Event-B Course

13.1 Hypervisor Formal Modeling with Event-B

(memory)

Jean-Raymond Abrial and Rustan Leino

September-October-November 2011

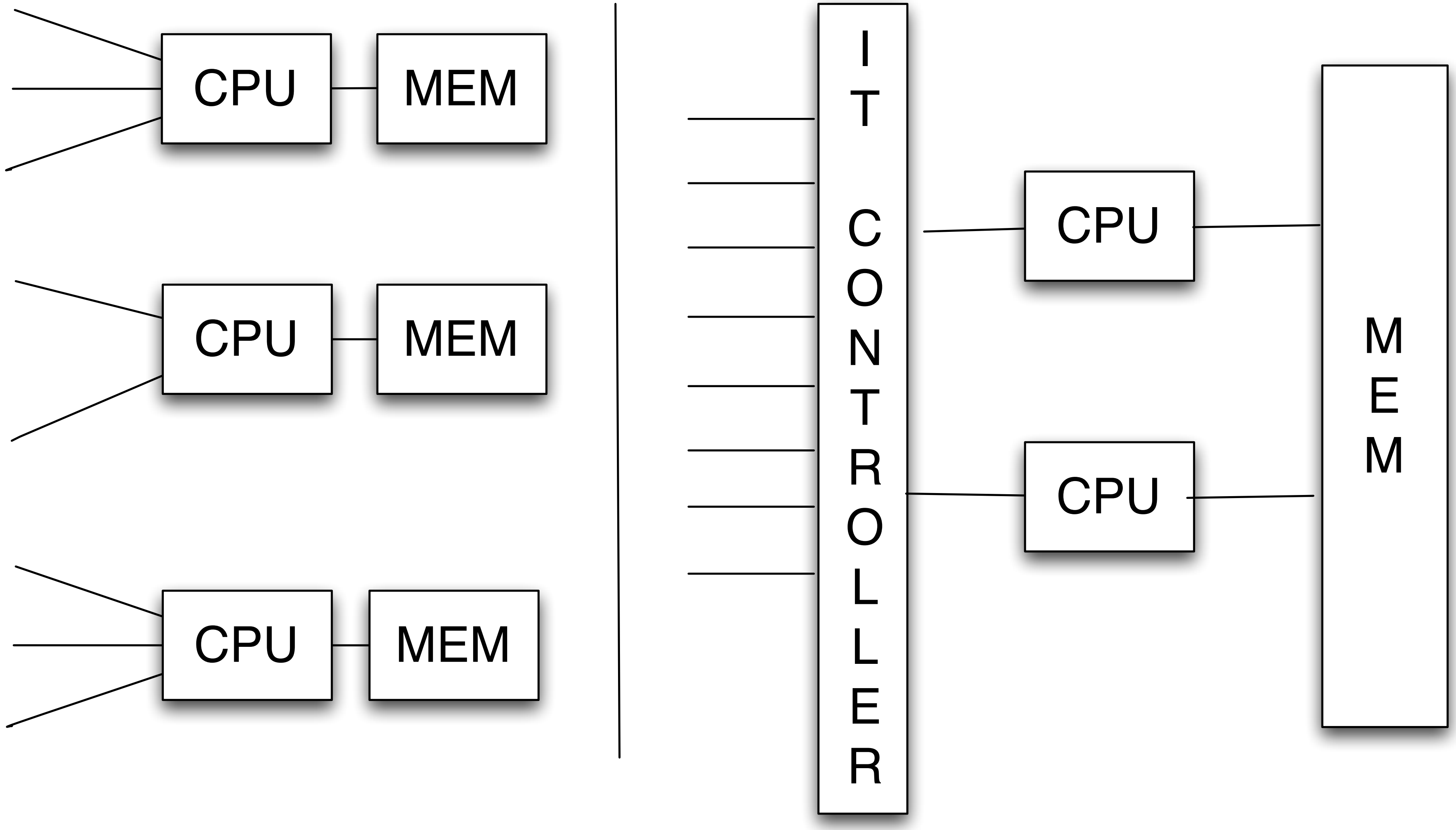
Goal of the Hypervisor Developement

- We want to transform a number of physical OSs
  (interrupts, CPU, memory)

- into some virtual OSs running on a multicore machine
  (interrupts, CPUs, memory)

- The various virtual OSs should not be aware of this

Diagram



Outline

- Writing the requirement document

- Defining the refinement strategy

- Making the formal model developments

- Conclusion

Reference Text Labeling

- Single system memory handling: SM

- Hypervisor: HV

- Hypervisor memory handling: HM

- Hypervisor scheduling: HS

Single System Memory Handling (1)

- SM-0: An operating system (OS) makes use of some number of cores (CPUs), uses some memory, and communicates with some number of devices and timers.

- SM-1: The amount of core memory accessible to an OS is determined at boot time, and is addressed as pages from 0 onwards. The addresses of these pages are called intermediate physical addresses (IPAs).

- SM-2: An OS also uses some virtual addresses

Single System Memory Handling (2)

- SM-3: An OS has access to a number of per-core hardware registers, including the Translation Look-aside Buffer (MMU-TLB).

- SM-4: The MMU-TLB is an associative memory. It contains pairs of the following form: "Virtual Address - IPA"

- SM-5: If the OS tries to access its memory through a non-existing IPA, then it crashes.

Producing an IPA from a Virtual Adress

(0) Running in kernel mode, the OS can issue instructions that address an IPA directly.

(1) Running in user mode, the OS can produce an IPA from a virtual address as follows:

- using the MMU-TLB if the virtual address in the MMU-TLB.
- using the page table (if not in MMU-TLB but in page table)
- using the disk (if not in page table)

- In the last two cases, the MMU-TLB is updated

- In the last case the page table is dpdated

Hypervisor (1)

- HV-0: The role of the hypervisor is to simulate on a single machine (with several cores) the behavior of independent operating systems, here known as guest OSs.

- HV-1: The number of guests is fixed and is determined at boot time.

- HV-2: A guest OS is not aware that it is being executed under the hypervisor.

Hypervisor (2)

- HV-3: Guests are really independent: a guest cannot inspect or influence the behavior of other guests, not even know of their presence.

- HV-4: It is the responsibility of the hypervisor to schedule the guests.

- HV-5: The hypervisor itself should not take the control of its hardware permanently. That is, it should do some scheduling of guests.

Hypervisor Handling of Memory (1)

- HM-0: The hypervisor controls a physical core memory, into which it embeds the core memory of its guests.

- HM-1: The hypervisor is not concerned by the virtual adresses of the guests, only the IPA of the guests

- HM-2: The hypervisor has a data structure called the SLAT (second-level address translation), which keeps track of, for each guest, a one-to-one mapping between the guest's IPAs and the physical addresses (PAs).

- HM-3: The size and contents of the SLAT associated with each guest is determined at boot time.

Hypervisor Handling of Memory (2)

- HM-4: The SLAT of each guest is made available through a base address (BA).

- HM-5: When a guest provides an IPA, the hardware attempts to map that IPA to a PA by consulting the SLAT associated with this guest.

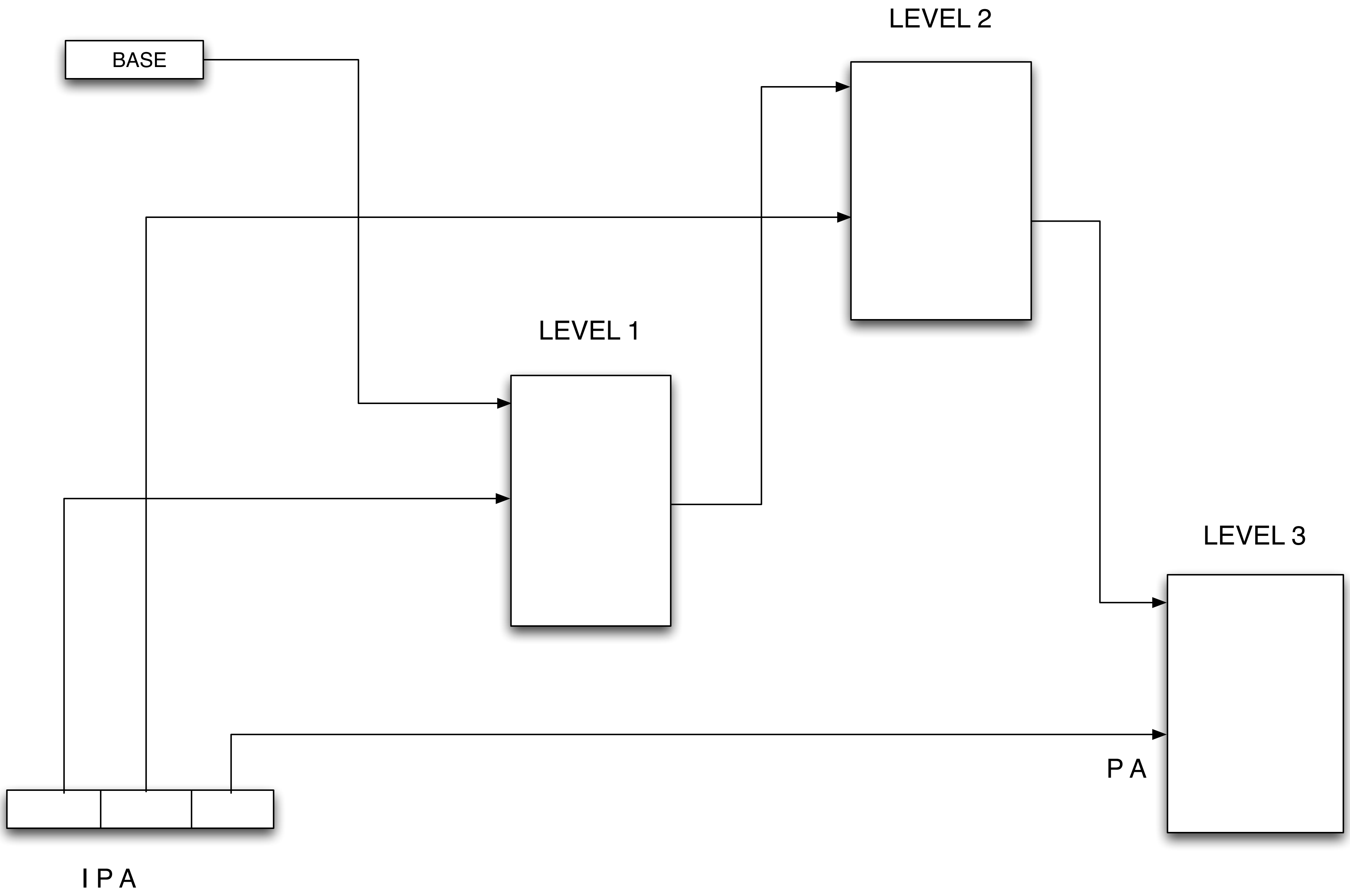
- HM-6: If the previous process fails (that is, if the SLAT does not contain an entry for that IPA for the requesting guest), then a second-level page fault occurs.

Hypervisor Handling of Memory (3)

- HM-7: The hypervisor traps second-level page faults. Upon such a page fault, the hypervisor will refuse the IPA request and will report this failure back to the guest (which may result in a "blue screen" on the guest). This corresponds to what has been described in requirement SM-5 for a single OS.

- HM-8: The SLAT of each guest is structured as a tree of pages with a root and two levels.

The Three Levels of the SLAT



Hypervisor Handling of Memory (4)

- HM-9: An IPA is a 32 bits word made of three parts: the level 1 part is made of the 10 upper bits of the IPA, the level 2 part is made of 10 intermediate bits, and the level 3 part is made of the 12 lower bits.

- HM-10: The level 1 bits of an IPA address the root page of the SLAT. This root page is itself pointed to by the base address of the SLAT (see HM-3). The root page is made of 1024 words. The contents of the word of this root page point to 1024 words level 2 pages.

Hypervisor Handling of Memory (5)

- HM-11: A level 2 page is a word page of size 1024. The level 2 bits of an IPA address a word in a level 3 page. The contents of the word of this level 2 page point to 1024 words level 3 pages.

- HM-12: A level 3 page is a byte page of size 4096. The level 3 bits of an IPA address a byte in a level 3 page. A level 3 page of a SLAT is part of the memory of the guest associated with that SLAT.

Hypervisor Handling of Memory (6)

- HM-13: The hypervisor is provided with a SLAT-TLB mapping some of the guest IPAs to the corresponding PAs. The SLAT-TLB is a short-circuit avoiding to always walk through the three level pages of the SLAT.

- HM-14: Once the hypervisor has walked through a SLAT in order to map an IPA to a PA, this pair is entered into the SLAT-TLB to be reused directly if another usage of that pair is needed. In doing so, a pair of the SLAT-TLB is evicted in order to make room for the new one.

Hypervisor Scheduling (1)

- HS-0: The hypervisor runs on a system providing several cores.

- HS-1: Each core is provided with a MMU-TLB and a SLAT-TLB.

- HS-2: Each core has a physical base register containing the base address of a SLAT (see HM-3).

- HS-3: The number of guests controlled by the scheduler might be greater than the number of cores.

Hypervisor Scheduling (2)

- HS-4: A guest that is not assigned to a core is said to be a sleeping.

- HS-5: The hypervisor regularly schedules a sleeping guest to a core. The guest currently running on the core is made sleeping. When scheduling, the MMU-TLB and the SLAT-TLB of that core are flushed.

- HS-6: When a guest is scheduled to a core, the base register of that core is updated with the base address of the SLAT of the scheduled guest (see HM-3).