RESEARCH CENTRE

Inria Saclay Centre at Institut Polytechnique de Paris

IN PARTNERSHIP WITH:

Institut Polytechnique de Paris, TELECOM SUDPARIS

2024

ACTIVITY REPORT

Project-Team

BENAGIL

Efficient and safe distributed systems

IN COLLABORATION WITH: Services répartis, Architectures, MOdélisation, Validation, Administration des Réseaux

DOMAIN

Networks, Systems and Services, Distributed Computing

THEME

Distributed Systems and middleware

Inria

Contents

Project-Team BENAGIL

Creation of the Project-Team: 2023 September 01

Keywords

Computer sciences and digital sciences

A1.1.1. – Multicore, Manycore

A1.1.4. – High performance computing

A1.1.13. – Virtualization

A1.3.5. – Cloud

Other research topics and application domains

B6.1.1. – Software engineering

1 Team members, visitors, external collaborators

Research Scientist

- Gael Thomas [Team leader, INRIA, Senior Researcher]

Faculty Members

- Mathieu Bacou [TELECOM SUDPARIS, Associate Professor]
- Elisabeth Brunet [TELECOM SUDPARIS, Associate Professor]
- Pierre Sutra [TELECOM SUDPARIS, Professor]
- Francois Trahay [TELECOM SUDPARIS, Professor]

Post-Doctoral Fellow

- Nicolas Derumigny [TELECOM PARIS, Post-Doctoral Fellow, from Feb 2024]

PhD Students

- Mickaël Boichot [CEA/DAM]
- Adam Chader [TELECOM SUDPARIS]
- Jean-Francois Dumollard [IP PARIS]
- Catherine Guelque [TELECOM SUDPARIS]
- Boubacar Kane [TELECOM SUDPARIS]
- Yohan Pipereau [TELECOM SUDPARIS]
- Marie Reinbigler [TELECOM SUDPARIS]
- Jules Risse [INRIA, from Nov 2024]
- Subashiny Tanigassalame [TELECOM SUDPARIS, until Aug 2024]
- Jana Toljaga [TELECOM PARIS]
- Lucas Van Lanker [CEA/DAM]
- Nevena Vasilevska [ECOLE POLY PALAISEAU, from Dec 2024]

Interns and Apprentices

- Jules Risse [TELECOM SUDPARIS, Intern, from Feb 2024 until Jul 2024]

Administrative Assistants

- Natalia Alves [INRIA]
- Julienne Moukalou [INRIA]

External Collaborator

- Valentin Honore [ENSIIE]

2 Overall objectives

Distributed systems are pivotal to many applications used in our daily life: AI, data analytics, online gaming, social networks, web services, healthcare, etc. Because they have to sustain massive workloads, these systems scatter computation across many units, which coordinate to store the input, execute the calculus and return results in a usable manner to the application. Inefficiencies in these infrastructures hinder the ability to handle large computations. They also lead to wasting energy and hardware resources. Errors at runtime may result in painful data losses and exploitable security loopholes. As a consequence, designing and implementing such systems in an efficient and safe manner is essential, and it has a strong commitment from all the major IT industries.

The Benagil team works on the design and implementation of more efficient and safer distributed systems. For that, the Benagil team focuses on the core system components at the frontier with the hardware: hypervisors, operating systems, language runtimes, storage systems and communication libraries. Improving the efficiency and safety of distributed systems is a challenging task. Modern distributed systems manage large pools of machines, a plethora of users and they process very large datasets. Consequently, they are inherently complex and both their design and implementation is notoriously hard. Complexity arises from the software stack, the algorithms at the core of these systems, as well as the hardware itself:

- System software level. A typical modern computer system runs many software components: hypervisors (e.g., KVM/Qemu), operating systems (e.g., Linux), container systems (e.g., Linux containers), language runtimes (e.g., the Java virtual machine) and specialized runtimes for HPC (e.g., MPI), data analytics (e.g., Spark) or AI (e.g. PyTorch). Such software are today very large. For instance, the last version of the Linux kernel runs over 22,000,000 lines of code.

- Distributed system level. As pointed above, modern systems are distributed, involving many machines. These machines are connected with heterogeneous networks, ranging from fast local networks (e.g., Infiniband or Ethernet 10Gb) to high-latency planet-scale connections. Many of these systems have to be highly available, that is they need to be responsive 99.999% of the time. This requires to use complex monitoring mechanisms and replication algorithms that solve trade-offs between availability and performance. Distributed systems need also to do fine-grained task and data placement choices. They aggregate resources, have to use them efficiently, and provide high-enough isolation levels between the multiple applications using them.

- Machine level. Internally, each machine is a very complex entity. It is today composed of multiple processors, memory banks and devices inter-connected with a complex network. A processor contains tens of cores with finely tunable cache hierarchies and out-of-order execution pipelines. Each core is a very dense unit of calculus, as testified by the specification of Intel Skylake that covers more than 4,800 pages. A machine also often includes multiple heterogeneous accelerators and specialized hardware such as persistent memory that provides durability at the nanosecond scale, GPUs specialized for massively parallel computations, FPGAs used to offload complex computations from the CPUs, and TPUs specialized in deep neural network computation. Accessing all these components is not uniform both in terms of bandwidth and latency. Heterogeneity must be taken into account at multiple levels of the system stack. This makes data access optimization especially challenging. This complexity also opens security breaches, such as cache timing attacks, code timing attacks and data access pattern attacks. Preventing these attacks requires to solve complex trade-offs between performance, security and usability.

The inherent complexity of distributed systems makes analyzing their performance and safety difficult. This difficulty is increased by complex and unexpected interactions between software and hardware components. Besides that, understanding and improving the system components in the context of distributed systems require an expertise in many areas: hypervisors, operating systems, containerization, language runtimes, compilation, network, architecture, web, databases, data analytics runtimes, cloud runtimes and distributed algorithms. As an example, in a previous work, we observed a large performance degradation in a data analytics application written in Scala (namely, PageRank in Apache Spark). This phenomenon was caused by a bad memory placement performed by the Java virtual machine on a non-uniform memory architecture. This issue was also reinforced by the use of a (system) virtual machine

that blindly allocates memory from any memory bank. Another source of inefficiencies was due to the hypervisor which was continuously moving memory without telling the virtual machine. All in all, understanding and solving the performance bottleneck at each level of the system stack took us 8 years. It involved 3 PhD students and 6 researchers with expertise in different system areas.

3 Research program

The Benagil team works on improving the performance and the safety of the core system components of the distributed systems. In order to achieve this goal, we propose a systematic approach. This approach first consists in profiling and analyzing current distributed systems to identify their limits in term of efficiency and/or safety when they execute large distributed applications. Then, building upon this analysis, we develop new algorithms, mechanisms and components to improve them.

The Benagil team is structured along three main axes which articulate the above approach. The first axis is devoted to performance profiling and analysis. In this axis, we introduce new tools and techniques to automatically analyze the performance of a large distributed system. Based on this analysis, we identify performance issues, which we use as input in the two other axes to improve performance. The two other axes study two aspects of the system components. In the system components for cloud infrastructure axis, we devise new system techniques to improve the performance and safety of two core system components used in cloud infrastructure: virtualization and storage. In the system components for emerging computing models, we propose new system mechanisms and interfaces for two pivotal upcoming programming models: serverless and edge computing.

3.1 Performance analysis

Due to the high complexity of modern large-scale distributed applications, understanding performance problems is a tedious task even for the most experienced programmers. A performance bottleneck may arise from different interactions, between hardware and software, or between different software components. Even just a single contended lock, or a falsely shared cache line, in one of the system components may lead to a dramatic slowdown.

Because of this complexity, manually identifying the root cause of a performance bottleneck is notoriously difficult. In this axis, we propose to help the developer by designing new profiling tools able to handle the complexity of hardware and software stacks, and able to scale with the size of the system.

3.2 System components for the cloud

In this axis, we aim at studying and designing the next generation of systems for cloud infrastructures. Today, these infrastructures are undergoing major changes at the hardware level with the generalization of ultra-fast networks at the micro-second scale (e.g., RDMA) and storage devices (e.g., NVMe or Non-Volatile Memory). Their joint arrivals require to radically revisit the way we design two core system components of any cloud infrastructure: the *virtualization system* and the *storage system*.

3.3 System components for emerging computing models

At a higher level of the system stack, we are witnessing the arrival of two new computing models: *serverless computing* and *edge computing*. These computing models deeply change the assumptions under which the current system components were built. Current system components assume long-running applications and powerful computing infrastructures. However, this is no more the case with these two new computing models. In serverless computing, applications are split into short-lived tasks. In edge computing, applications execute at the border of the network, atop low performance hardware.

4 Application domains

Overall, the Benagil team is mostly specialized on the low-level components of distributed systems. This specialization is at the frontier of security, hardware, high-performance computing (HPC), machine

learning, data analytics and databases. With respect to security, the team studies some system aspects, such as trusted execution environments (e.g., Intel SGX) to protect applications, or data replication to improve availability. However, the Benagil team is not a security one per se. Regarding hardware, the Benagil team has a strong background in using modern hardware such as persistent memory or GPU. This knowledge is crucial to efficiently use the hardware in system components. However, the team is only consuming hardware and does not directly design it. This is also the case with HPC, machine learning and data analytics. The Benagil team understand the system requirements of these highly-demanding applications, and use them to benchmark their system components. However the team only rarely contribute to these runtimes themselves. The Benagil team has also a strong knowledge regarding the storage system components used in databases. This includes the algorithmic and implementation concerns related to data distribution, consistency, replication and persistence. However, the Benagil team is not specialized in database in general.

5 Highlights of the year

- Pierre Sutra defended his HDR in January, and was promoted *Professeur* at Télécom SudParis in November 2024.
- Two PhD students of the team defended their PhD in 2024: Subashiny Tanigassalame and Boubacar Kane.

6 New software, platforms, open data

6.1 New software

6.1.1 EZTrace

Keywords: MPI communication, Execution trace, Traces, High performance computing, Performance analysis, HPC, OpenMP, CUDA

Functional Description: The improvement of the performances of parallel applications (numerical simulation for example) is an important phase of the development. For that it is necessary to detect the various phases of the application and to understand the performances of them.

The automatic generation of traces of execution makes it possible the developer to quickly detect simply and the various phases of the application and to understand the behavior of it.

URL: https://gitlab.com/eztrace/eztrace

Publications: hal-01257904v1, hal-00707236v1, hal-03215663v1, hal-03276036v1, hal-02179717v1, inria-00587216v1, hal-00918733v1, hal-00865845v1, tel-03278305v1

Contact: Francois Trahay

Participants: Francois Trahay, Valentin Honore

6.1.2 Pallas

Keywords: Performance analysis, HPC, High performance computing, Execution trace

Functional Description: Pallas is a generic trace format tailored for conducting various post-mortem performance analyses of traces describing large executions of HPC applications. During the execution of the application, Pallas collects events and detects their repetitions on-the-fly. When storing the trace to disk, PALLAS groups the data from similar events or groups of events together in order to later speed up trace reading. The Pallas format allows faster trace analysis compared to other trace formats.

URL: https://gitlab.inria.fr/pallas/pallas

Contact: Francois Trahay

Participant: Valentin Honore

6.1.3 numamma

Keywords: NUMA, Memory Allocation, Profiling

Functional Description: NumaMMa is both a NUMA memory profiler/analyzer and a NUMA application execution engine. The profiler allows to run an application while gathering information about memory accesses. The analyzer visually reports information about the memory behavior of the application allowing to identify memory access patterns. Based on the results of the analyzer, the execution engine is capable of executing the application in an efficient way by allocating memory pages in a clever way.

URL: https://github.com/numamma/numamma

Publications: cea-01854072v2, tel-03278305v1

Contact: Francois Trahay

Participant: Francois Trahay

6.1.4 ForkNox

Name: ForkNox: a micro-hypervisor to protect Linux

Keywords: Virtualization, Security

Functional Description: ForkNox is a micro-hypervisor designed to protect Linux. By leveraging virtualization techniques, ForkNox can revoke read, write, and execute permissions for specific memory regions of Linux. This ensures that, even if Linux is under attack, the attacker cannot modify those parts of the system.

Release Contributions: Initial version of the software.

News of the Year: Initial version of the software.

URL: https://gitlab.inria.fr/benagil/fkx/fork-nox

Contact: Gael Thomas

Participants: Gael Thomas, Jean-Francois Dumollard, Mathieu Bacou, Nicolas Derumigny

6.1.5 VoliMem

Name: VoliMem: a lightweight virtualization for processes

Keyword: Virtualization

Functional Description: VoliMem is a small library that remaps a native process inside a virtual machine. Thanks to this, the process gains access to low-level system hardware primitives, such as a page table in user space or fast inter-processor interrupts.

Release Contributions: Initial version of the prototype

News of the Year: Initial implementation of the software.

URL: https://gitlab.inf.telecom-sudparis.eu/VoliMembers/libvolimem

Contact: Gael Thomas

Participants: Gael Thomas, Nicolas Derumigny, Jana Toljaga

6.1.6 Tele-GC

Name: Tele-GC: a garbage collector for disaggregated memory

Keywords: Garbage Collection, Java, Disaggregated memory

Functional Description: Tele-GC is a garbage collector specifically designed for disaggregated memory. It runs the application on the compute node while the garbage collector operates on the memory node. Tele-GC leverages the discrepancy between the cache on the compute node and the memory on the memory node to avoid any synchronization during a collection.

URL: https://github.com/Adchad/RemoteSpace

Contact: Gael Thomas

6.1.7 FaaSLoad

Keywords: Cloud computing, Serverless, Function-as-a-Service, Measures, Resource utilization, Workload injection, Performance measure

Scientific Description: FaaSLoad is a tool to gather fine-grained data about performance and resource usage of the programs that run on Function-as-a-Service cloud platforms. It considers individual instances of functions to collect hardware and operating-system performance information, by monitoring them while injecting a workload. FaaSLoad helps building a dataset of function executions to train machine learning models, studying at fine grain the behavior of function runtimes, and replaying real workload traces for in situ observations.

Functional Description: Invoke functions in a Function-as-a-Service platform, and gather data about their performance and their resource usage to understand their behavior in Serverless environments.

Release Contributions: Stabilization and opening to outsiders.

News of the Year: Release of public version 2.0 (and then 2.1.0), the first mature and useful to outsiders. Published in a dedicated scientific paper at OPODIS'24.

URL: https://gitlab.com/faasload/faasload

Publications: hal-03211416, hal-04886267

Contact: Mathieu Bacou

Participant: Mathieu Bacou

7 New results

This year, the Benagil team carried out research projects on the three axes. In the performance analysis axis, the Benagil team studied two topics: (i) optimizing the representation of a performance trace in order to improve analysis time (see Section 7.1.1), and (ii) predicting the speedup that can be expected by upgrading a GPU (see Section 7.1.2).

In the system components for cloud infrastructures axis, the Benagil team studied four topics: (i) easing the development of applications that enforce privacy in a confidential computing context (see Section 7.2.1), (ii) optimizing a consensus algorithm by leveraging programmable switches (see Section 7.2.2), (iii) optimizing a consensus algorithm using a new principle called double voting (see Section 7.2.3), and (iv) combining two communication primitives into a single one, called generic multicast (see Section 7.2.4).

In the system components for emerging computing models axis, the Benagil team worked on one topic, the implementation of a tool that generates representative benchmarks in the Function-as-a-Service context.

7.1 Performance analysis

7.1.1 Scalable trace format

Participants: Catherine Guelque, Valentin Honoré, Philippe Swartvegher *(Inria TOPAL)*, François Trahay.

Identifying performance bottlenecks in a parallel application is tedious, especially because it requires analyzing the behavior of various software components, as bottlenecks may have several causes and symptoms. Detecting a performance problem means investigating the execution of an application and applying several performance analysis techniques. To do so, one can use a tracing tool to collect information describing the behavior of the application. At the end of the execution, a trace file in a specific format is available to the application user, which can be used to conduct a complete post-mortem investigation. When analyzing the performance of application running at a large scale, the post-mortem analysis needs to load thousands of trace files in memory, and process them. This quickly becomes impractical for large scale applications, as memory gets exhausted and the number of opened files exceeds the system capacity.

As part of the Exa-SofT project, Catherine Guelque proposes Pallas, a generic trace format tailored for conducting various post-mortem performance analyses of traces describing large executions of HPC applications. During the execution of the application, Pallas collects events and detects their repetitions on-the-fly. When storing the trace to disk, Pallas groups the data from similar events or groups of events together in order to later speed up trace reading. We conducted large-scale experiments on the Jean-Zay supercomputer to evaluate Pallas. Our experiments show that the Pallas format allows faster trace analysis compared to other evaluated trace formats. Overall, the Pallas trace format allows an interactive analysis of a trace that is required when a user investigates a performance problem. These results are to appear at IPDPS'25.

7.1.2 Performance prediction

Participants: Lucas Van Lanker, Hugo Taboada *(CEA/DAM)*, Elisabeth Brunet, François Trahay.

With the advent of heterogeneous systems that combine CPUs and GPUs, designing a supercomputer becomes more and more complex. The hardware characteristics of GPUs significantly impact the performance. Choosing the GPU that will maximize performance for a limited budget is tedious because it requires predicting the performance on a non-existing hardware platform.

Lucas Van Lanker's PhD explores means for predicting the performance of kernels running on GPUs [7]. We propose a methodology that analyzes the behavior of an application running on an existing platform, and projects its performance on another GPU based on the target hardware characteristics. The performance projection relies on a hierarchical roofline model as well as on a comparison of the kernel's assembly instructions of both GPUs to estimate the operational intensity of the target GPU. Our experiments show that the performance can be predicted accurately at a low cost.

7.2 System components for the cloud

7.2.1 Privagic: code partitioning for confidential computing made easy

Participants: Subashiny Tanigassalame, Yohan Pipereau *(Benagil, defended in 2023)*, Adam Chader, Jana Toljaga, Gaël Thomas.

Partitioning a multi-threaded application between a secure and a non-secure memory zone remains a challenge. The current tools rely on data flow analysis techniques, which are unable to handle multi-threaded C or C++ applications. To avoid this limitation, in the PhD thesis of Subashiny Tanigassalame,

we propose to trade the ease-of-use of data flow analysis for another language construct: explicit secure typing. With secure typing, as with data flow analysis, the developer annotates memory locations that contain sensitive values. However, instead of analyzing how the sensitive values flow, we propose to use these annotations to only check typing rules, such as ensuring that the code never stores a sensitive value in an unsafe memory location. By avoiding data flow analysis, the developer has to annotate more memory locations, but the partitioning tool can handle multi-threaded C and C++ applications.

We implemented our explicit secure typing principle in a compiler named Privagic [10, 9]. Privagic takes a legacy application enriched with secure types as input. It outputs an application partitioned for Intel SGX. Our evaluation with micro- and macro-applications shows that (i) explicit secure typing can handle multi-threaded C and C++ applications, (ii) adding explicit secure types requires a modest engineering effort of less than 10 modified lines of codes in our use cases, (iii) using explicit secure typing is more efficient than embedding a complete application in an enclave both in terms of performance and security in our use cases.

7.2.2 Consensus over RDMA at line speed

Participants: Rémi Dulong *(Benagil, defended in 2023)*, Nathan Felber *(Université de Neuchâtel)*, Pascal Felber *(Université de Neuchâtel)*, Gilles Hopin *(IP Paris)*, Baptiste Lepers *(The university of Sydney)*, Valerio Schiavoni *(Université de Neuchâtel)*, Gaël Thomas, Sébastien Vaucher *(Swiss Federal Laboratories for Materials Science and Technology)*.

P4CE is the first replication protocol that exhibits the same latency and requires the same network capacity as sending data to a single server [6]. P4CE builds upon previous RDMA-based consensus protocols. They achieve consensus with a single network round-trip, but with a reduced network throughput. P4CE also achieves consensus with a single round-trip, but without degrading throughput by decoupling the consensus decisions from the RDMA communications. The decision part of the consensus protocol runs on a commodity server, but the communication part of P4CE is fully implemented on a programmable switch, which replicates data and aggregates the acknowledgements in the network, avoiding the throughput bottleneck at the leader. Although simple in its principle, the implementation of P4CE raises many challenging issues, notably caused by the complexity of RDMA and the underlying network protocols, the intricacies of packet rewriting during replication and aggregation, and the restricted set of operations that can be implemented at wire speed in the programmable switch. We implemented P4CE and deployed it on a commerciallyavailable Intel Tofino switch, achieving up to 4× better throughput and better latency than state-of-the-art consensus protocols.

7.2.3 SwiftPaxos: Fast Geo-Replicated State Machines

Participants: Fedor Ryabinin *(IMDEA Software Institute)*, Alexey Gotsman *(IMDEA Software Institute)*, Pierre Sutra.

Cloud services improve their availability by replicating data across sites in different geographical regions. A variety of state-machine replication protocols have been proposed for this setting that reduce the latency under workloads with low contention. However, when contention increases, these protocols may deliver lower performance than Paxos.

This work [8] introduces SwiftPaxos—a protocol that lowers the best-case latency in comparison to Paxos without hurting the worst-case one. SwiftPaxos executes a command in 2 message delays if there is no contention, and in 3 message delays otherwise. To achieve this, the protocol allows replicas to vote on the order in which they receive state-machine commands. Differently from previous protocols, SwiftPaxos permits a replica to vote twice: first for its own ordering proposal, and then to follow the leader. This mechanism avoids restarting the voting process when a disagreement occurs among replicas, saving

computation time and message delays. Our evaluation shows that the throughput of SwiftPaxos is up to 2.9x better than state-of-the-art alternatives.

7.2.4 Generic Multicast

Participants: José Augusto Bolina *(Red Hat, Inc.)*, Douglas Antunes Rocha *(Federal University of Uberlândia)*, Lasaro Camargos *(Federal University of Uberlândia)*, Pierre Sutra.

Communication primitives play a central role in modern computing. They offer a panel of reliability and ordering guarantees for messages, enabling the implementation of complex distributed interactions. In particular, atomic broadcast is a pivotal abstraction for implementing fault-tolerant distributed services. This primitive allows disseminating messages across the system in a total order. There are two group communication primitives closely related to atomic broadcast. Atomic multicast permits targeting a subset of participants, possibly stricter than the whole system. Generic broadcast leverages the semantics of messages to order them only where necessary (that is when they conflict).

In this work, we propose to combine all these primitives into a single, more general one, called generic multicast. We formally specify the guarantees offered by generic multicast and present efficient algorithms. Compared to prior works, our solutions offer appealing properties in terms of time and space complexity. In particular, when a run is conflict-free, that is no two messages conflict, a message is delivered after at most three message delays.

This work has received the *best paper award* at the 13th Latin-American Symposium on Dependable and Secure Computing (LADC 2024) [5].

7.3 System components for emerging computing models

7.3.1 Fine-Grained Performance and Resource Measurement for Function-As-a-Service

Participants: Mathieu Bacou.

Cloud computing relies on a deep stack of system layers: virtual machine, operating system, distributed middleware and language runtime. However, those numerous, distributed, virtual layers prevent any low-level understanding of the properties of FaaS applications, considered as programs running on real hardware. As a result, most research analyses only consider coarse-grained properties such as global performance of an application, and existing datasets include only sparse data.

FaaSLoad is a tool to gather fine-grained data about performance and resource usage of the programs that run on Function-as-a-Service cloud platforms [4]. It considers individual instances of functions to collect hardware and operating-system performance information, by monitoring them while injecting a workload. FaaSLoad helps building a dataset of function executions to train machine learning models, studying at fine grain the behavior of function runtimes, and replaying real workload traces for in situ observations.

This research software project aims at being useful to cloud system researchers with features such as guaranteeing reproducibility and correctness, and keeping up with realistic FaaS workloads. FaaSLoad helps understanding the properties of FaaS applications, and studying the latter under real conditions.

8 Bilateral contracts and grants with industry

8.1 Bilateral contracts with industry

Participants: Mickaël Boichot, Lucas Van Lanker.

- Contract with CEA for the PhD of Mickaël Boichot (2021-2025), and Lucas Van Lanker (2024-2027)

9 Partnerships and cooperations

9.1 Other european programs/initiatives

CHIST-ERA Redonda

Participants: Guillermo Toyos Marfurt, Pierre Sutra.

Partners: Mines-Télécom, IMDEA Software Institute, University of London, University of Surrey, University of Neuchâtel.

Coordinator: Institut Mines-Télécom

Funding: 319 k€

Date: 2024-2027

Summary: The Redonda project's ambition is to design a next-generation replication protocol for blockchain. To achieve this, the project taps into recent advances in networking, secure computing and distributed systems. At the scale of a datacenter, the protocol relies on two recent technologies: RDMA and TEE. Both technologies are leveraged to create a sub-microsecond consensus layer that tolerates Byzantine failures. TEEs are also used in a novel upgradable and portable smart contract engine to execute blockchain transactions across a variety of infrastructures and hardware. Between datacenters, the protocol relies on leaderless state-machine replication. This recent approach decomposes transaction ordering into two sub-tasks that can execute in parallel, without a central coordinator to bottleneck the system. To ensure security and safety at runtime, the Redonda project creates the blockchain protocol by composing mechanically-verified building blocks. The new blockchain protocol is assessed using real hardware against benchmarks and publicly available traces. We target that it scales across hundreds of geo-distributed nodes while offering 100k+ transactions per second and split-second latency.

9.2 National initiatives

PEPR NumPex – Exa-SofT

Participants: Catherine Guelque, Élisabeth Brunet, Valentin Honoré, François Trahay.

Partners: Université Paris Saclay, Télécom SudParis, CEA, CNRS, Inria

Date: 2023-2028

Summary: Though significant efforts have been devoted to the implementation and optimization of several crucial parts of a typical HPC software stack, most HPC experts agree that exascale supercomputers will raise new challenges, mostly because the trend in exascale compute-node hardware is toward heterogeneity and scalability: Compute nodes of future systems will have a combination of regular CPUs and accelerators (typically GPUs), along with a diversity of GPU architectures. Meeting the needs of complex parallel applications and the requirements of exascale architectures raises numerous challenges which are still left unaddressed. As a result, several parts of the software stack must evolve to better support these architectures. More importantly, the links between these parts must be strengthened to form a coherent, tightly integrated software suite. Our project aims at consolidating the exascale software ecosystem by providing a coherent, exascale-ready software stack featuring breakthrough research advances enabled by multidisciplinary collaborations between researchers. The main scientific challenges we intend to address are: productivity, performance portability, heterogeneity, scalability and resilience, performance and energy efficiency.

PEPR Cloud – DiVa

Participants: Jana Toljaga, Nevena Vasilevska, Mathieu Bacou, Nicolas Derumigny, Gaël Thomas.

Partners: LIP6, LIG, IRIT, Inria Paris, Benagil/Telecom SudParis
Coordinator: Gaël Thomas, Télécom SudParis
Funding: 864 k€
Date: 2023-2030
Summary: The DiVa project investigates new virtualization mechanisms tailored for a disaggregated infrastructure and for an infrastructure composed of small edge infrastructures connected to powerful data centers. In the context of a disaggregated cloud, the DiVa project will focus on the virtualization interfaces, the scheduling, the use of programmable networks, and replication mechanisms. In the context of the continuum between the edge and the cloud, the DiVa project will focus on migration between heterogeneous machines, edge/edge and edge/data center network optimizations, and virtualization interfaces for micro virtual machines.

PEPR Cloud – Archi-CESAM

Participants: Jean-François Dumollard, Mathieu Bacou, Nicolas Derumigny, Gaël Thomas.

Partners: Université de Rennes, Benagil/Telecom SudParis, Institut Polytechnique de Grenoble, CEA, Inria
Coordinator: Denis Dutoit, CEA
Funding: 580 k€
Date: 2023-2030
Summary: European sovereignty in the cloud also means sovereignty over hardware, especially processors and accelerators. Dennard's Law is now over and Moore's Law is slowing down. In this technological context, which will continue, the improvement of processor performance will require hardware architectures that evolve towards more parallelism (multi-core), more specialization (accelerators), towards a closer relationship between computing and memory and new types of interconnections between components. On the other hand, by dissociating hardware resources (computing, memory, interconnection) from logical resources, virtualization facilitates the deployment of converged architectures that bring together the computing, storage and network infrastructure. The cloud gains in modularity, speed and agility for the deployment of new services with optimal use of resources. Hardware disaggregation on the one hand and resource virtualization on the other are making the intermediate adaptation layer increasingly complex, difficult to validate and prone to failure. The Archi-CESAM project proposes to rethink the hardware (computing, memory and interconnection) so that it is co-designed with the application in a perspective of converged architecture and trust, in an environment known for its abundance of data to be processed. The Archi-CESAM project addresses this major evolution of the Cloud in a global and coordinated approach between distributed architectures, acceleration, interconnection and security bricks, without forgetting the design methods.

ANR PRC – FrugalDinet

Participants: Gaël Thomas.

Partners: LIP6, LISTIC, Benagil/Inria Saclay, New-York University Shanghai
Coordinator: Pierre Sens, LIP6
Funding: 171 k€
Date: 2024-2028

Summary: In recent years, innovative hardware technologies have emerged to enhance distributed computations in datacenters. Programmable switches enable packet processing with user-defined functionality on packets in transit. Similarly, SmartNIC DPUs offload data-centric computations from host CPUs. Simultaneously, the urgency of climate and energy crises has emphasized the need for frugal architectures. These technologies present an opportunity to reduce overall network traffic from distributed services, offloading computations from CPUs to the network itself. They should be integrated in designing fundamental distributed system components like failure detectors, group membership, reliable broadcast, or consensus. We propose FrugalDinet a framework to build reliable, low-cost distributed services, leveraging these technologies which minimizes CPU usage in datacenters and subsequently their energy consumption. Our holistic approach extends key algorithms such as leader election, group membership and broadcasting, necessary for the creation of reliable services. We intend not only to offload algorithmic logics on network elements, but also to make opportunistic use of the information available at the switch level. We also plan to introduce a new high-level programming language facilitating transparent utilization of these frugal, reliable distributed services. The implemented frugal algorithms and programming abstractions will be applied to design a distributed transaction system

ANR PRCE – Centeanes

Participants: Pierre Sutra.

Partners: Télécom SudParis, Université Paris Cité, École Polytechnique, Université de Paris 6.
Coordinator: Pierre Sutra
Funding: 196 k€
Date: 2025-2029
Summary: Cloud computing of the past was concerned with the management of infrastructure resources, e.g., servers, VMs or containers. Today, *serverless computing* promises to abstract this worry away. In this new paradigm, the quantum of computation is the *function*; a function-as-a-service platform automatically manages deployment of functions, executing them on demand and at scale. This greatly simplifies access to the cloud, letting the application developer focus on getting the application code right, and ignore infrastructure issues.

Unfortunately, serverless computing remains difficult to use and to reason about. Indeed, the serverless environment is inherently unpredictable and non-deterministic, making it hard to understand and to control. Being distributed, serverless must cope with concurrency, unpredictable failures, or impossibility of consensus. On top of that, serverless poses more, new challenges to the application programmer. Events may trigger the same function invoked multiple times and/or terminate it before it has finished. Functions are stateless, starting from afresh every time; but often it must access an external storage service, thus being exposed to stale or inconsistent state. Finally, existing platforms suffer from inefficiencies, such as excessive data movement or random placement.

The Centeanes project aims to address these challenges from the perspectives of correctness, efficiency, and expressivity, in a real application context. It will develop tools for specifying, programming and running correct-by-design serverless applications. In detail, we propose a formal framework to study the foundations of serverless computing, including function composition and fault-tolerance. This framework is implemented in a lightweight runtime environment, where stateful operations and data locality are first class citizen. We also construct a toolchain to program and verify serverless applications executing in the runtime. This verification toolchain simplifies the programming of applications and helps enforce their correctness. The design is informed by, and will be validated against, benchmarks and full-scale industrial cloud or edge applications built with Eclipse Zenoh.

ANR PRC – Maplurinum

Participants: Adam Chader, Mathieu Bacou, Gaël Thomas.

Partners: INPG, Inria Rennes, CEA, Benagil/Telecom SudParis
Coordinator: Gaël Thomas, Telecom SudParis
Funding: 184 k€
Date: 2021-2025
Summary: High-Performance architectures are increasingly heteregenous and incorporate often specialized hardware. We have first seen the generalization of GPUs in the most powerful machines, followed a few years later by the introduction of FPGAs. More recently we have seen nascence of many other accelerators such as tensor processor units (TPUs) for DNNs or variable precision FPUs. Recent hardware manufacturing trends make it very likely that specialization will not only persist, but increase in future supercomputers. Because manually managing this heterogeneity in each application is complex and not maintainable, we propose in this project to revisit how we design both hardware and operating systems in order to better hide the heterogeneity to supercomputer users. In summary, we propose to rethink the hardware/software boundary in order to hide the heterogeneity behind a common minimal instruction set and a unified address space.

10 Dissemination

10.1 Promoting scientific activities

10.1.1 Scientific events: organisation

Member of the organizing committees

- François Trahay: participation to the organization of the Per3S workshop as part of the steering committee;
- Gaël Thomas: head of the steering committee of the Compas conference;
- Mathieu Bacou: participation to the organization of the thematic day ‘Virtualization of Systems and Networks’ of the CNRS’s GDR RSD.

10.1.2 Scientific events: selection

Chair of conference program committees

- François Trahay: Chair of the parallelism track for Compas 2024
- François Trahay: PC Chair of the system programming track for Cluster 2024

Member of the conference program committees

- Gaël Thomas: member of the Usenix ATC ’25, Eurosys ’25 and SoCC ’24 program comittees.
- Pierre Sutra: member of the Eurosys ’24, Middleware ’24, OPODIS ’24 and NETYS ’24 program comittees.

10.1.3 Invited talks

- François Trahay: Grenoble university seminar
- Gaël Thomas: Journées scientifiques de Inria, invited talk in the CORSE team (Inria Grenoble)

10.1.4 Research administration

- François Trahay: head of research action "Energy Efficiency" of the Energy4Climate interdisciplinary center.
- Mathieu Bacou: co-head of Working Group ‘Virtualization’ of CNRS’s GDR RSD.

10.2 Teaching - Supervision - Juries

10.2.1 Teaching

- Master: Pierre Sutra and Gaël Thomas are the heads of the Parallel & Distributed Systems master track at Institut Polytechnique de Paris
- Master: François Trahay is the head of the master of Computer Science at Institut Polytechnique de Paris
- Engineering: Élisabeth Brunet is in charge of the AI 3rd year track at Télécom SudParis
- Engineering: Pierre Sutra is in charge of the ASR 3rd year track at Télécom SudParis

10.2.2 Supervision

Phd in progress:

- Mickaël Boichot, "Caracterizing parallel applications for porting to multi-GPUs systems", supervised by P. Carribault, and E. Brunet
- Adam Chader, "Large-scale garbage collectors", supervised by G. Thomas, and M. Bacou
- Jean-Francois Dumollard, "Virtualization techniques to enforce the security of an operating system", supervised by G. Thomas, M. Bacou and N. Derumigny
- Catherine Guelque, "Large scale performance analysis", supervised by F. Trahay, and V. Honoré
- Marie Reinbigler, "Consensus over RDMA at Line Speed", supervised by C. Fetita, and E. Brunet
- Jules Risse, "Fine-grain energy consumption measurement", supervised by F. Trahay, and A. Guermouche
- Jana Toljaga, "Virtualization techniques for persistent memory", supervised by G. Thomas, M. Bacou and N. Derumigny
- Guillermo Toyos Marfurt, "A Next-Generation State-Machine Replication Protocol for Blockchain", supervised by P. Sutra and P. Kuznetsov (not yet in the database)
- Lucas Van Lanker, "Performance projection of GPU applications", supervised by F. Trahay, E. Brunet, and H. Taboada
- Nevena Vasilevska, "Hardware cache controlled by software for memory disaggregation", supervised by G. Thomas, J. Dumas, and N. Derumigny

Defended Phd:

- Boubacar Kane, "Adjusted Objects: An Efficient and Principled Approach to Scalable Programming", supervised by P. Sutra
- Subashiny Tanigassalame, "Privagic : confidential computing made practical with secure typing", supervised by G. Thomas (defended)

10.2.3 Juries

- François Trahay :
  - Reviewer of the PhD of Alexis Bandet, U. Bordeaux
  - Reviewer of the HDR of Baptiste Lepers, U. Grenoble
  - President of the PhD committee for Boubacar Kane, Institut Polytechnique de Paris
- Gaël Thomas :

  - Reviewer of the PhD of Pierre Jacquet, U. Lille
  - Reviewer of the PhD of Yasmine Djebrouni, U. Grenoble
  - Examiner for the PhD of Quentin Ducasse, ENSTA Bretagne
  - Examiner for the HDR of Baptiste Lepers, U. Grenoble
  - President of the PhD committee for Julien Mirval, U. Paris Saclay
  - President for the HDR of Pierre Sutra, IP Paris
- Pierre Sutra :
  - President of the PhD committee for Luciano Freitas de Souza, Télécom Paris

10.3 Popularization

- Mathieu Bacou: presentations at Capitole du Libre (Toulouse), to a technically experienced audience of 80. Replay on YouTube.

11 Scientific production

11.1 Major publications

[1] F. Ryabinin, A. Gotsman and P. Sutra. 'SwiftPaxos: fast geo-replicated state machines'. In: 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI '24). Santa Clara (CA), United States, 16th Apr. 2024. URL: https://hal.science/hal-04325103.

[2] S. Tanigassalame, Y. Pipereau, A. Chader, J. Toljaga and G. Thomas. 'Privagic: automatic code partitioning with explicit secure typing'. In: *MIDDLEWARE '24: Proceedings of the 25th International Middleware Conference*. 25th International Middleware Conference(Middleware 2024). Hong Kong, China: ACM, 2nd Dec. 2024, pp. 199–210. DOI: 10.1145/3652892.3700759. URL: https://inria.hal.science/hal-04895327.

11.2 Publications of the year

International journals

[3] J. Wu, Z. Cai, F. Yang, J. Li, F. Trahay, Z. Yang, C. Wang and J. Liao. 'Polling sanitization to balance I/O latency and data security of high-density SSDs'. In: *Transactions on Storage* 20.2 (6th Jan. 2024), pp. 1–23. DOI: 10.1145/3639826. URL: https://hal.science/hal-04378830.

International peer-reviewed conferences

[4] M. Bacou. 'FaaSLoad: Fine-Grained Performance and Resource Measurement for Function-As-a-Service'. In: *OPODIS 2024 : 28th International Conference on Principles of Distributed Systems*. OPODIS 2024 - 28th International Conference on Principles of Distributed Systems. Vol. 324. Leibniz International Proceedings in Informatics (LIPIcs). Lucca, Italy: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 8th Jan. 2025, 22:1–22:21. DOI: 10.4230/LIPIcs.OPODIS.2024.22. URL: https://hal.science/hal-04886267 (cit. on p. 10).

[5] J. Bolina, P. Sutra, D. Antunes and L. Camargos. 'Generic Multicast'. In: *Proceedings of the 13th Latin-American Symposium on Dependable and Secure Computing*. LADC 2024: 13th Latin-American Symposium on Dependable and Secure Computing. Recife (Brazil), Brazil: ACM, 10th Dec. 2024, pp. 81–90. DOI: 10.1145/3697090.3697095. URL: https://hal.science/hal-04909501 (cit. on p. 10).

[6] R. Dulong, N. Felber, P. Felber, G. Hopin, B. Lepers, V. Schiavoni, G. Thomas and S. Vaucher. 'P 4 ce: Consensus over RDMA at Line Speed'. In: ICDCS 2024 - IEEE 44th International Conference on Distributed Computing Systems. Jersey City, United States: IEEE, 23rd July 2024, pp. 508–519. DOI: 10.1109/ICDCS60910.2024.00054. URL: https://inria.hal.science/hal-04895326 (cit. on p. 9).

[7] L. V. Lanker, H. Taboada, E. Brunet and F. Trahay. 'Predicting GPU kernel's performance on upcoming architectures'. In: *Euro-Par 2024 : 30th International European Conference on Parallel and Distributed Computing*. The 30th International European Conference on Parallel and Distributed Computing (Euro-Par). Madrid, Spain, 26th Aug. 2024. URL: https://hal.science/hal-04614350 (cit. on p. 8).

[8] F. Ryabinin, A. Gotsman and P. Sutra. 'SwiftPaxos: fast geo-replicated state machines'. In: 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI '24). Santa Clara (CA), United States, 16th Apr. 2024. URL: https://hal.science/hal-04325103 (cit. on p. 9).

[9] S. Tanigassalame, Y. Pipereau, A. Chader, J. Toljaga and G. Thomas. 'FastSGX: A Message-Passing Based Runtime for SGX'. In: *Lecture Notes on Data Engineering and Communications Technologies*. AINA 2024 - 38th International Conference on Advanced Information Networking and Applications. Vol. LNDECT-202. Advanced Information Networking and Applications Proceedings of the 38th International Conference on Advanced Information Networking and Applications (AINA-2024), Volume 4. Kitakyushu, Japan: Springer Nature Switzerland, 9th Apr. 2024, pp. 74–85. DOI: 10.1007/978-3-031-57916-5_7. URL: https://inria.hal.science/hal-04895325 (cit. on p. 9).

[10] S. Tanigassalame, Y. Pipereau, A. Chader, J. Toljaga and G. Thomas. 'Privagic: automatic code partitioning with explicit secure typing'. In: *MIDDLEWARE '24: Proceedings of the 25th International Middleware Conference*. 25th International Middleware Conference(Middleware 2024). Hong Kong, China: ACM, 2nd Dec. 2024, pp. 199–210. DOI: 10.1145/3652892.3700759. URL: https://inria.hal.science/hal-04895327 (cit. on p. 9).

[11] F. Yang, Z. Cai, J. Li, B. Gerofi, F. Trahay, Z. Sha, M. Zhao and J. Liao. 'Adaptive selection of parity chunk update methods in RAID-enabled SSDs'. In: *MSST 2024 : Proceeding of the 38th International Conference on Massive Storage Systems and Technology*. MSST 2024 - The 38th International Conference on Massive Storage Systems and Technology. Santa Clara, United States, 3rd June 2024. URL: https://hal.science/hal-04569056.

Doctoral dissertations and habilitation theses

[12] P. Sutra. 'Contributions to the practice and theory of state-machine replication'. Institut Polytechnique Paris, 16th Jan. 2024. URL: https://theses.hal.science/tel-04588230.

[13] S. Tanigassalame. 'Privagic : confidential computing made practical with secure typing'. Institut Polytechnique de Paris, 5th Apr. 2024. URL: https://theses.hal.science/tel-04860592.

Reports & preprints

[14] A. Dutilleul, H. Pompougnac, N. Derumigny, G. Rodríguez, V. Trophime, C. Guillon and F. Rastello. *Performance debugging through microarchitectural sensitivity and causality analysis*. INRIA, 3rd Dec. 2024, pp. 1–13. DOI: 10.48550/arXiv.2412.13207. URL: https://inria.hal.science/hal-04851704.

[15] H. Pompougnac, A. Dutilleul, C. Guillon, N. Derumigny and F. Rastello. *Performance bottlenecks detection through microarchitectural sensitivity*. Institut National de Recherche en Informatique et en Automatique (INRIA), 24th Feb. 2024, pp. 1–15. URL: https://inria.hal.science/hal-04796942.