Contribution ID: **384** Type: **not specified**

Push based load-balancing for fair tasks

During the discussion at OSPM '25, the idea of using push-based load balancing as an alternate to idle and newidle balance was proposed. A prototype [1] was sent soon after OSPM to gather feedback from the community.

During the review, Peter mentioned optimizing the global nohz idle tracking to be reduced to per-LLC tracking to reduce the cost of access and update to this shared data being used for load balancing [2]. With the infrastructure to track the idle CPUs more efficiently in place [3], there are more fundamental challenges that require further discussion:

1. Reducing the overhead of pushing the task: Push is always done from a busy CPU's context, adding latency to runnable task's execution. Is there a better mechanism to address this by offloading the push to a smp_call_function on the idle target?
2. Avoid selecting the same CPU for push: Multiple busy CPUs can observe the same CPU to be a potential idle target for push and can overload this single CPU. Is there an inexpensive indicator to ensure task pileup is avoided?
3. Beyond nohz idle tracking: CPUs can idle without disabling the periodic tick. Is there a potential to extend nohz idle tracking to all idle CPUs efficiently to truly replace newidle balance?

A larger prototype addressing some of the issues listed above will be posted close to the conference proceedings. We also seek to reach an agreement on the infrastructure for pushing fair task in order to align the efforts to enable it for both capacity aware scheduling (CAS) and energy aware scheduling (EAS) use cases.

[1] https://lore.kernel.org/lkml/20250409111539.23791-1-kprateek.nayak@amd.com/
[2] https://lore.kernel.org/lkml/20250410102945.GD30687@noisy.programming.kicks-ass.net/
[3] https://lore.kernel.org/lkml/20250904041516.3046-1-kprateek.nayak@amd.com/

Primary author: NAYAK, Prateek (AMD Inc.)

Presenter: NAYAK, Prateek (AMD Inc.)

Session Classification: Scheduler and Real-Time MC

Track Classification: Scheduler and Real-Time MC