



Contribution ID: 305

Type: **not specified**

## A case for using para-virtualized scheduling information with sched\_ext schedulers

*Wednesday, 18 September 2024 12:00 (20 minutes)*

Applications running inside a virtual machine experience dual level of task scheduling i.e. the Guest scheduler decides how to place application threads on the vCPUs, and the Host scheduler decides how to place the vCPU threads on the pCPUs. Neither the Guest nor the Host scheduler has the complete information to make optimal task placement decisions across both the levels. This leads to various known issues like the Lock Holder Preemption problem, the Blocked Waiter Wake-up problem, the RCU Reader Preemption problem, the problem of the Guest scheduler being oblivious to the runtime changes to vNUMA on the Host, the problem of delayed IO in the Guest, etc.

The root cause of these problems is the semantic gap between the decisions made by the Host and the Guest schedulers. Many academic as well as in-kernel solutions demonstrate that the semantic gap can be addressed efficiently by para-virtualizing the scheduling related information between the Host and the Guest. This talk discusses about the technical challenges encountered in the endeavour of aggregating and using para-virtualized scheduling information with the sched\_ext schedulers.

**Primary author:** CHHAYA-SHAILESH, Himadri (Inria)

**Presenter:** CHHAYA-SHAILESH, Himadri (Inria)

**Session Classification:** Sched-Ext: The BPF extensible scheduler class MC

**Track Classification:** Sched-Ext: The BPF extensible scheduler class MC