Performance analysis and optimisation of distributed and parallel C++ codes

Sud Touati
Internship 2018

Title: Performance analysis and optimisation of C++ codes
Advisor: Sid Touati (http://www-sop.inria.fr/members/Sid.Touati)
Grade: Professor, Université Côte d’Azur (emerged from Université Nice-Sophia Antipolis)
Laboratory: INRIA Sophia Antipolis
Teams: KAIROS (https://team.inria.fr/kairos)
Email: Sid.Touati@inria.fr
Keywords: Efficient programming, code optimisation, object oriented programming, distributed programming, parallel programming.

Context

Distributed computing applications are often programmed with a high level view of the underlying hardware platform. Usually, a distributed application encodes a parallel algorithm with distributed memory model, with an abstract view of the parallel machine. The used performance metrics of distributed applications are theoretical and high level; they have no direct relationship with the final observed execution times: high level parallel degree, CPU occupation rate, amount of communication and network transactions, scalability and speedup are all high level performance metrics. All these high level metrics are used to judge whether a distributed application or algorithm is efficient or not.

However, it is very common in practice that theoretically “good” distributed applications end with poor and disappointing measured performances. That is, the observed execution times of distributed applications are often far from the expected performance on high performance computers, especially on modern multicore processors. The reasons are known: distributed application programmers do neither write compiler friendly codes nor modern processor friendly codes. The consequences are that applications written with high-level languages are not well optimised to run in harmony with the modern processor architecture and micro-architectures. Even if CPU (core) occupancy as reported by the operating systems are high (let say 100
Detailed description of expected work

During this research internship, the student will make advanced and intensive experiments with a complete distributed and/or parallel C++ application executed on a distributed physical platform. The student will learn about the interaction between hardware and software to understand precisely the performance bottlenecks and the potential of improvement: network traffic, memory access, instruction level parallelism, thread/process scheduling and placement, memory management of operating systems, optimising compilation, etc. This work will increase the expertise of the student in the areas of distributed computing, parallelism, code optimisation, performance analysis, to understand precisely how the code interacts with the multicore processors and the operating systems.

Funding

A grant of about 500 € per month is provided for this internship (the exact amount is fixed by law). This master internship prepares for a PhD position next year.