Operational Semantics and Proof Rules for VMs in Halfnium

Zongyuan Liu

November 13, 2020

1 Operational Semantics

1.1 Settings of Halfnium

Halfnium is a type-1 hypervisor running on the AArch64 artitecture(for now), who is claimed can provide memory isolation between a set of virtual machines(VMs) running on it. Halfnium provides memory isolation by managing the stage-2 page tables of VMs. It controls the stage-2 translation by manipulating entries in stage-2 page tables only according to the FF-A calls issued by VMs. The hardware does the actual address translation by looking up page table entries, which we assume acting correctly as described. Therefore Halnium can guarantee that one VM can never access to the memory owned by another VM without that VM's consent.

For the sake of simplicity, stage-2 page tables managed by the Hafnium are simply one-to-one mappings, i.e. the IPAs are always equal to PAs. Besides, it is also the case for the stage-1 page table of the Hafnium itself.

FF-A framework are designed as a protocol for components running on the ARM-A processor, the main purpose of which is to allow components in the normal world to communicate with components in the secure world. While in Hafnium, VMs and the hypervisor are all counted as components residing in the normal world. The FF-A framework is then used as an approach for VMs to do message passing and memory sharing with each other. We will see some examples of how exactly the FF-A framework is used in Hafnium later.

One can read more details on the Hafnium artitacture at here.

1.2 System state

In our operational semantics, we abstract away the Halfnium hypersior, considering the whole system only consists of several virtual machines plus some additional information for the FF-A framwork.

1.3 Syntax and Instructions

1.4 Reduction rules

2 Proof Rules