Zhenyu Yan

The Battle of the Schedulers: FreeBSD ULE vs. Linux CFS

From

Justinien Bouron, Sebastien Chevalley, Baptiste Lepers, Willy Zwaenepoel EPFL
Redha Gouicem, Julia Lawall, Gilles Muller, Julien Sopena, Sorbonne University, Inria, LIP6

In this paper, arthors analyzes the impact on application performance of the design and implementation choices between ULE and CFS. There is no overall winner. On many workloads the two schedulers perform similarly, but for some workloads. There are significant and even surprising differences. First they introduces the CFS and ULE schedulers. Then describes our ULE to Linux ports and the main difference between the native ULE implementation and the port. After that, they describes the machines and workloads used in our experiments and analyzes the impact of each core schedule in CFS and ULE. At the end, they analyzes the load balancers of CFS and ULE, and talks about the relate work.

First they talks about what is ULE and CFS. ULE is the default scheduler for the FreeBSD operating system. ULE primary goal is to make better use of symmetric multiprocessing and simultaneous multithreading. CFS stand for completely fair scheduler. CFS is default scheduler for Linux. It handles CPU resource allocation for executing processes, and aims to maximize overall CPU utilization while also maximizing interactive performance. The main difference between CFS and ULE in per-core scheduling is in the handling of batch threads: CFS tries to be fair to all threads, while ULE gives priority to interactive threads. For load balancer, CFS relies on a rather complex load metric. It uses a hierarchical load balancing strategy that runs every 4ms. ULE only tries to even out the number of threads on the cores. Load balancing happens less often and ignores the hardware topology. The average performance difference between CFS and ULE is small: 2.75% in favor of ULE for load balancer, and 1.5% in favor of ULE for per-core scheduling. In the related work, they found out bugs in the Linux scheduler that could lead to cores being permanently left idle while work was waiting to be scheduled on other cores.

For result, Scheduling threads on multicore computers is difficult. In this article, they made a fair comparison of the design choices of two widely used schedulers: the ULE scheduler from FreeBSD and CFS from Linux. They show that they behave differently even on simple workloads, and that no scheduler performs better on all workloads than on other workloads.

I think this article is very useful and it is written 2 years ago which is fairly recent. I think in the future, Linux will reimplement the scheduler, or the FreeBSD and other new OS will dominate the operating system because the scheduler in Linux is buggy and it is very complex. Simplifier and efficiency is the future.

The presenter Ethan Wang and Zhenyu Yan, which is me. I think we did good. Our slide were detail and include some tables and graphs. We talks about the different aspect of the CFS and ULE scheduler. We also compare the performance by using the graph that it is in the article. I also talks about the relate work that Linux scheduler is more buggy than ULE. Overall I learn a lot when I read this article.