The Linux Scheduler: a Decade of Wasted Cores Paper Critique

Summary

This paper identifies and analyses four performance bugs in the Linux scheduler which makes the core idle while there are tasks waiting. Those bugs can cause a major degradation of performance from 13% up to 24%. Fixes to those bug are also provided in this paper. Furthermore, the authors developed two tools to check for bugs that are often overlooked by conventional testing and debugging methods: The *sanity checker* and a *scheduler visualization tool*.

Strengths

The paper bring up the problem thought to be solved in modern operating systems: Scheduling, therefore raise awareness about this problem. This proves that bugs and performance issues exist even in well-researched operating systems such as Linux. Those bugs may have cost businesses billions of dollar.

The authors also mention the problem of extensibility in the current scheduler architecture. As more and more optimizations is made, there must be a way to effectively combine them. An approach of separating *the core module* from *optimization modules* is suggested

The importance of visualization tools is also emphasized in this paper. They help finding outliers that cause bugs and performances issues can be easily overlooked.

Weaknesses and improvement suggestions

All bugs from this paper are discovered on a very specific system with 64 cores and eight-node. The fundamental cause of those bug may not lies in the scheduler architecture but in the lack of optimization on modern NUMA systems. More testing should be done to see if those bugs affect general computing systems or not.

The ideal scheduler architecture that separate the core module from optimization modules the authors suggested is very brief and is not thoroughly tested. A prototype system could be made to prove that this idea is effective.