

An Analysis of Linux Scalability to Many Cores

Vidit Jain (NetID: vjain018, SID:862394973)

Summary

This article investigates the challenges encountered when scaling applications to a large number of cores. The authors explore seven applications – Exim, Memcached, Apache, PostgreSQL, gmake, Psearchy, and Metis. Some of these applications have mechanisms to utilize multiple cores but are unable to due to several bottlenecks. Some of the identified bottlenecks were caused by locking requirements, utilizing shared memory locations, and competition for hardware resources. The authors used one 48-core machine and disabled a number of cores to obtain a detailed scalability trend versus the number of available cores. To avoid bottlenecks, the authors presented several ideas and were able to showcase significant improvements

Strengths

The authors identified specific tasks causing bottlenecks in each application and the appropriate fixes. One of the issues were caused by the current packet processing architecture, which requires each packet to pass through multiple queues before reaching application. The suggested remedy was to utilize network cards with multiple hardware queues, allowing assignment of each hardware queue to a different core. Another issue was caused by processes in multiple cores trying to access same memory, causing bottleneck by the reference counter. Using a sloppy counter, the authors demonstrated significant scalability improvements. Other similar remedial fixes were to use lock-free comparisons, per core data structures, and eliminating false sharing.

Weaknesses

The evaluation framework utilized one machine, with the desired number of cores disabled. Since the chip architecture might be heterogenous, some further discussion around the core selection methodology would have been helpful. Moreover, the evaluation could have been more comprehensive. Although the selected seven applications represented a wide variety of underlying processes, a larger scale study including more applications was lacking. The authors identified that most of the bottlenecks were caused either by hardware or the interaction with said hardware. Therefore, including a variety of CPU architectures would have made the article even more comprehensive.

Other Comments

I think the article did a good job of summarizing new and recently available ideas and their impact in improving performance scalability, I found the scope a bit lacking. However, I did find the paper extremely intuitive and was able to understand the proposed ideas. I found the sloppy counter and lock free comparison ideas very innovative, effectively solving the targeted problems without causing significant additional implementation overhead.

Overall, I really enjoyed reading the article.