Hypervisor Extension for a RISC-V Processor

J. Gauchola¹, J. Costa¹, E. Moranchő¹, R. Canal¹, X. Carril², M. Doblas², B. Otero¹, M. A. Pajuelo¹, E. Rodríguez¹, J. Salamero² and J. Verdú¹

¹Universitat Politècnica de Catalunya – Barcelona TECH, ²Barcelona Supercomputing Center

jaume.gauchola@estudiantat.upc.edu

Abstract
Data centers and cloud environments require efficient virtualization. We present the modifications done on the Lagarto core to add an initial implementation of the RISC-V Hypervisor extension:

- Add new privilege levels (VU, VS, HS).
- Modify trap handling mechanism.
- Add two-stage virtual memory address translation.

New Privilege Levels
The hypervisor extension adds several control and status registers to control and manage the new modes, among other features. These new execution modes are Hypervisor, Virtual Supervisor, and Virtual User.

Memory Address Translation
Address translation in RISC-V relies on memory paging. Sv39 supports three page sizes (4 KiB, 2 MiB and 1 GiB). All address translations require traversing a 3-level tree of page tables structure. The Page Table Walker (PTW) is responsible for this traversal and stores the result in the Translation Look-aside Buffer (TLB) to enable faster translations in the future.

Trap Handling
When a trap is triggered, it is handled in machine mode by default. However, it can be delegated to a less privileged mode. The new execution modes require the trap delegation mechanism to be modified and new exceptions to be added from virtualized environments.

Bibliography

Conclusion
Multiple modules such as the memory management unit, privilege levels, and trap handling are modified to implement a first approach of hypervisor extension in a RISC-V processor. The knowledge gained is used to implement hypervisor extensions for other projects like Vitamin-V.

This work is partially supported by the DRAC (IU16-011591), the HORIZON Vitamin-V (101093062) and the Computación de Altas Prestaciones VIII (PID2019-107255GB) projects.

Guest apps
Host OS
Virtual user
Virtual supervisor
Virtualized environments
Non-virtualized environments

OS applications
Guest OS
Guest OS

Physical machine

44 bits
9 bits
9 bits
12 bits

Data
Data
Data
Data

Virtual user
Guest applications

Virtual supervisor
Guest OS

Hypervisor
Host OS

Machine Kernel

User Applications

Supervisor
Host OS

Least privileged

Most privileged

Trap Handling

Possible delegation

M-mode
HS-mode
VS-mode
VU-mode

M-mode
Raised
HS-mode
Raised
VS-mode

Sv39 support

Page Table Walker

Translation Look-aside Buffer

Virtual user
Guest applications

Virtual supervisor
Guest OS

Hypervisor
Host OS

Machine Kernel

Non-virtualized environments

Virtualized environments

Virtual user
Guest applications

Virtual supervisor
Guest OS

Hypervisor
Host OS

Machine Kernel

User Applications

Supervisor
Host OS

Least privileged

Most privileged

Page table

Virtual user
Guest applications

Virtual supervisor
Guest OS

Hypervisor
Host OS

Machine Kernel

Non-virtualized environments

Virtualized environments

When virtualization is enabled, a second translation stage (G-stage) is required to translate all virtualized physical addresses (from VS-stage) from the virtualized environment to real physical addresses in the host.

When a trap is triggered, it is handled in machine mode by default. However, it can be delegated to a less privileged mode. The new execution modes require the trap delegation mechanism to be modified and new exceptions to be added from virtualized environments.