KVM/ARM: The Design and Implementation Of the Linux ARM Hypervisor

Dineshkumar RAJAGOPAL (M2R-MoSIG-PDES)

05 Jan 2015

KVM/ARM Hypervisor:

Hypervisor is dual mode hosted virtual machine and supports multiprocessor, use Hardware virtualization features in ARM v7 cortex A15 processor and Linux kernel features. KVM-ARM is added in Linux kernel successfully to support full virtualization (without any modification in guest and host OS).

Summary:

Hardware virtualization support is available in new CPUs to support native virtual machine (bare metal hypervisor) and hosted virtual machine (hypervisor on the host OS). ARM new CPU architectures ARMv7 and ARMv8 supports H/W virtualization is completely differs from X86 virtualization extension in Intel, so hypervisors developed for X86 can not be ported to ARM platform easily. Xen is a bare metal hypervisor for server platform. ARM has different platform, so supporting all the ARM platforms makes bloated hypervisor and maintenance is difficult, so Xen is suffering continuous platform support.

Linux successfully supporting all the ARM platforms, so leveraging the Linux capability in the hypervisor was an approach in earlier KVM-ARM hypervisor but it is lacked architecture design made the para-virtualization support and under utilized the hardware virtualization capability.

Hardware virtualization support will reduce the software virtualization overhead and improve the performance of a VM like a native machine. In ARMv7 virtualization support are targeted for bare metal hypervisor. In KVM-ARM use the existing virtualization support in ARMv7 to integrate KVM in Linux kernel to support Virtual machine in dual mode to improve performance and use host kernel features to reduce code complexity in KVM-ARM hypervisor.

ARMv7 CPU architecture provides virtualization supports to all the components CPU, Memory, Timer, Interrupt and I/O. In KVM-ARM all the virtualization support has to use effectively. For achieving virtualization, ISA should be emulated effectively especially sensitive instructions. In the KVM-ARM split mode virtualization is used to leverage new privilege mode (Hyp mode) in ARM processors. Hyp mode only supports trap and emulate and the high privilege mode than all other (user, kernel). Split mode architecture splits hypervisor into two parts lowvisor (Hyp mode) and highvisor (kernel mode). Lowvisor uses Hardware virtualization support and Highvisor uses operating system feature to make an efficient hypervisor.

For guest and host physical memory management stage-2 and stage-1 page table translation was used with the hardware support. For interrupt Virtual Generic Inerrupt Controller (VGIC), For timer VirtualTimer (vTimer) was used for each core in SMP.

X86 CPUs supports less virtualization than AMD, so s/w virtualization overhead and code complexity line of code is less for KVM-ARM than KVM-X86. Performance of the KVM-ARM with VGIC and vTimer support is two times better than without VGIC and vTimer. KVM-ARM performance is almost equal to KVM-X86 and has better performance in some application in VM.

Pros:

- Sensitive instructions execute in VM user mode will be handled by guest OS itself (VM kernel mode)
- Virtual machine mostly run as a host machine with the same modes and mapping in the general purpose register
- Isolation is done by Lowvisor to isolate the access from Host and Guest machines effectively to achieve the security
- KVM-ARM Trap cost is only few clock cycles than KVM-X86
- Implementation complexity is very less, because of using operating system functionalities
- More privileged portion of the Lowvisor is having mechanism and less dense, so the virtualization is very safe

Cons:

- A trap from the VM to hypervisor involves multiple mode transition, so it takes double trap cost (From VM to Lowvisor and Lowvisor to Highvisor).
- World switch is costly because of saving VGIC and vTimer state information
- Hyper call takes high number of clock cycles because of world switch

Detailed discussion:

KVM-ARM split mode virtualization is a new design for virtualization to run the hypervisor in different modes to leverage the functionalities in the different modes. This is the first full virtualization KVM for ARM used the features of Hardware virtualization. In the paper performance measurement methodology and explanation are very clear and they mentioned they measured the hypervisor performance not the ARMv7 cortex 15 processor or Intel core i7. Hypervisors used the virtualization supports in CPUs and measured the performance, If the old processor does not have the virtualization support then the new KVM-ARM will compatible with the earlier CPUs, as well How the performance vary is not covered in the paper.

If the KVM-ARM will support recursive virtualization or not. The CPU virtualization for VM is same as an ARMv7 cortex 15 processor and How they will virtualize Hyp mode for virtual CPU.

In this hardware virtualization support is important for Hypervisor flexibility and performance. In X86 provides store and restore on the world switch made good performance than the AMD lack of the support. From this hardware and hypervisor developers to work together to get a good virtualization environment. For that reason they gave some suggestion for the new CPUs hardware virtualization support.