



**Offer #2025-08811**

## **PhD Position F/M Accelerating Hardware Coherence Using Programmer Input in Multi/Manycore Systems**

**Contract type :** Fixed-term contract

**Level of qualifications required :** Graduate degree or equivalent

**Fonction :** PhD Position

**Level of experience :** Recently graduated

### **About the research centre or Inria department**

The Centre Inria de l'Université de Grenoble groups together almost 600 people in 24 research teams and 9 research support departments.

Staff is present on three campuses in Grenoble, in close collaboration with other research and higher education institutions (Université Grenoble Alpes, CNRS, CEA, INRAE, ...), but also with key economic players in the area.

The Centre Inria de l'Université Grenoble Alpes is active in the fields of high-performance computing, verification and embedded systems, modeling of the environment at multiple levels, and data science and artificial intelligence. The center is a top-level scientific institute with an extensive network of international collaborations in Europe and the rest of the world.

### **Context**

The PhD is funded by the Défi Inria Cocorisco (<https://project.inria.fr/cocorisco/>), which brings several Inria teams and CEA together to build high performance platforms based on RISC-V through HW/SW interactions. As a result, this posting

will entail regular travel for project meetings (around once a year).

## Assignment

To improve performance, a general purpose processor associates a private cache memory to each of its cores, in order to keep a subset of data close to the execution units of the core and thereby accelerate data access. The chip also features a larger cache shared by all cores. This architecture is depicted in Figure \ref{fig:coh}.

To facilitate the development of parallel applications, the various cache memories implemented on chip provide data coherency. That is, a given memory address may be cached in several private caches only if the associated cores are only reading the data. Any write to the data requires invalidating all existing copies in the private caches (except for that of the writer). A cache that lost its copy following a write will have to reload the new version of the data (e.g., from memory). In general purpose processors, coherency is handled by hardware and is completely transparent to the programmer. If that were not the case, software would have to manage coherency explicitly for the program to be correct, which would significantly slow parallel application development down.

However, the hardware handling coherency is forced to make sub-optimal choices as it does not have a global vision of access and sharing patterns across the system. For instance, when data is loaded from memory, a core can ask to insert it into its private cache either with read permission or both read and write permissions. Depending on the access and sharing patterns, the correct decision (for performance or chip traffic) is not always the same. If the data is not shared, the core should obtain write permission in order to avoid sending a second request asking for write permission in the future. If the data is shared but the core is only going to read it, then, the core should ask for read permission only. Asking for write permission would imply invalidating the other copies and would be wasteful in this case. In addition, generally speaking, depending on the number of cycles between when a data is read by a core and when it is written by that core, it can be more interesting to obtain the write permission early (when reading) in order to not slow down the write if it is close in time, or late (when performing the write), in order to allow other cores to keep their copies as long as possible.

Those access and sharing patterns are known by the developer. It would therefore be interesting to express those patterns in the source code to help the hardware make correct decisions at runtime, rather than just trying to guess what that decision might be. The RISC-V instruction set being open source and extensible, it provides us with an opportunity to study this technique, by adding instructions conveying with what access and sharing patterns a data is being manipulated.

## Main activities

The thesis is built around three items:

- First, reviewing the literature on sharing patterns in multicore programs as well as hardware techniques to identify them will allow the candidate to identify patterns that we would want to express via dedicated instructions.
- Second, quantifying the frequency at which such patterns occur at runtime in typical multicore workloads (PARSEC) will be needed, in order to confirm the usefulness of such patterns and prioritize which patterns to support.
- Finally, the candidate will study the performance gain that the introduction of new instructions will bring by i) Adding support for those instructions in gcc or LLVM (through intrinsic) ii) Adding those instructions in the PARSEC benchmarks and iii) By simulating PARSEC benchmarks on a multicore processor model to which the candidate will have added support for the new RISC-V instructions, both in the processor core and the blocks handling cache coherency. For this last step, a high level simulator will be used (gem5), and hardware will not be developed using VHDL or Verilog.

## **Skills**

### **Personal Skills**

- The candidate should expect to be autonomous in developing software, experiments, and analyzing results.
- The candidate should be able to clearly express their ideas and conclusions, and to motivate their research directions.
- The candidate should be open to constructive criticism from their peers and supervisors.

### **Technical Skills**

- Programming: C/C++. Strong knowledge and understanding of data structures, testing and debugging tools.  
Linux scripting: Python, bash or other.
- At least basic level in computer architecture (caches, virtual memory, pipelining) and Instruction Set Architecture concepts. Advanced level is a plus.
- At least basic level in cache coherence concepts (coherence protocols, snooping vs. directory)
- Understanding of synchronization concepts for parallel programming (threads, locks...)
- Strong english level (B2) is required as scientific articles are written and presented in english.

## **Benefits package**

- Subsidized meals
- Partial reimbursement of public transport costs
- Leave: 7 weeks of annual leave + 10 extra days off due to RTT (statutory reduction in working hours) + possibility of exceptional leave (sick children, moving home, etc.)
- Possibility of teleworking (90 days / year) flexible organization of working hours
- Professional equipment available (videoconferencing, loan of computer equipment, etc.)
- Social, cultural and sports events and activities
- Access to vocational training

## Remuneration

2200€ gross salary per months

## General Information

- **Town/city** : Grenoble
- **Inria Center** : [Centre Inria de l'Université Grenoble Alpes](#)
- **Starting date** : 2025-09-01
- **Duration of contract** : 3 years
- **Deadline to apply** : 2025-05-17

## Contacts

- **Inria Team** : AT-EQUIPE-GRA (DIR-RAL)
- **PhD Supervisor** :  
Fuguet Tortolero Cesar / [cesar.fuguet-tortolero@inria.fr](mailto:cesar.fuguet-tortolero@inria.fr)

## About Inria

Inria is the French national research institute dedicated to digital science and technology. It employs 2,600 people. Its 200 agile project teams, generally run jointly with academic partners, include more than 3,500 scientists and engineers working to meet the challenges of digital technology, often at the interface with other disciplines. The Institute also employs numerous talents in over forty different professions. 900 research support staff contribute to the preparation and development of scientific and entrepreneurial projects that have a worldwide impact.

## The keys to success

The candidate should be aware that a PhD programme is vastly different from a BS or MS programme. Pursuing a PhD requires strong motivation and the ability to focus on a specific topic for three years.

**Warning :** you must enter your e-mail address in order to save your application to Inria. Applications must be submitted online on the Inria website. Processing of applications sent from other channels is not guaranteed.

## Instruction to apply

Applications must be submitted online via the Inria website. Processing of applications submitted via other channels is not guaranteed.

### **Defence Security :**

This position is likely to be situated in a restricted area (ZRR), as defined in Decree No. 2011-1425 relating to the protection of national scientific and technical potential (PPST). Authorisation to enter an area is granted by the director of the unit, following a favourable Ministerial decision, as defined in the decree of 3 July 2012 relating to the PPST. An unfavourable Ministerial decision in respect of a position situated in a ZRR would result in the cancellation of the appointment.

### **Recruitment Policy :**

As part of its diversity policy, all Inria positions are accessible to people with disabilities.