Contribution ID: **86** Type: **not specified**

Paravirt Scheduling: Framework for better physical CPU utilization

In para-virtualized environment, vCPU overcommit is a common configuration which helps customer to make better use of CPU resources since not all VMs would be active at the same time and hence underlying hypervisor will be able to meet the CPU demand and workloads running on VMs can benefit from the extra resource.

Acronyms:
vCPU - virtual CPU - CPU in VM
pCPU - physical CPU - CPU governed by Hypervisor.

But when all or most of the VM request for CPU resource at the same time, hypervisor wont be able to meet those requirements and has to context switch among them to meet the CPU demand and be fair. This context switch is called as vCPU preemption. This vCPU preemption is much more expensive than the task preemption within the VM. Workload performance degrades significantly as the result.

In such a situation, if VMs and hypervisor can co-operate among themselves and use lesser number of vCPUs it improves overall performance. Such vCPU which shouldn't be used at this point are called as paravirt CPU. Provide a framework in Linux kernel to identify these paravirt vCPUs and consolidate workload on non paravirt vCPUs.

This is achieved by:
1. Not scheduling any new task on paravirt CPUs.
2. Make load-balancer aware of paravirt CPUs.
3. Push the workload away from paravirt CPUs if it is running on them.

Which vCPUs to mark as paravirt is left to the architecture. Experimentation was done with help of hint from debugfs file.There is effort ongoing to plug in steal time values to decide the hint. It would ideal if the hint is provided by the hypervisor.

Discussion Points:
1. Comparison of different methods (capacity, load balancer, offline)
2. Plugging in steal time to set paravirt CPUs
3. Subtle issues related to implementation.
4. Do all scheduler classes need it to honor it.
5. How userspace such as irqbalance/SCHED_EXT can exploit it.

This is a follow up of discussion in Plumbers 2024.
http://www.youtube.com/watch?v=vMgTAdYAMeQ
http://www.youtube.com/watch?v=pZaO5TlzEjo

Patches: (Latest being first)
https://lore.kernel.org/all/20250625191108.1646208-1-sshegde@linux.ibm.com/
https://lore.kernel.org/all/20250217113252.21796-1-huschle@linux.ibm.com/

Primary author: HEGDE, Shrikanth

Presenters: LEOSHKEVICH, Ilya; HEGDE, Shrikanth

Session Classification: Scheduler and Real-Time MC

Track Classification: Scheduler and Real-Time MC