Contribution ID: **205** Type: **not specified**

Load balancing via scalable task stealing

Wednesday, 14 November 2018 09:30 (30 minutes)

Summary:
In this talk I discuss scalability of load balancing algorithms in the task scheduler, and present my work on tracking overloaded CPUs with a bitmap, and using the bitmap to steal tasks when CPUs become idle.

Abstract:
The scheduler balances load across a system by pushing waking tasks to idle CPUs, and by pulling tasks from busy CPUs when a CPU becomes idle. Efficient scaling is a challenge on both the push and pull sides on large systems. For pulls, the scheduler searches all CPUs in successively larger domains until an overloaded CPU is found, and pulls a task from the busiest group. This is very expensive, so search time is limited by the average idle time, and some domains are not searched. Balance is not always achieved, and idle CPUs go unused.

I propose an alternate mechanism that is invoked after the existing search limits itself and finds nothing. I maintain a bitmap of overloaded CPUs, where a CPU sets its bit when its runnable CFS task count exceeds 1. The bitmap is sparse, with a limited number of significant bits per cacheline. This reduces cache contention when many threads concurrently set, clear, and visit elements. There is a bitmap per last-level cache. When a CPU becomes idle, it finds the first overloaded CPU in the bitmap and steals a task from it. For certain configurations and test cases, this optimization improves hackbench performance by 27%, OLTP by 9%, and tbench by 16%, with a minimal cost in search time. I present schedstat data showing the change in vital scheduler metrics before and after the optimization.

For now the new stealing is confined to the LLC to avoid NUMA effects, but it could be extended to steal across nodes in the future. It could also be extended to the realtime scheduling class. Lastly, the sparse bitmap could be used to track idle cores and idle CPUs and used to optimize balancing on the push side.

Presenter: SISTARE, Steven (Oracle)

Session Classification: Performance and Scalability MC