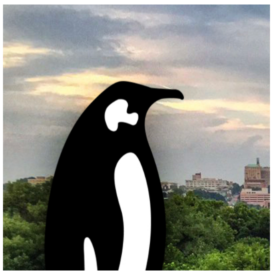

Contribution ID: **226** Type: **not specified**

newidle load balance optimization on high core count system

Before a CPU becomes idle, it kicks off idle load balance to pull tasks from other run queues to utilize the CPU and prevent it from idling. However, this search has a potential scalability problem when the number of CPUs and sched groups in the sched domain increases.

Idle load balance potentially traverses all sched domains and calculates the statistics one by one. The time cost on idle load balance could be O(n^2), where n is the number of sched groups in the domain. Overhead becomes high if there are many groups within 1 domain. For example, a system with many Cores in a LLC domain. And there are CPUs on the horizon that this is the case. This is not good for latency, performance, or power. We'll share several proposals on statistic calculation optimization and the test results to mitigate this issue. We hope to get advice/feedback on tuning these proposals and making them viable for upstreaming.

Primary author: YU, Chen

Presenter: YU, Chen

Session Classification: Real-time and Scheduling MC

Track Classification: LPC Microconference: Real-time and Scheduling MC