account special features of JavaScript such as implicit type coercions and programs that actively try to bypass the inlined enforcement mechanisms. The compiler guarantees that third-party programs cannot (1) access the compiler internal state by randomizing the names of the resources through which it is accessed and (2) change the behaviour of native functions that are used by the enforcement mechanisms inlined in the compiled code.

- Contact: Tamara Rezk
- URL: <http://www-sop.inria.fr/indes/ifJS/>

5.4. HipHop.js

KEYWORDS: Web 2.0 - Synchronous Language - Programming language

FUNCTIONAL DESCRIPTION: HipHop.js is an Hop.js DLS for orchestrating web applications. HipHop.js helps programming and maintaining Web applications where the orchestration of asynchronous tasks is complex.

- Contact: Manuel Serrano
- URL: <http://hop-dev.inria.fr/hiphop>

5.5. Server-Side Protection against Third Party Web Tracking

KEYWORDS: Privacy - Web Application - Web - Architecture - Security by design - Program rewriting techniques

FUNCTIONAL DESCRIPTION: We present a new web application architecture that allows web developers to gain control over certain types of third party content. In the traditional web application architecture, a web application developer has no control over third party content. This allows the exchange of tracking information between the browser and the third party content provider.

To prevent this, our solution is based on the automatic rewriting of the web application in such a way that the third party requests are redirected to a trusted third party server, called the Middle Party Server. It may be either controlled by a trusted party, or by a main site owner and automatically eliminates third-party tracking cookies and other technologies that may be exchanged by the browser and third party server

- Contact: Francis Dolière Some
- URL: <http://www-sop.inria.fr/members/Dolierre.Some/essos/>

5.6. webstats

Webstats

KEYWORDS: Web Usage Mining - Statistic analysis - Security

FUNCTIONAL DESCRIPTION: The goal of this tool is to perform a large-scale monthly crawl of the top Alexa sites, collecting both inline scripts (written by web developers) and remote scripts, and establishing the popularity of remote scripts (such as Google Analytics and jQuery). With this data, we establish whether the collected scripts are actually written in a subset of JavaScript by analyzing the different constructs used in those scripts. Finally, we collect and analyze the HTTP headers of the different sites visited, and provide statistics about the usage of HTTPOnly and Secure cookies, and the Content Security Policy in top sites.

- Contact: Francis Dolière Some
- URL: <https://webstats.inria.fr>

5.7. Skini

Platform for creation and execution for audience participative music

KEYWORDS: Music - Interaction - Web Application - Synchronous Language

FUNCTIONAL DESCRIPTION: Skini is a platform form designing et performing collaborative music. It is based on two musical concept: pattern and orchestration. The orchestration is design using HipHop.js.

RELEASE FUNCTIONAL DESCRIPTION: Can be use for performance and création.

- Contact: Bertrand Petit

5.8. Platforms

5.8.1. BehaviorTrack

Keyword: Web tracking detection, Large-scale measurement

Description: In our study, we propose a tracking detection method inspired by analyzing behavior of invisible pixels. By crawling 84,658 webpages from 8,744 domains, we detect that third-party invisible pixels are widely deployed: they are present on more than 94.51% of domains and constitute 35.66% of all third-party images. We propose a fine-grained behavioral classification of tracking based on the analysis of invisible pixels. BehaviorTrack uses this classification to detect new categories of tracking and uncover new collaborations between domains on the full dataset of 4,216,454 third-party requests.

- Contact: Imane Fouad
- URL: <http://www-sop.inria.fr/members/Imane.Fouad/pixeltrack>

RMOD Project-Team

6. New Software and Platforms

6.1. Moose

Moose: Software and Data Analysis Platform

KEYWORDS: Software engineering - Meta model - Software visualisation

FUNCTIONAL DESCRIPTION: Moose is an extensive platform for software and data analysis. It offers multiple services ranging from importing and parsing data, to modeling, to measuring, querying, mining, and to building interactive and visual analysis tools. The development of Moose has been evaluated to 200 man/year.

Mots-cles : MetaModeling, Program Visualization, Software metrics, Code Duplication, Software analyses, Parsers

- Participants: Anne Etien, Nicolas Anquetil, Olivier Auverlot, Stéphane Ducasse, Julien Delplanque, Guillaume Larcheveque, Cyril Ferlicot-Delbecque and Pavel Krivanek
- Partners: Université de Berne - Sensus - Synectique - Pleiad - USI - Vrije Universiteit Brussel
- Contact: Stéphane Ducasse
- URL: <http://www.moosetechnology.org>

6.2. Pharo

KEYWORDS: Live programmation objet - Reflective system - Web Application

FUNCTIONAL DESCRIPTION: Pharo is a pure object reflective and dynamic language inspired by Smalltalk. In addition, Pharo comes with a full advanced programming environment developed under the MIT License. It provides a platform for innovative development both in industry and research. By providing a stable and small core system, excellent developer tools, and maintained releases, Pharo's goal is to be a platform to build and deploy mission critical applications, while at the same time continue to evolve. Pharo 60 got 100 contributors world-wide. It is used by around 30 universities, 15 research groups and around 40 companies.

- Participants: Christophe Demarey, Clement Bera, Damien Pollet, Esteban Lorenzano, Marcus Denker, Stéphane Ducasse and Guillermo Polito
- Partners: BetaNine - Reveal - Inceptive - Netstyle - Feenk - ObjectProfile - GemTalk Systems - Greyc Université de Caen - Basse-Normandie - Université de Berne - Yesplan - RMod - Pleiad - Sensus - Université de Bretagne Occidentale - École des Mines de Douai - ENSTA - Uqbar foundation Argentina - LAM Research - ZWEIDENKER - LifeWare - JPMorgan Chase - KnowRoaming - ENIT - Spesenfuchs - FINWorks - Esug - FAST - Ingenieubüro Schmidt - Projector Software - HRWorks - Inspired.org - Palantir Solutions - High Octane - Soops - Osoco - Ta Mère SCRL - University of Yaounde 1 - Software Quality Laboratory, University of Novi Sad - Software Institute Università della Svizzera italiana - Universidad Nacional de Quilmes - UMMISCO IRD - Université technique de Prague
- Contact: Marcus Denker
- URL: <http://www.pharo.org>

6.3. Pillar

KEYWORDS: HTML - LaTeX - HTML5

FUNCTIONAL DESCRIPTION: Pillar is a markup syntax and associated tools to write and generate documentation and books. Pillar is currently used to write several books and other documentation. It is used in the tools developed by Feenk.com.

- Partner: Feenk
- Contact: Stéphane Ducasse
- URL: <https://github.com/Pillar-markup/pillar>

AGORA Project-Team

6. New Software and Platforms

6.1. TAPASCologne

Travel and Activity PATterns Simulation Cologne

KEYWORDS: Mobility - Traces

FUNCTIONAL DESCRIPTION: TAPASCologne is an initiative by the Institute of Transportation Systems at the German Aerospace Center (ITS-DLR), aimed at reproducing, with the highest level of realism possible, car traffic in the greater urban area of the city of Cologne, in Germany.

To that end, different state-of-art data sources and simulation tools are brought together, so to cover all of the specific aspects required for a proper characterization of vehicular traffic:

The street layout of the Cologne urban area is obtained from the OpenStreetMap (OSM) database, The microscopic mobility of vehicles is simulated with the Simulation of Urban Mobility (SUMO) software, The traffic demand information on the macroscopic traffic flows across the Cologne urban area (i.e., the O/D matrix) is derived through the Travel and Activity PATterns Simulation (TAPAS) methodology, The traffic assignment of the vehicular flows described by the TAPASCologne O/D matrix over the road topology is performed by means of Gawron's dynamic user assignment algorithm.

- Participants: Marco Fiore and Razvan Stanica
- Contact: Marco Fiore
- URL: <http://kolntrace.project.citi-lab.fr/#download>

6.2. Sense in the City

KEYWORDS: Sensors - Sensors network - Wireless Sensor Networks

FUNCTIONAL DESCRIPTION: Sense in the city is a lightweight experimentation platform for wireless sensor networks in development. The main objective of this platform is to be easily transferable and deployable on the field. It allows a simplified deployment of the code running on the sensors and the collection of logs generated by the instrumentation of the code on a centralized database. In the early stage of the platform, the sensors are powered by small PCs, e.g. Raspberry Pi, but we are investigating the integration of energy harvesting capabilities such as solar panels.

- Participants: Hervé Rivano and Khaled Boussetta
- Contact: Khaled Boussetta

6.3. PrivaMovApp

KEYWORD: Crowd-sensing

FUNCTIONAL DESCRIPTION: Agora is leading the development of an Android application for user data collection purposes. The application is based on the Funf framework, and is currently available on Google Play.

- Participants: Stéphane d'Alu, Hervé Rivano, Razvan Stanica and Solohaja Rabenjamina
- Contact: Razvan Stanica

6.4. WSNet

KEYWORD: Network simulator

FUNCTIONAL DESCRIPTION: WSNNet is a modular event-driven simulator targeted to Wireless Sensor Networks. Its main goals are to offer scalability, extensibility and modularity for the integration of new protocols/hardware models and a precise radio medium simulation. We still hope to find the proper resource to make WSNNet evolve into a wireless capillary network simulator suitable for conducting simulations at the urban scale.

- Participants: Rodrigue Domga Komguem and Fabrice Valois
- Partner: CEA-LETI
- Contact: Guillaume Chelius
- URL: <https://gforge.inria.fr/projects/wsnet-3/>

6.5. Platforms

6.5.1. PPAIR Plateforme LoRa - Campus Connecté

The project aims at providing a platform that offers connectivity through a long-range, low-energy network to smart objects. The platform uses LoRa technology, which offers a wide connectivity, covering the entire INSA Lyon campus and providing a data collection service to all campus users. The main purpose of the LoRa platform is: (i) research (researchers can use it for studying reliability and capacity problems, privacy related challenges, etc.), and (ii) teaching (several courses from INSA, especially in the Telecom department can use this platform as a pedagogical tool).

Part of the software is mutualized with the University of Paris 13, where a LoRaWan testbed project is under deployment at the campus of Villeteuse. This project, is supported by a local BQR and is lead by Khaled Boussetta. The mutualization of the software tools will allow us to conduct multi sites experiments, at Lyon and at Paris. Since 2019, this platform is used in the European Project Interreg Med ESMARTCITY and for the PHC Ulysses (joint collaboration with Nimbus Center, Ireland).

6.5.2. UrPolSens Platform

We designed from scratch an energy efficient air pollution sensor network using Atmega micro-controllers and electrochemical air pollution probes. The micro-controller is integrated into a lab-designed printed circuit which includes among others: a high precision ADC, a micro-SD card reader and a radio communication module. The designed nodes measure the nitrogen dioxide (NO₂) pollutant in addition to temperature and humidity and transmit data using LoRa to a gateway, which is connected to our servers using a 4G connection. The sensors are also equipped with solar panels in order to extend their lifetime when their batteries are drained. Our platform has been operational in the downtown of the Lyon city with 12 sensor nodes deployed in the Garibaldi street from mid-July to Mid-October 2018.

6.5.3. 3M'air sensor platform

We developed the 3M'Air sensor platform to be used in participatory sensing of temperature and air quality. We have built our own nodes equipped with multiple sensors measuring Nitrogen-Dioxide (NO₂), Particulate Matter (PM₁, PM_{2.5}, PM₁₀), temperature and humidity. They are battery-powered and equipped with a GPS module to have the position of the measurements. Data are stored on a micro SD card and at the same time sent over LoRa to a server we have developed that is responsible to store these data for future analyses. A web platform has also been developed to display the collected concentration measurements in real time. This developed solution is used in several participatory planned measurement campaigns in Lyon city.

COATI Project-Team

6. New Software and Platforms

6.1. GRPH

The high performance graph library for Java

KEYWORDS: Graph - Graph algorithmics - Java

FUNCTIONAL DESCRIPTION: Grph is an open-source Java library for the manipulation of graphs. Its design objectives are to make it portable, simple to use/extend, computationally/memory efficient, and, according to its initial motivation: useful in the context of graph experimentation and network simulation. Grph also has the particularity to come with tools like an evolutionary computation engine, a bridge to linear programming solvers, a framework for distributed computing, etc.

Grph offers a very general model of graphs. Unlike other graph libraries which impose the user to first decide if he wants to deal with directed, undirected, hyper (or not) graphs, the model offered by Grph is unified in a general class that supports mixed graphs made of undirected and directed simple and hyper edges. Grph achieves great efficiency through the use of multiple code optimization techniques such as multi-core parallelism, caching, adequate data structures, use of primitive objects, exploitation of low-level processor caches, on-the-fly compilation of specific C/C++ code, etc. Grph attempts to access the Internet in order to check if a new version is available and to report who is using it (login name and hostname). This has no impact whatsoever on performance and security.

- Participants: Aurélien Lancin, David Coudert, Issam Tahiri, Luc Hogie and Nathann Cohen
- Contact: Luc Hogie
- URL: <http://www.i3s.unice.fr/~hogie/grph/>

6.2. BigGraphs

KEYWORDS: Graph algorithmics - Distributed computing - Java - Graph processing

FUNCTIONAL DESCRIPTION: The objective of BigGraphs is to provide a distributed platform for very large graphs processing. A typical data set for testing purpose is a sample of the Twitter graph : 240GB on disk, 398M vertices, 23G edges, average degree of 58 and max degree of 24635412.

We started the project in 2014 with the evaluation of existing middlewares (GraphX / Spark and Giraph / Hadoop). After having tested some useful algorithms (written according to the Bulk Synchronous Parallel (BSP) model) we decided to develop our own platform.

This platform is based on the existing BIGGRPH library and we are now working on improving the quality of the code. In particular we have designed strong test suites and some non trivial bugs have been fixed. We also have solved problems of scalability, in particular concerning the communication layer with billions of messages exchanged between BSP steps. We also have implemented specific data structures for BSP and support for distributed debugging. This comes along with the implementation of algorithms such as BFS or strongly connected components that are run on the NEF cluster.

In 2017 we have developed a multi-threaded shared-memory parallel version of the BSP framework. This new version uses advanced synchronization mechanisms and strategies to minimize the congestion of multiple threads working on the same graph. Using the NEF cluster (Inria Sophia Antipolis), this parallel version exhibits speed-ups up to 6.5 using 8 nodes (16 cores each) when computing a BFS on the 23 G edges Twitter graph sample.

- Participants: Luc Hogie, Michel Syska and Nicolas Chleq
- Partner: CNRS
- Contact: Luc Hogie
- URL: <http://www.i3s.unice.fr/~hogie/software/?name=biggrph>

6.3. JMaxGraph

KEYWORDS: Java - HPC - Graph algorithmics

FUNCTIONAL DESCRIPTION: JMaxGraph is a collection of techniques for the computation of large graphs on one single computer. The motivation for such a centralized computing platform originates in the constantly increasing efficiency of computers which now come with hundred gigabytes of RAM, tens of cores and fast drives. JMaxGraph implements a compact adjacency-table for the representation of the graph in memory. This data structure is designed to 1) be fed page by page, à-la GraphChi, 2) enable fast iteration, avoiding memory jumps as much as possible in order to benefit from hardware caches, 3) be tackled in parallel by multiple-threads. Also, JMaxGraph comes with a flexible and resilient batch-oriented middleware, which is suited to executing long computations on shared clusters. The first use-case of JMaxGraph allowed F. Giroire, T. Trollet and S. Pérennes to count K2,2s, and various types of directed triangles in the Twitter graph of users (23G arcs, 400M vertices). The computation campaign took 4 days, using up to 400 cores in the NEF Inria cluster.

- Contact: Luc Hogue
- URL: <http://www.i3s.unice.fr/~hogie/software/?name=jmaxgraph>

6.4. Sagemath

SageMath

KEYWORDS: Graph algorithmics - Graph - Combinatorics - Probability - Matroids - Geometry - Numerical optimization

SCIENTIFIC DESCRIPTION: SageMath is a free open-source mathematics software system. It builds on top of many existing open-source packages: NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, R and many more. Access their combined power through a common, Python-based language or directly via interfaces or wrappers.

FUNCTIONAL DESCRIPTION: SageMath is an open-source mathematics software initially created by William Stein (Professor of mathematics at Washington University). We contribute the addition of new graph algorithms along with their documentations and the improvement of underlying data structures.

RELEASE FUNCTIONAL DESCRIPTION: See <http://www.sagemath.org/changelogs/>

NEWS OF THE YEAR: 1) Improvement of shortest path computation algorithms. Done in the context of Google Summer of Code 2019. 2) Main contributor for making the graph module (more than 100,000 lines of code) of SageMath compatible with Python3. Version 9.0 of Sagemath, released on January 1st, 2020, is 100% Python3 compliant.

- Participant: David Coudert
- Contact: David Coudert
- URL: <http://www.sagemath.org/>

DANTE Project-Team

6. New Software and Platforms

6.1. GraSP

Graph Signal Processing

KEYWORDS: Matlab - LaTeX - Graph - Graph visualization - Signal processing - GNU Octave

FUNCTIONAL DESCRIPTION: Matlab / GNU Octave toolbox to manipulate and visualize signals on graphs. LaTeX package to draw signals.

- Contact: Benjamin Girault

6.2. IoT-LAB aggregation-tools

KEYWORD: Internet of things

FUNCTIONAL DESCRIPTION: IoT-LAB aggregation-tools allow aggregating data results from many nodes at a time. It connects to several tcp connections and handle the received data.

- Participant: Gaetan Harter
- Contact: Eric Fleury
- URL: <https://github.com/iot-lab/aggregation-tools>

6.3. IoT-LAB cli-tools

KEYWORD: Internet of things

FUNCTIONAL DESCRIPTION: IoT-LAB cli-tools provide a basic set of operations for managing IoT-LAB experiments from the command-line.

- Participants: Frederic Saint-Marcel and Gaetan Harter
- Contact: Eric Fleury
- URL: <https://github.com/iot-lab/cli-tools>

6.4. IoT-LAB gateway

KEYWORD: Internet of things

FUNCTIONAL DESCRIPTION: IoT-LAB software embedded on a IoT-LAB gateway node new generation provides the local management of the experiment on that node. It is a software bridge between the IoT-LAB server, the user open node and the control node.

- Contact: Frederic Saint-Marcel
- URL: <https://github.com/iot-lab/iot-lab-gateway>

6.5. IoT-LAB robots

KEYWORDS: Internet of things - Robotics

FUNCTIONAL DESCRIPTION: IoT-LAB robots is an embedded robot controller on a Turtlebot2 providing the IoT-LAB node mobility functionality

- Partner: Université de Strasbourg
- Contact: Julien Vandaele
- URL: <https://github.com/iot-lab/>

6.6. Queueing Systems

FUNCTIONAL DESCRIPTION: This tool aims at providing a simple web interface to promote the use of our proposed solutions to numerically solve classical queueing systems.

- Participants: Alexandre Brandwajn and Thomas Begin
- Contact: Thomas Begin
- URL: <http://queueing-systems.ens-lyon.fr/>

6.7. WSNet

KEYWORD: Network simulator

FUNCTIONAL DESCRIPTION: WSNet is a modular event-driven simulator targeted to Wireless Sensor Networks. Its main goals are to offer scalability, extensibility and modularity for the integration of new protocols/hardware models and a precise radio medium simulation. We still hope to find the proper resource to make WSNet evolve into a wireless capillary network simulator suitable for conducting simulations at the urban scale.

- Participants: Rodrigue Domga Komguem and Fabrice Valois
- Partner: CEA-LETI
- Contact: Guillaume Chelius
- URL: <https://gforge.inria.fr/projects/wsnet-3/>

DIANA Project-Team

5. New Software and Platforms

5.1. ACQUAmobile

KEYWORDS: Android - Internet access - Performance measure - Quality of Experience

FUNCTIONAL DESCRIPTION: ACQUA is an Application for prediCting QQuality of Experience (QoE) at Internet Access [21]. It is developed by the Diana team at Inria Sophia Antipolis – Méditerranée and was supported by Inria under the ADT ACQUA grant. The scientific project around ACQUA is supported by Inria Project Lab BetterNet and the French National Project ANR BottleNet. The project also got the approval of Inria COERLE and French CNIL for the part on experimentation with real users. ACQUA presents a new way for the evaluation of the performance of Internet access. Starting from network-level measurements as the ones we often do today (bandwidth, delay, loss rates, jitter, etc), ACQUA targets the estimated Quality of Experience (QoE) related to the different applications of interest to the user without the need to run them (e.g., estimated Skype quality, estimated video streaming quality).

An application in ACQUA is a function, or a model, that links the network-level and device-level measurements to the expected Quality of Experience. Supervised machine learning techniques are used to establish such link between measurements both at the network level and the device level, and estimations of the Quality of Experience for different Internet applications. The required data for such learning can be obtained either by controlled experiments as we did in [26] on YouTube Quality of Experience, or by soliciting the crowd (i.e. crowdsourcing) for combinations (i.e. tuples) of measurements and corresponding application-level Quality of Experience. Our current work is concentrating on using the ACQUA principle in the estimation and prediction of the Quality of Experience for main user's applications. We refer to the web site of the project for further details.

The ACQUA Android application is supposed to be on one hand the reference application for QoE forecasting and troubleshooting for end users at their Internet access, and on the other hand, the feedback channel that allows end users to report to us (if they are willing) on their experience together with the corresponding network measurements so as to help us calibrating better and more realistic models. For this calibration, we are currently performing extensive, efficient and automatic measurements in the laboratory, we will count on end users to help us completing this dataset with further applications and more realistic network and user conditions.

ACQUA is mainly meant for end users, but it is also of interest to (mobile) network operators and to content providers to estimate the QoE of their customers and their networks without each time having to run expensive application-level traffic and to involve real users.

Assessment: Audience = 3, Software Originality = 4, Software Maturity = 3, Evolution and Maintenance = 3, Software Distribution and Licensing = 5.

- Authors: Thierry Spetebroot and Chadi Barakat
- Contact: Chadi Barakat
- URL: <http://project.inria.fr/acqua/>

5.2. ElectroSmart

KEYWORDS: Crowd-sourcing - UMTS - GSM - Bluetooth - Wi-Fi - 4G - 3G - 2G - Electromagnetic waves - Android - LTE

FUNCTIONAL DESCRIPTION: The Internet and new devices such as smartphones have fundamentally changed the way people communicate, but this technological revolution comes at the price of a higher exposition of the general population to microwave electromagnetic fields (EMF). This exposition is a concern for health agencies and epidemiologists who want to understand the impact of such an exposition on health, for the general public who wants a higher transparency on its exposition and the health hazard it might represent, but also for cellular operators and regulation authorities who want to improve the cellular coverage while limiting the exposition, and for computer scientists who want to better understand the network connectivity in order to optimize communication protocols. Despite the fundamental importance to understand the exposition of the general public to EMF, it is poorly understood because of the formidable difficulty to measure, model, and analyze this exposition.

The goal of the ElectroSmart project is to develop the instrument, methods, and models to compute the exposition of the general public to microwave electromagnetic fields used by wireless protocols and infrastructures such as Wi-Fi, Bluetooth, or cellular. Using a pluri-disciplinary approach combining crowd-based measurements, in-lab experiments, and modeling using sparse and noisy data, we address challenges such as designing and implementing a measuring instrument leveraging on crowd-based measurements from mobile devices such as smartphones, modeling the exposition of the general public to EMF to compute the most accurate estimation of the exposition, and analyzing the evolution of the exposition to EMF with time. This technological breakthrough will have scientific, technical, and societal applications, notably on public health politics, by providing the scientific community and potential users with a unique measuring instrument, methods, and models to exploit the invaluable data gathered by the instrument.

This project is supported by the UCN@Sophia Labex in 2016/2017/2018 (funding the engineer Mondri Ravi), by an Inria ADT (funding the engineer Abdelhakim Akodadi) 2017/2018, by and Inria ATT (funding the business developer David Migliacci) in 2017/2018, and by the academy 1 of UCAJedi (funding a Ph.D. student Yanis Boussad) 2017/2020.

In August 2016, we released the first stable public release of ElectroSmart. On the 07th January 2020, we acquire 1000 new daily users, and have 20k weekly active users.

Assessment: A-5, SO-4, SM-4, EM-3-up4, SDL-1

We are in a process of creating a startup to commercialize the exposition maps we can build with the data we are collecting.

- Participants: Arnaud Legout, Abdelhakim Akodadi, Hackob Melconian, Inderjeet Singh and Mondri Ravi
- Contact: Arnaud Legout
- URL: https://es.inria.fr/home/index?path_prefix=en

5.3. nepi-ng

KEYWORDS: Wireless network - Experimentation

FUNCTIONAL DESCRIPTION: In the specific context of R2lab, we have created a tool suite for orchestrating network experiments, that for historical reasons we refer to collectively as nepi-ng, for NEPI new generation. An umbrella website is available at <https://nepi-ng.inria.fr/>.

At this point, nepi-ng has a much smaller scope than its NEPI ancestor used to have, in that it only supports remote control of network experiments over ssh. As a matter of fact, in practice, this is the only access mechanism that we need to have for running experiments on both R2lab, and PlanetLab Europe.

The design of nepi-ng of course is modular, so that it will be perfectly possible to add other control mechanisms to this core if and when it becomes necessary.

nepi-ng is currently made of two separate Python libraries:

- **asynciojobs:**
 - URL: <http://asynciojobs.readthedocs.io/en/latest/>
 - Version: asynciojobs v0.5.4
 - Keywords: networking experimentation, orchestration
 - License: CC BY-SA 4.0
 - Type of human computer interaction: Python library
 - OS/Middleware: Linux
 - Required library or software: Python-3.5 / asyncio
 - Programming language: Python3
- **apssh:**
 - URL: <http://apssh.readthedocs.io/en/latest/>
 - Version: apssh v0.7.1
 - Keywords: networking experimentation, orchestration
 - License: CC BY-SA 4.0
 - Type of human computer interaction: Python library
 - OS/Middleware: Linux
 - Required library or software: Python-3.5 / asyncio
 - Programming language: Python3

Assessment: A-2, SO-3, SM-3, EM-3, DSL-4

- Contact: Thierry Parmentelat
- URL: <http://nepi-ng.inria.fr>

5.4. Distrinet

KEYWORDS: SDN - Emulation - Large-scale Emulators - Network simulator

SCIENTIFIC DESCRIPTION: Networks have become complex systems that combine various concepts, techniques, and technologies. As a consequence, modelling or simulating them is now extremely complicated and researchers massively resort to prototyping techniques. Two experimental techniques are mainly used when it comes to testing a network: simulation and emulation. Emulation provides a good accuracy and allows to test the applications directly in an environment that is similar to a real one. Most of the emulators do not take into account the scalability, because usually they are designed to be executed in a single machine. Among other tools, Mininet is the most popular when it comes to evaluate SDN propositions. It allows to emulate SDN networks on a single computer. Unfortunately, Mininet shows its limitations with resource intensive experiments as the emulating host may become overloaded. To tackle this issue, we propose Distrinet, a distributed implementation of Mininet over multiple hosts. Distrinet uses the same API than Mininet, meaning that it is compatible with Mininet programs. It is generic and can deploy experiments in Linux clusters or in the Amazon EC2 cloud.

Assessment: A5, SO3, SM2, EM2-down, SDL4

FUNCTIONAL DESCRIPTION: Distrinet is an extension of Mininet that relies on LXC to be distributed in the cloud, and particularly in Amazon.

RELEASE FUNCTIONAL DESCRIPTION: First release

- Participants: Damien Saucez, Giuseppe Di Lena, Andrea Tomassilli, Frédéric Giroire, Thierry Turlitti and Walid Dabbous
- Partner: Orange Labs
- Contact: Walid Dabbous
- URL: <https://distrinet-emu.github.io>

5.5. Platforms

5.5.1. *Reproducible research Lab - R2lab*

Scientific work around network protocols and related software stacks requires experiments, hence experimental conditions, to be reproducible. This is a particularly challenging requirement in the wireless networking area, where characteristics of wireless channels are known to be variable, unpredictable and hardly controllable.

The R2lab wireless testbed was designed with reproducibility as its central characteristics; it is built around an isolated and anechoic chamber, featuring RF absorbers that prevent radio waves reflections, and a Faraday cage blocking external interferences. R2lab thus provides an ideal environment for running reproducible wireless experiments.

R2lab has been operated for 4 years now, in the context of the FIT (Future Internet of Things) Equipment of Excellence project, and as such, it is now federated with the other testbeds that are part of the FIT initiative. As of early 2019, it is now also federated within the Fed4Fire initiative.

Available toolsets, both hardware and software, are mostly stable apart from low noise marginal deployment of new kinds of radio devices, that now encompass among 5G and LoRa, among others. Our focus at this point of the project is to leverage our initial technical and financial investment, and to produce scientific work around reproducibility, particularly from a methodological standpoint, as illustrated by various publications [33], [34].

Worth being mentioned as well, as part of a partnership with the OpenAirInterface initiative, R2lab is used on a daily basis for system-wide regression tests of the OAI stack, which in return allows us to offer up-to-date images for running OAI-based experiments.

Access to R2lab is open 24/7. We currently have around 200 active users from all over the world among them 40 new users registered in 2019. For more details see <http://r2lab.inria.fr>.

5.5.2. *Network simulator for aircrafts*

- Keywords: network, simulation, real-time
- Functional Description: In collaboration with Safran Electrical and Power we produced a network design tool for aircrafts. This tool simulates aircraft networks. The tool is about 10,000 lines of code, out of which we produced 2,000.
- Assessment: A-2up,SO-3,SM-2up,EM-4,SDL-3,OC-DA-CD-TPM
- Licence: confidential
- URL: confidential
- Contact: Damien Saucez

DIONYSOS Project-Team

6. New Software and Platforms

6.1. IPv6 Test Toolkit

FUNCTIONAL DESCRIPTION: These test suites are developed using the TTCN-3 environment.

The packages contains the full Abstract Test Suites written in TTCN-3 and the source files for building the codecs and adapters with the help of T3DevKit.

- Participants: Annie Floch, Anthony Baire, Ariel Sabiguero, Bruno Deniaud, César Viho and Frédéric Roudaut
- Contact: César Viho

6.2. Passive Test Tool

- Participants: Anthony Baire and César Viho
- Contact: Anthony Baire

6.3. T3DevKit

KEYWORDS: IPv6 - Conformance testing - TTCN-3

SCIENTIFIC DESCRIPTION: We have built a toolkit for easing executing tests written in the standardized TTCN-3 test specification language. This toolkit is made of a C++ library together with a highly customizable CoDec generator that allows fast development of external components (that are required to execute a test suite) such as CoDec (for message Coding/Decoding), System and Platform Adapters. It also provides a framework for representing and manipulating TTCN-3 events so as to ease the production of test reports. The toolkit addresses issues that are not yet covered by ETSI standards while being fully compatible with the existing standard interfaces: TRI (Test Runtime Interfaces) and TCI (Test Control Interfaces), it has been tested with four TTCN-3 environments (IBM, Elvior, Danet and Go4IT) and on three different platforms (Linux, Windows and Cygwin).

FUNCTIONAL DESCRIPTION: T3DevKit is a free open source toolkit to ease the development of test suites in the TTCN-3 environment. It provides:

a CoDec generator (t3cdgen) that automates the development process of the CoDec needed for coding TTCN-3 values into physically transmittable messages and decoding incoming messages a library (t3devlib) that provides an object oriented framework to manipulate TTCN-3 entities (values, ports, timers, external functions. . .) an implementation of the TRI and TCI standard interfaces default implementations for the system adapter (SA), platform adapter (PA), test management (TM), test logging (TL) and component handling (CH) modules default codecs build scripts for the generation of executable test suites, these are tool-independent and facilitate the distribution of test suite sources

- Participants: Annie Floch, Anthony Baire, Ariel Sabiguero, César Viho and Frédéric Roudaut
- Contact: Federico Sismondi

6.4. ttproto

Testing Tool Prototype

KEYWORDS: Interoperability - Conformance testing - TTCN-3

FUNCTIONAL DESCRIPTION: ttproto is an experimental tool for implementing testing tools, for conformance and interoperability testing.

It was first implemented to explore new features and concepts for the TTCN-3 standard, but we also used it to implement a passive interoperability test suite we provided for the CoAP interoperability event held in Paris in March 2012.

This tool is implemented in python3 and its design was influenced mainly by TTCN-3 (abstract model, templates, snapshots, behaviour trees, communication ports, logging) and by Scapy (syntax, flexibility, customisability)

Its purpose is to facilitate rapid prototyping rather than experimentations (rather than production use). We choosed to maximise its modularity and readability rather than performances and real-time considerations.

Now you should have a look at the Features page: https://www.irisa.fr/tipi/wiki/doku.php/testing_tool_prototype:features

- Contact: Federico Sismondi
- URL: https://www.irisa.fr/tipi/wiki/doku.php/testing_tool_prototype

6.5. CoAP Testing Tool

KEYWORDS: Test - Interoperability - Conformance testing - Plugtests

FUNCTIONAL DESCRIPTION: The software helps developers of the CoAP protocol assessing if their implementations (either CoAP clients or CoAP servers) are conformant to protocol specifications, and interoperable with other implementations. It encompasses:

- Coordination of CoAP interoperability tests
- Analysis of CoAP traces & issuing verdicts
- Automation of open source CoAP implementations for based reference interop testing
- Authors: Federico Sismondi and César Viho
- Contact: Federico Sismondi

6.6. ioppytest

Interoperability testing

KEYWORDS: Interoperability - Conformance testing - CoAP - 6LoWPAN - OneM2M

FUNCTIONAL DESCRIPTION: The software is a framework for developing interoperability tests. The interoperability tests help developers of network protocol assessing if their implementations are conformant to protocol specifications, and interoperable with other implementations.

The software already integrates interoperability tests for CoAP, OneM2M and 6LoWPAN The framework provides the following features to the users:

- Coordination of the interoperability tests (enabling remote testing)
- VPN-like connectivity between users' implementations (enabling remote testing)
- Analysis of exchanged network traces & issuing verdicts
- Automation of open source implementations for based reference interop testing

This framework is the evolution of the CoAP Testing Tool (<https://bil.inria.fr/fr/software/view/2937/tab>)

- Contact: Federico Sismondi
- URL: <https://gitlab.f-interop.eu/f-interop-contributors/ioppytest>

DYOGENE Project-Team

6. New Software and Platforms

6.1. Platforms

6.1.1. *CapRadio*

Cellular network dimensioning toolbox *CapRadio* is being developed by Orange in a long-term collaboration between TREC/DYOGENE represented by B. Błaszczyszyn, and Orange Labs, represented by M. K. Karray. This year we are working on taking into account the “massive MIMO” in 5G cellular networks; see [8.1.1](#) .

EVA Project-Team

6. New Software and Platforms

6.1. OpenWSN

KEYWORDS: Internet of things - 6TiSCH - 6LoWPAN - CoAP

FUNCTIONAL DESCRIPTION: OpenWSN is an open-source implementation of a fully standards-based protocol stack for the Internet of Things. It has become the de-facto implementation of the IEEE802.15.4e TSCH standard, has a vibrant community of academic and industrial users, and is the reference implementation of the work we do in the IETF 6TiSCH standardization working group.

- Partner: University of California Berkeley
- Contact: Thomas Watteyne
- URL: <http://www.openwsn.org/>

6.2. 6TiSCH Simulator

High-level simulator of a 6TiSCH network

KEYWORDS: Network simulator - 6TiSCH

FUNCTIONAL DESCRIPTION: The simulator is written in Python. While it doesn't provide a cycle-accurate emulation, it does implement the functional behavior of a node running the full 6TiSCH protocol stack. This includes RPL, 6LoWPAN, CoAP and 6P. The implementation work tracks the progress of the standardization process at the IETF.

- Contact: Malisa Vucinic

6.3. Argus

KEYWORDS: Cloud - Low-Power Wireless - Sniffer

FUNCTIONAL DESCRIPTION: There are three pieces to the Argus:

The Argus Probe is the program which attaches to your low-power wireless sniffer and forwards its traffic to the Argus Broker.

The Argus Broker sits somewhere in the cloud. Based on MQTT, it connects Argus Probes with Argus Clients based on a pub-sub architecture.

Several Argus Clients can be started at the same time. It is a program which subscribes to the Argus Broker and displays the frames in Wireshark.

- Contact: Remy Leone

6.4. SolSystem

Sensor Object Library System

KEYWORDS: Low-Power Wireless - Back-End System - SmartMesh IP

FUNCTIONAL DESCRIPTION: The source code is composed of the definition of the SOL structure (<https://github.com/realms-team/sol>), the code that runs on the manager (<https://github.com/realms-team/solmanager>, written in Python) and the code that runs on the server receiving the data (<https://github.com/realms-team/solserver>, written in Python)

- Contact: Keoma Brun-Laguna
- URL: <http://www.solssystem.io/>

6.5. 6TiSCH Wireshark Dissector

KEYWORDS: 6TiSCH - Wireshark

FUNCTIONAL DESCRIPTION: Implementation on the dissectors is done through an open-source repository, stable code is regularly contributed back to the main Wireshark code base.

- Contact: Jonathan Muñoz

6.6. F-Interop

Remote Conformance and Interoperability Tests for the Internet of Thing

KEYWORDS: Interoperability - Iot - Conformance testing - Standardization

- Partners: UPMC - IMEC - ETSI - EANTC - Mandat International - Digital Catapult - University of Luxembourg - Device Gateway
- Contact: Remy Leone

6.7. Mercator

KEYWORDS: Deployment - Low-Power Wireless - Testbeds - Connectivity

FUNCTIONAL DESCRIPTION: The firmware is written as part of the OpenWSN project. Scripts and analysis tools are written in Python.

- Contact: Keoma Brun-Laguna

FUN Project-Team

5. New Software and Platforms

5.1. AspireRFID ALE

FUNCTIONAL DESCRIPTION: AspireRFID middleware is a modular OW2 open source RFID middleware. It is compliant with EPC Global standards. This new module integrates the modifications of the new standard release, including new RP and LLRP definitions and fixing bugs. This module has been implemented in the framework of the MIAOU project.

- Participants: Ibrahim Amadou, Julien Vandaele, Nathalie Mitton and Rim Driss
- Contact: Nathalie Mitton

5.2. ETINODE-CONTIKI-PORT

KEYWORD: Iot

FUNCTIONAL DESCRIPTION: Contiki is an open source embedded OS for Internet of Things (IoT). It is light and portable to different hardware architectures. It embeds communication stacks for IoT. This driver allows the running of Contiki OS over Etnode-MSP430. The code also allows the use of radio chip and embedded sensors. This module has been implemented in the framework of the ETIPOPS project.

- Participants: Nathalie Mitton, Roudy Dagher and Salvatore Guzzo Bonifacio
- Contact: Salvatore Guzzo Bonifacio

5.3. ETINODE-DRIVERS

FUNCTIONAL DESCRIPTION: These drivers for Etnode-MSP430 control the different embedded sensors and hardware components available on an Etnode-MSP430 node such as gyroscope, accelerometer and barometric sensor. This module has been implemented in the framework of the ETIPOPS project.

- Participants: Nathalie Mitton, Roudy Dagher and Salvatore Guzzo Bonifacio
- Contact: Salvatore Guzzo Bonifacio

5.4. EVe-TCF

Embedded Verifier for Transitive Control Flow

KEYWORDS: Control Flow - JavaCard - Embedded systems - Embedded - Security - Code analysis

FUNCTIONAL DESCRIPTION: Verification of transitive control flow policies on JavaCard 2.x bytecode. Control flow policies expressed using a DSL language are embedded in JavaCard packages (CAP files) using EVe-TCF convert tool. Control flow policies are then statically verified on-device at loading-time thanks to an embedded verifier (designed for smart cards in EVe-TCF). EVe-TCF (Embedded Verifier for Transitive Control Flow) also contains an off-device (i.e. PC tool) to simulate on-device loading process of JavaCard 2.x platforms with GlobalPlatform 2.x installed.

- Participants: Arnaud Fontaine and Isabelle Simplot Ryl
- Contact: Nathalie Mitton

5.5. GOLIATH

Generic Optimized Lightweight communication stack for Ambient Technologies

KEYWORDS: WSN - WSN430

FUNCTIONAL DESCRIPTION: GOLIATH (Generic Optimized LIghtweight communication stack for Ambient TecHnologies) is a full protocol stack for wireless sensor networks. This module has been implemented in the framework of the ETIPOPS project.

- Participants: David Simplot Ryl, Fadila Khadar, Nathalie Mitton and Salvatore Guzzo Bonifacio
- Contact: Nathalie Mitton
- URL: <https://gforge.inria.fr/projects/goliath/>

5.6. IoT-LAB robots

KEYWORDS: Internet of things - Robotics

FUNCTIONAL DESCRIPTION: IoT-LAB robots is an embedded robot controller on a Turtlebot2 providing the IoT-LAB node mobility functionality

- Partner: Université de Strasbourg
- Contact: Julien Vandaele
- URL: <https://github.com/iot-lab/>

5.7. T-SCAN

KEYWORDS: Rfid - RFID Middleware

FUNCTIONAL DESCRIPTION: T-Scan is an interface ensuring the translation from a SGTIN tag format to an ONS hostname format according to the EPCGlobal standards. It allows the sending of a DNS request to look up the EPC-IS aides to which the product belongs in order to access the data relative to that product. This module has been implemented in the framework of the TRACAVERRERRE project.

- Participants: Gabriele Sabatino and Nathalie Mitton
- Contact: Gabriele Sabatino

5.8. FIT IoT-Lab

Participants: Nathalie Mitton [contact person], Julien Vandaele, Matthieu Berthome.

FIT IoT-LAB (<http://www.iot-lab.info>) is a very large scale open testbed that features over 2700 wireless sensor nodes and more than 200 robots spread across six different sites in France. Nodes are either fixed or mobile and can be allocated in various topologies throughout all sites. A variety of wireless sensors are available, with different processor architectures (MSP430, STM32 and Cortex-A8) and different wireless chips (802.15.4 PHY at 800 MHz or 2.4 GHz). In addition, "open nodes" can receive custom wireless sensors for inclusion in IoT-LAB testbed. This platform is completely open and can be used by any one wishing to run experiment on wireless sensors and robots.

The Lille site displays 2 subsets of the platforms:

- Haute Borne: this site features 256 M3 sensor nodes operating in the 2.4GHz band and 64 mobile robots (32 turtlebots and 32 wifibots) completely remotely programmable.
- Opennodes: this site features 64 hardware open slots to allow any one to plug his own hardware and benefits from the platform debugging and monitoring tools.

GANG Project-Team

6. New Software and Platforms

6.1. big-graph-tools

KEYWORD: Graph algorithmics

FUNCTIONAL DESCRIPTION: Gang is developing a software for big graph manipulation. A preliminary library offering diameter and skeleton computation. This library was used to compute the diameters of the worldwide road network (200M edges) and the largest strongly connected component of the Twitter follower-follower graph (23G edges).

- Contact: Laurent Viennot
- URL: <https://who.rocq.inria.fr/Laurent.Viennot/dev/big-graph-tools/>

6.2. GRPH

The high performance graph library for Java

KEYWORDS: Graph - Graph algorithmics - Java

FUNCTIONAL DESCRIPTION: Grph is an open-source Java library for the manipulation of graphs. Its design objectives are to make it portable, simple to use/extend, computationally/memory efficient, and, according to its initial motivation: useful in the context of graph experimentation and network simulation. Grph also has the particularity to come with tools like an evolutionary computation engine, a bridge to linear programming solvers, a framework for distributed computing, etc.

Grph offers a very general model of graphs. Unlike other graph libraries which impose the user to first decide if he wants to deal with directed, undirected, hyper (or not) graphs, the model offered by Grph is unified in a general class that supports mixed graphs made of undirected and directed simple and hyper edges. Grph achieves great efficiency through the use of multiple code optimization techniques such as multi-core parallelism, caching, adequate data structures, use of primitive objects, exploitation of low-level processor caches, on-the-fly compilation of specific C/C++ code, etc. Grph attempts to access the Internet in order to check if a new version is available and to report who is using it (login name and hostname). This has no impact whatsoever on performance and security.

- Participants: Aurélien Lancin, David Coudert, Issam Tahiri, Luc Hogie and Nathann Cohen
- Contact: Luc Hogie
- URL: <http://www.i3s.unice.fr/~hogie/grph/>

GANG Project-Team

6. New Software and Platforms

6.1. big-graph-tools

KEYWORD: Graph algorithmics

FUNCTIONAL DESCRIPTION: Gang is developing a software for big graph manipulation. A preliminary library offering diameter and skeleton computation. This library was used to compute the diameters of the worldwide road network (200M edges) and the largest strongly connected component of the Twitter follower-follower graph (23G edges).

- Contact: Laurent Viennot
- URL: <https://who.rocq.inria.fr/Laurent.Viennot/dev/big-graph-tools/>

6.2. GRPH

The high performance graph library for Java

KEYWORDS: Graph - Graph algorithmics - Java

FUNCTIONAL DESCRIPTION: Grph is an open-source Java library for the manipulation of graphs. Its design objectives are to make it portable, simple to use/extend, computationally/memory efficient, and, according to its initial motivation: useful in the context of graph experimentation and network simulation. Grph also has the particularity to come with tools like an evolutionary computation engine, a bridge to linear programming solvers, a framework for distributed computing, etc.

Grph offers a very general model of graphs. Unlike other graph libraries which impose the user to first decide if he wants to deal with directed, undirected, hyper (or not) graphs, the model offered by Grph is unified in a general class that supports mixed graphs made of undirected and directed simple and hyper edges. Grph achieves great efficiency through the use of multiple code optimization techniques such as multi-core parallelism, caching, adequate data structures, use of primitive objects, exploitation of low-level processor caches, on-the-fly compilation of specific C/C++ code, etc. Grph attempts to access the Internet in order to check if a new version is available and to report who is using it (login name and hostname). This has no impact whatsoever on performance and security.

- Participants: Aurélien Lancin, David Coudert, Issam Tahiri, Luc Hogie and Nathann Cohen
- Contact: Luc Hogie
- URL: <http://www.i3s.unice.fr/~hogie/grph/>

MARACAS Team

6. New Software and Platforms

6.1. cortexlab-fftweb

KEYWORDS: Experimentation - Data visualization - SDR (Software Defined Radio)

FUNCTIONAL DESCRIPTION: fftweb is a real-time spectral (FFT) visualization of one or several signal, embedded in a web page. The FFT is computed in a GNURadio block, then sent to a gateway server, which serves the web page, associated javascripts, and signal websockets. The end user only has to use the GNURadio block and the web page, and doesn't need to bother about the internal details of the system. fftweb has been developed specially for the CorteXlab testbed but with minor adaptations, it can be used in other contexts, and also can be used to draw more generic real-time graphs, not only FFTs. Technologies: GNURadio, python, python-gevent, Javascript, D3JS

- Contact: Matthieu Imbert

6.2. cortexlab-minus

KEYWORDS: Experimentation - SDR (Software Defined Radio)

FUNCTIONAL DESCRIPTION: Minus is an experiment control system able to control, the whole lifecycle of a radio experiment in CorteXlab or any other testbed inspired by it. Minus controls and automates the whole experiment process starting from node power cycling, experiment deployment, experiment start and stop, and results collection and transfer. Minus is also capable of managing multiple queues of experiments which are executed simultaneously in the testbed.

- Contact: Matthieu Imbert

6.3. cortexlab-webapp

KEYWORDS: Experimentation - SDR (Software Defined Radio)

FUNCTIONAL DESCRIPTION: User management module, which aims at easing platform usage and improving the metadata that we can associate with each experimenter and experiment. This metadata aims at improving the metrics we can gather about the platform's usage

- Partner: Insa de Lyon
- Contact: Pascal Girard

6.4. CorteXlab-IoT Framework

Framework for PHY-MAC layers Prototyping in Dense IoT Networks using CorteXlab Testbed

KEYWORDS: SDR (Software Defined Radio) - Iot - CorteXlab - GNU Radio

FUNCTIONAL DESCRIPTION: This framework was developed in the project "Enhanced Physical Layer for Cellular IoT" (EPHYL). It provides a customizable and open source design for IoT networks prototyping in a massive multi-user, synchronized and reproducible environment thanks to the hardware and software capabilities of the testbed.

- Author: Othmane Oubejja
- Contact: Othmane Oubejja
- Publication: [Framework for PHY-MAC layers Prototyping in Dense IoT Networks using FIT/CorteXlab Testbed](#)
- URL: <https://github.com/CorteXlab/gr-ephyl>

6.5. Platforms

6.5.1. FIT/CorteXlab

FIT (Future Internet of Things) is a french Equipex (Équipement d'excellence) which aims to develop an experimental facility, a federated and competitive infrastructure with international visibility and a broad panel of customers. FIT is composed of four main parts: a Network Operations Center (FIT NOC), a set of IoT test-beds (FIT IoT-Lab), a set of wireless test-beds (FIT-Wireless) which includes the FIT/CorteXlab platform deployed previously by the Socrate team and now managed by Maracas team in the Citi lab, and finally a set of Cloud test-beds (FIT-Cloud). In 2014 the construction of the room was done and SDR nodes have been installed in the room: 42 industrial PCs (Aplus Nuvo-3000E/P), 22 NI radio boards (usrp) and 18 Nutaq boards (PicoSDR, 2x2 and 4X4) can be programmed remotely, from internet now.

New features are under developments, and the platform is used or has been used by several research groups : i) Nokia Bell Labs - Inria common labs for research in machine learning, ii) ANR projects EPHYL and ARBURST for research on massive access for IoT, iii) ERC of prof. Michele Wigger, IMT Paris, for validating distributed coding strategies, iv) Nokia Bell Labs New Jersey for a proof of concept of wireless caching, Greentouch international consortium to develop a prototype of interference alignment.



Figure 3. FIT/CorteXlab facility

NEO Project-Team

6. New Software and Platforms

6.1. marmoteCore

Markov Modeling Tools and Environments - the Core

KEYWORDS: Modeling - Stochastic models - Markov model

FUNCTIONAL DESCRIPTION: marmoteCore is a C++ environment for modeling with Markov chains. It consists in a reduced set of high-level abstractions for constructing state spaces, transition structures and Markov chains (discrete-time and continuous-time). It provides the ability of constructing hierarchies of Markov models, from the most general to the particular, and equip each level with specifically optimized solution methods.

This software is developed within the ANR MARMOTE project: ANR-12-MONU-00019.

- Participants: Alain Jean-Marie, Hlib Mykhailenko, Benjamin Briot, Franck Quessette, Issam Rabhi, Jean-Marc Vincent and Jean-Michel Fourneau
- Partner: UVSQ
- Contact: Alain Jean-Marie
- Publications: [marmoteCore: a Markov Modeling Platform](#) - [marmoteCore: a software platform for Markov modeling](#)
- URL: <http://marmotecore.gforge.inria.fr/>

RESIST Team

6. New Software and Platforms

6.1. Distem

KEYWORDS: Large scale - Experimentation - Virtualization - Emulation

FUNCTIONAL DESCRIPTION: Distem is a distributed systems emulator. When conducting research on Cloud, P2P, High Performance Computing or Grid systems, it can be used to transform an homogenous cluster (composed of identical nodes) into an experimental platform where nodes have different performance, and are linked together through a complex network topology, making it the ideal tool to benchmark applications targetting such environments, or aiming at tolerating performance degradations or variations which are frequent in the Cloud or in other applications distributed at large scale (P2P for example).

RELEASE FUNCTIONAL DESCRIPTION: New features in Distem 1.3 include: (1) New network emulation parameters: loss, duplication, corruption, reordering and jitter, (2) Support for Debian Stretch, (3) Added many tests, (4) Moved project from GForge to GitHub (<https://github.com/madynes/distem>).

NEWS OF THE YEAR: New version 1.3

- Participants: Luc Sarzyniec, Lucas Nussbaum and Tomasz Buchert
- Partners: CNRS - Université de Lorraine - Loria - Grid'5000 - Inria
- Contact: Lucas Nussbaum
- URL: <http://distem.gforge.inria.fr>

6.2. Grid'5000

Grid'5000 testbed

KEYWORDS: HPC - Cloud - Big data - Testbeds

FUNCTIONAL DESCRIPTION: The Grid'5000 experimental platform is a scientific instrument to support computer science research related to distributed systems, including parallel processing, high performance computing, cloud computing, operating systems, peer-to-peer systems and networks. It is distributed on 10 sites in France and Luxembourg. Grid'5000 is a unique platform as it offers to researchers many and varied hardware resources and a complete software stack to conduct complex experiments, ensure reproducibility and ease understanding of results.

NEWS OF THE YEAR: This year's highlights include the TILECS workshop, and various improvements (update to Debian 10, several new clusters including the addition of 72 GPUs, etc.). More information on <https://www.grid5000.fr/w/News>

- Participants: Christian Pérez, David Loup, Frédéric Desprez, Laurent Lefèvre, Laurent Pouilloux, Marc Pinhède, Simon Delamare, Lucas Nussbaum, Teddy Valette and Alexandre Merlin
- Contact: Lucas Nussbaum
- URL: <https://www.grid5000.fr/>

6.3. SCUBA

A Tool Suite for the automated security assessment of IoT environments

KEYWORDS: Cybersecurity - Internet of things - Machine learning - Artificial intelligence

FUNCTIONAL DESCRIPTION: IoT devices are used in different fields of application, not only for the general public, but also in industrial environments. SCUBA is tool suite for the security assessment of industrial and general public IoT devices. It mainly relies on collected information through passive and active scanning of a running IoT device in its exploitation environment to build its Security Knowledge Base (SKB). The knowledge base contains all relevant information of the device regarding its network communications extracted from PCAP files, the enumeration of its used hardware and software represented in the CPE (Common Platform Enumeration) format, the list of its known vulnerabilities in the CVE (Common Vulnerabilities and Exposures) format associated to their CWE (Common Weakness Enumeration) and CAPEC (Common Attack Pattern Enumeration and Classification) descriptions. The SKB is used by SCUBA to predict the intrusion chains associated to an IoT device and its environment. SCUBA tries to be as automated as possible to face the large scale and the great heterogeneity of IoT networks.

NEWS OF THE YEAR: First release

- Participants: Abdelkader Lahmadi, Frédéric Beck, Thomas Lacour and Jérôme François
- Contact: Abdelkader Lahmadi

6.4. Platforms

6.4.1. CPS Security Assessment Platform

NEWS OF THE YEAR :

During 2019, we have extended our IoT (Internet of Things) and CPS (Cyber-Physical Systems) security assessment platform with more IoT devices dedicated to home networks (Alexa and Google Home voice assistants, smart door bell, smart door lock, alarm system). The platform is used for several demonstrations and it is extensively used for the development carried on the SCUBA (see 6.3) tool suite to automate the assessment of the security of IoT and SCADA systems by using ML/AI methods.

- Participants: Abdelkader Lahmadi, Frédéric Beck, Thomas Lacour and Jérôme François
- Contact: Abdelkader Lahmadi

SOCRATE Project-Team

4. New Software and Platforms

4.1. FloPoCo

Floating-Point Cores, but not only

KEYWORD: Synthesizable VHDL generator

FUNCTIONAL DESCRIPTION: The purpose of the open-source FloPoCo project is to explore the many ways in which the flexibility of the FPGA target can be exploited in the arithmetic realm.

- Participants: Florent de Dinechin and Luc Forget
- Partners: CNRS - ENS Lyon - UCBL Lyon 1 - UPVD
- Contact: Florent de Dinechin
- URL: <http://flopoco.gforge.inria.fr/>

4.2. Sytare

KEYWORDS: Embedded systems - Operating system - Non volatile memory

FUNCTIONAL DESCRIPTION: Sytare is an embedded operating system targeting tiny platforms with intermittent power. In order to make power failures transparent for the application, the system detects imminent failures and saves a checkpoint of program state to non-volatile memory. Hardware peripherals are also made persistent without requiring developer attention.

- Authors: Tristan Delizy, Gautier Berthou, Guillaume Salagnac, Kevin Marquet and Tanguy Risset
- Contact: Guillaume Salagnac
- Publication: [Peripheral State Persistence For Transiently Powered Systems](#)
- URL: <https://hal.inria.fr/hal-01460699>

4.3. NanoTracer

KEYWORDS: Embedded systems - Power monitoring - Low power consumption

FUNCTIONAL DESCRIPTION: NanoTracer is a high performance ammeter dedicated to power measurements for small devices. The system measures currents between 100nA and 100mA (gain is auto-adjusted dynamically) with a sampling frequency of 2Msps. Data is streamed to a PC over USB which enables long-running experiments, or just real-time visualization of data.

- Contact: Guillaume Salagnac
- URL: <https://gitlab.inria.fr/nanotracer/>

4.4. marto

Modern Arithmetic Tools

KEYWORDS: High-level synthesis - Arithmetic - FPGA

FUNCTIONAL DESCRIPTION: Marto provides C++ headers to implement custom sized arithmetic operators such as:

Custom sized posits and their environment (including the quire) Custom sized IEEE-754 numbers Custom sized Kulisch accumulators (and sums of products)

- Participants: Yohann Uguen, Florent de Dinechin and Luc Forget
- Contact: Yohann Uguen
- Publication: [hal-02130912v4](#)
- URL: <https://gitlab.inria.fr/lforget/marto>

4.5. hint

High-level synthesis Integer Library

KEYWORD: High-level synthesis

FUNCTIONAL DESCRIPTION: Hint is an header-only arbitrary size integer API with strong semantics for C++. Multiple backends are provided using various HLS libraries, allowing a user to write one operator and synthesize it using the main vendor tools.

- Participants: Yohann Uguen, Florent de Dinechin and Luc Forget
- Contact: Luc Forget
- Publication: [hal-02131798v2](#)
- URL: <https://github.com/yuguen/hint>

TRIBE Project-Team

5. New Software and Platforms

5.1. RIOT

KEYWORDS: Internet of things - Operating system - Sensors - Iot - Wireless Sensor Networks - Internet protocols

SCIENTIFIC DESCRIPTION: While requiring as low as 1,5kB of RAM and 5kB of ROM, RIOT offers real time and energy efficiency capabilities, as well as a single API (partially POSIX compliant) across heterogeneous 8-bit, 16-bit and 32-bit low-hardware. This API is developer-friendly in that it enables multi-threading, standard C and C++ application programming and the use of standard debugging tools (which was not possible so far for embedded programming). On top of this, RIOT includes several network stacks, such as a standard IPv6/6LoWPAN stack and an information-centric network stack (based on CCN).

FUNCTIONAL DESCRIPTION: RIOT is an Open Source operating system that provides standard protocols for embedded systems. RIOT allows, for example, the development of applications that collect sensor data and transmit it to a central node (e.g. a server). This data can then be used for smart energy management for instance.

RIOT is specially designed for embedded systems, which are strongly constrained in memory and energy. Further, RIOT can easily be ported to different hardware devices and follows the latest evolution of IP standards.

RIOT applications can readily be tested in the FIT IoT-Lab, which provides a large-scale infrastructure facility with 3000 nodes for testing remotely small wireless devices.

- Participants: Emmanuel Baccelli and Oliver Hahm
- Partners: Freie Universität Berlin - University of Hamburg
- Contact: Emmanuel Baccelli
- URL: <http://www.riot-os.org>

5.2. openschc

KEYWORDS: Internet of things - Internet protocols - Low-Power Wireless

FUNCTIONAL DESCRIPTION: OpenSCHC is a OpenSource Implementation of SCHC (Static Context Header Compression) currently being standardized by the LPWAN Working Group at the IETF. Oversimplifying, this is essentially IPv6 compression and fragmentation intended for low datarate, long range IoT networks.

The short/medium term goal is to organize SCHC Hackathons that occur at the IETF meetings.

The long term wish is to have a stable, open-source, reference codebase for the SCHC protocol (in Python).

The authors of OpenSCHC are listed here: <https://github.com/openschc/openschc/blob/master/AUTHORS.txt>

- Partner: OpenSCHC Authors
- Contact: Cédric Adjih
- URL: <https://github.com/openschc/openschc>

5.3. Gardinet

KEYWORD: Distributed networks

FUNCTIONAL DESCRIPTION: Gardinet (previously DragonNet) is a generic framework for network coding in wireless networks. It is an initial result of the GETRF project of the Hipercom2 team.

It is based on intra-flow coding where the source divides the flow in a sequence of payloads of equal size (padding may be used). The design keys of DragonNet are simplicity and universality, DragonNet does not use explicit or implicit knowledge about the topology (such as the direction or distance to the source, the loss rate of the links, ...). Hence, it is perfectly suited to the most dynamic wireless networks. The protocol is distributed and requires minimal coordination. DragonNet architecture is modular, it is based on 5 building blocks (LIB, SIG, Protocol, SEW and DRAGON). Each block is almost independent. This makes DragonNet generic and hence adaptable to many application scenarios. DragonNet derives from a prior protocol called DRAGONCAST. Indeed, DragonNet shares the same principles and theoretical overview of DRAGONCAST. It enriches DRAGONCAST by the information base and signaling required to perform broadcast in wireless networks and in wireless sensor networks in particular.

- Participants: Antonia Masucci, Cédric Adjih, Hana Baccouch and Ichrak Amdouni
- Contact: Cédric Adjih
- URL: <http://gitlab.inria.fr/gardinet>