

Xen-ARM Update: status and latency analysis

Sang-bum Suh, Jae Min Ryu
{ sbuk.suh, jm77.ryu }@samsung.com
Future IT Research Center
SAIT, Samsung Electronics
April 28~29, 2010

Xen Summit North America 2010

- Xen-ARM Support
- New Features
- Latency Analysis of Xen-ARM
 - Instrumentation for Profiling
 - Interrupt Latency Analysis
 - System Call Latency and Hypercall Overhead Analysis
 - Context Switching Latency Analysis
- Discussions

H/W Platform

H/W Platform

- ARM926EJ-S (i.MX21, OMAP5912)
- Xscale 3rd Generation Architecture (PXA310, Core Only)
- ARM1136/ARM1176(Core Only)
- Goldfish (QEMU Emulator)
- Versatile Platform Board
- ARM11MPCore (Realview PB11MP)
- Cortex-A9 (Core Only)

Para-virtualized O/S

Para-virtualized O/S

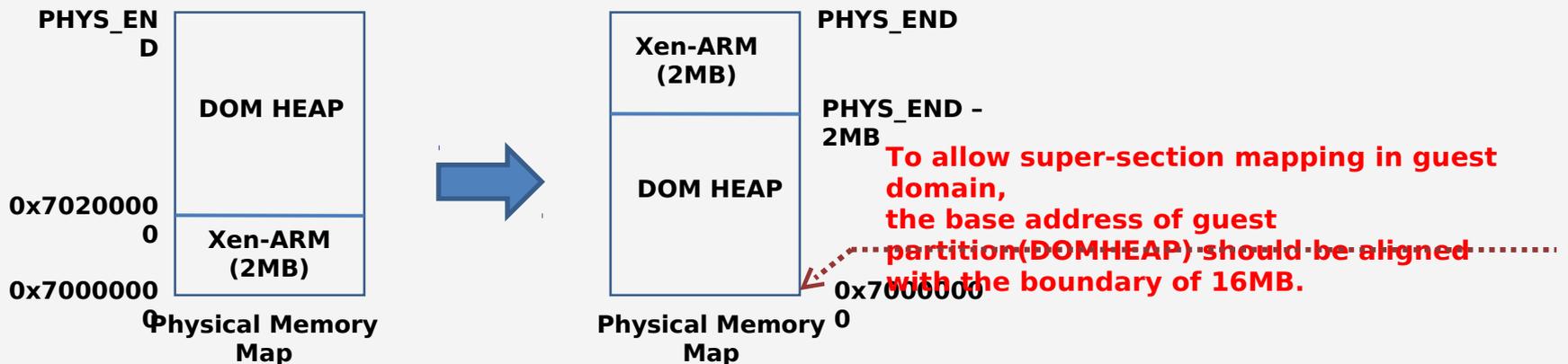
- Linux-2.6.21
 - Multi-core support patch applied
- Linux-2.6.24
 - Multi-core support patch applied
- uC/OS-II

Multi-core Support

- Synchronization
 - Locking Primitives
 - spinlock, read-write lock
 - Inter-processor Communication
- Core Topology Management
- VCPU Migration Enabled

Super-section Mapping(16MB) Support

- From ARM v6 core, it provides multiple-sized mapping unit.
 - Page (4KB), Section (1MB), Super-section(16MB)
- To support this mapping unit, the physical memory map of Xen ARM (v5) is changed.

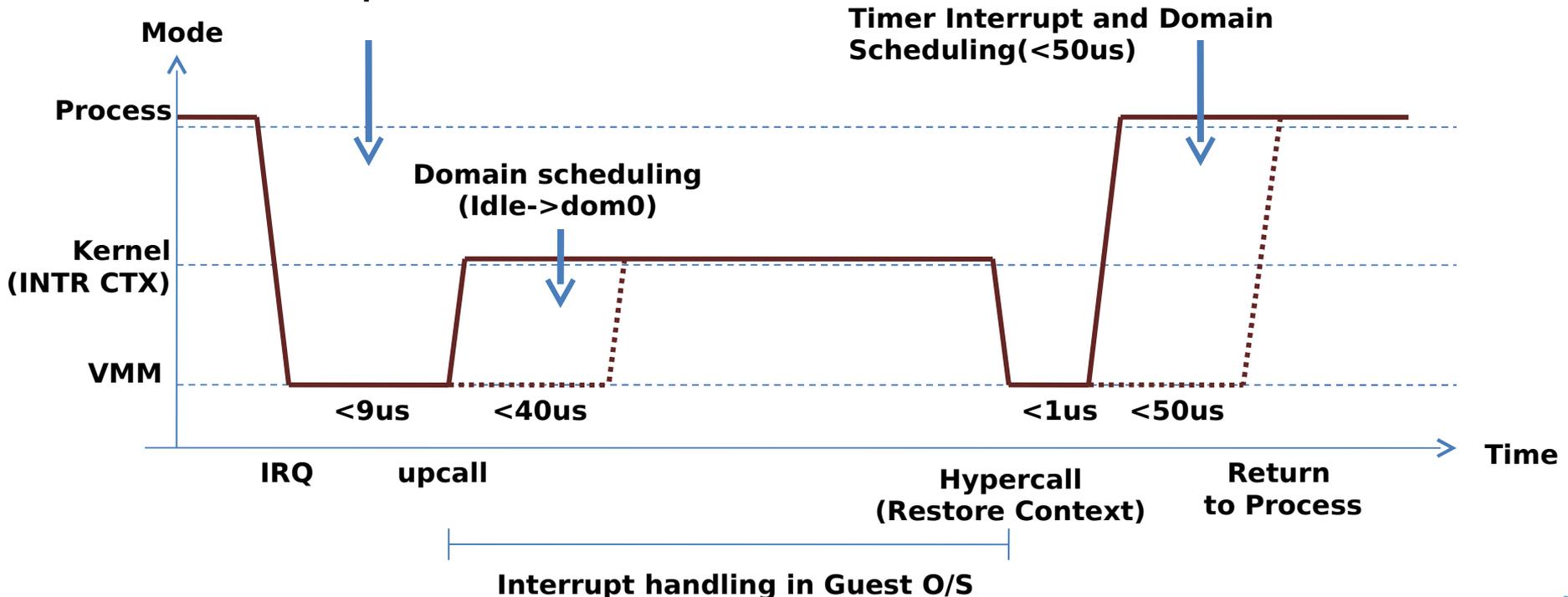


- **Target Hardware**
 - **CPU : Xscale 3rd Generation 624MHz**
 - **L1 Cache :**
 - **Instruction Cache : 32KB**
 - **Data Cache : 32KB**
 - **L2 Cache : 256KB (Disabled)**
 - **Memory : 128MB**
- **Xen-ARM**
 - **Scheduler : SEDF**
- **Guest Operating System**
 - **Linux-2.6.21**

Interrupt Latency

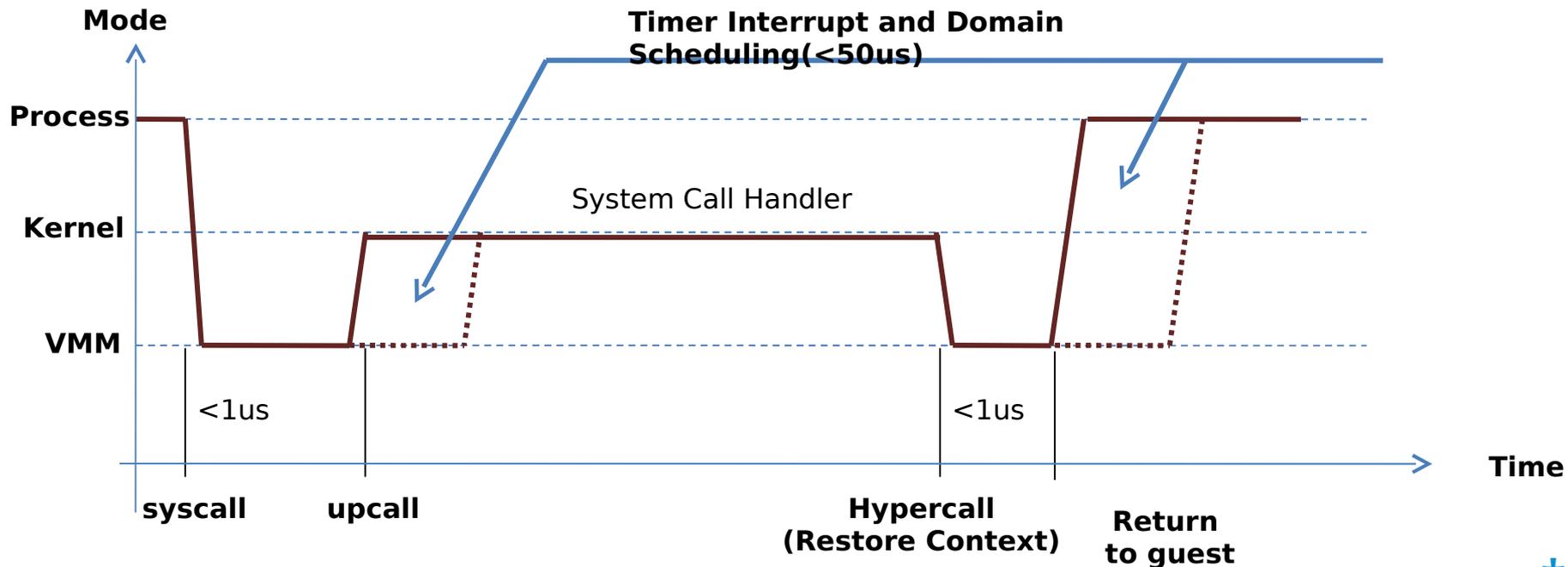
- From a guest domain point of view, interrupt latency consists of
 - CPU cycles to virtualize interrupt
 - CPU cycles to schedule domain
- In case of single domain, the worst case interrupt latency is less than around 50us.
 - However, in case of two domains, interrupt delivery to an inactive guest domain may be delayed further until the inactive domain is scheduled.

Guest Domain's Context Manipulation Timer Interrupt Virtualization



System Call Latency and Hypercall Overhead

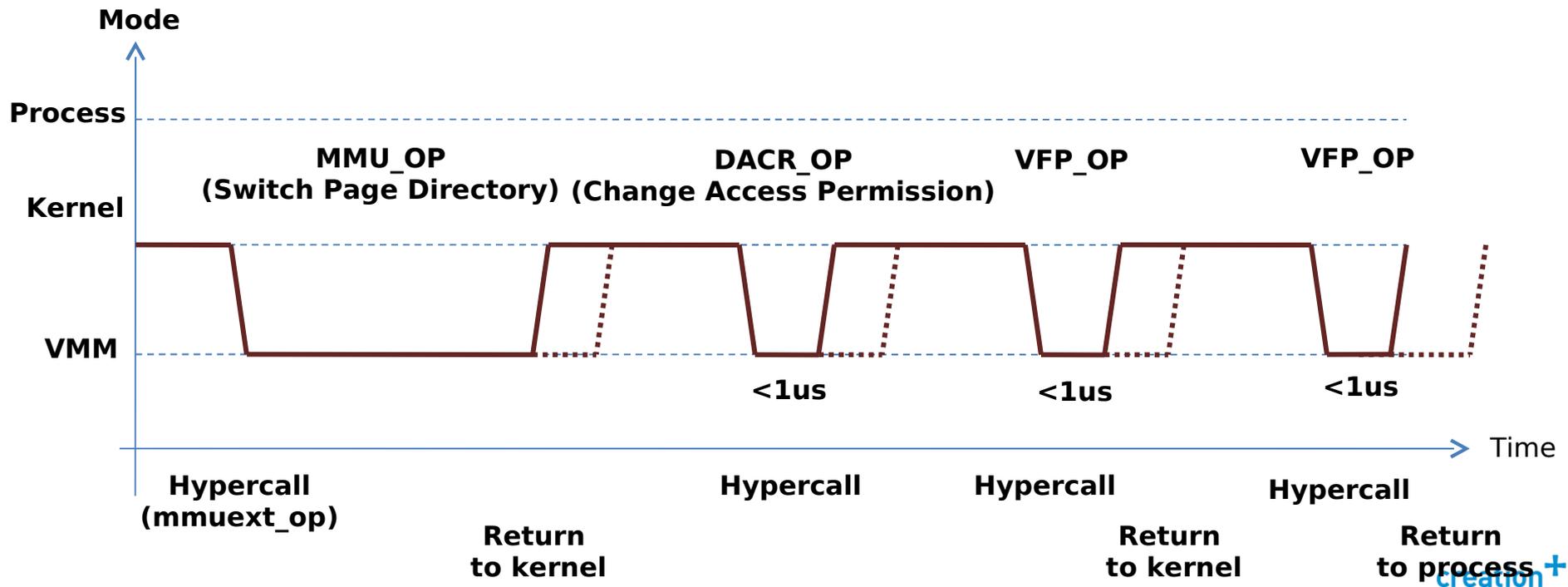
- As both system call and hypercall use the same SWI instruction, Xen-ARM should decode the SWI instruction issued from unprivileged mode to identify system call and hypercall.
 - > system call latency has increased by 1 us,
guest domain's context manipulation time + SWI instruction decode time



Context Switching Overhead

- During task switching in a guest domain, it is possible for Xen-ARM to receive any physical interrupts (even though virtualized interrupt to a guest domain is disabled).

-> context switching overhead would be increased.



- Though interrupt latency by Xen ARM does not seem large,
 - Event delivery to an inactive guest domain may be delayed until the domain is scheduled.
- To guarantee predictable responsiveness required for real-time CE systems,
 - We will change Xen ARM to be preemptible.

Xen on ARM Project in Xen Community:

<http://wiki.xensource.com/xenwiki/XenARM>
<http://www.xen.org/>

THANK YOU !>.<!