Demande de bourse de thèse

Thesis subject title

SCHEDFAST : Understanding Linux scheduling bottlenecks

Laboratory name : INRIA Rennes
Team : WIDE
Localization : Rennes - University of Rennes 1 - INRIA - FRANCE

PhD supervisor
Name : **David Bromberg**
Position : **Professeur des universités , ½ délégation Inria Rennes en 2021**
Email : **david.bromberg@irisa.fr**
Phone number : **+ 33 2 99 84 22 05**

Co-supervisor
Name : **Djob Mvondo**
Position : **Maître de conferences**
Email : **barbe-thystere.mvondodjob@univ-rennes1.fr**

External collaborators
Name : **Julia Lawall**
Position : **Directrice de recherche**
Email : **julia.lawall@inria.fr**

Keywords. System, Large scale distributed system,Linux Kernel, Linux scheduling,

Research axis. Understanding and optimizing the Linux scheduler.

1 Thesis proposal

(1233 words)

Context. The process scheduler is the part of an operating system that decides what thread runs on what core at what time. As such, it has a critical impact on application performance, particularly for multithreaded applications. While some applications simply spread out across the available cores, with each thread running continuously without disturbance on its preferred core, others involve

frequent synchronization, I/O, etc. In practice, whenever a thread gives up access to a core, another thread can replace it, leading to migrations, loss of locality, and degraded performance. At the same time, evolutions in the scheduler can introduce errors, that can also degrade performance on specific workloads. Tools exist to trace scheduling behavior, that can help identify the presence of such problems, but the sheer volume of information available makes it difficult to map a scheduling trace to a root cause.

Objectives. The main aim of the PhD is to understand the behavior of the multicore Linux process scheduler under heavy load. We will particularly focus on the diagnosis of scheduling anomalies, as can occur in specific runs with a given scheduler implementation and as can be introduced over time by bugs in the scheduler implementation. Our starting point is the observation that we can produce an unlimited number of execution traces, including over multiple versions of the Linux kernel. Analogous to the diagnosis of illnesses using image processing, we would like to develop a model of the expected behavior of a scheduler on a given application through the use of statistical models such as for instance Markov chain models, and then to detect as anomalous execution traces that deviate from these models. The goal is to detect scheduling problems quickly, when bugs are introduced into the source code, to detect scheduling problems from short-running examples where the problem may not easily be visible to a person looking at a trace, and to connect scheduling problems reflected in a trace to specific elements of the scheduler or the application source code. Based on the results, we will consider how to improve the Linux scheduler to provide better performance.

Expected background : This PhD requires familiarity with the C language and operating systems concepts, as well as a background in machine learning.

2 References

[1] Baptiste Lepers, Redha Gouicem, Damien Carver, Jean-Pierre Lozi, Nicolas Palix, Maria-Virginia Aponte, Willy Zwaenepoel, Julien Sopena, Julia Lawall, Gilles Muller : Provable multicore schedulers with Ipanema : application to work conservation. EuroSys 2020 : 3:1-3:16

[2] Redha Gouicem, Damien Carver, Jean-Pierre Lozi, Julien Sopena, Baptiste Lepers, Willy Zwaenepoel, Nicolas Palix, Julia Lawall, Gilles Muller : Fewer Cores, More Hertz : Leveraging High-Frequency Cores in the OS Scheduler for Improved Application Performance. USENIX Annual Technical Conference 2020 : 435-448

[3] Cédric Courtaud, Julien Sopena, Gilles Muller, Daniel Gracia Pérez : Improving Prediction Accuracy of Memory Interferences for Multicore Platforms. RTSS 2019 : 246-259

[4] Justinien Bouron, Sebastien Chevalley, Baptiste Lepers, Willy Zwaenepoel, Redha Gouicem, Julia Lawall, Gilles Muller, Julien Sopena : The Battle of the Schedulers : FreeBSD ULE vs. Linux CFS. USENIX Annual Technical Conference 2018 : 85-96