Keynote Speech:
Xen ARM Virtualization

VP Sang-bum Suh, Ph.D.
sbuk.suh@samsung.com
S/W Platform Team
DMC Research Center
SAMSUNG Electronics

2 November 2011, Seoul Korea

Xen Summit Asia 2011

Contents

- SEC Overview
- DMC R&D Center Overview
- Xen ARM Virtualization

SEC Overview

Corporate Philosophy

We will devote our people and technology
to create superior products and services
thereby contributing to a better global society.

History

1969
- Established the company

1972
- Started manufacturing B&W TV

1992
- Ranked #1 in DRAM
- Developed the cellular telephone system

2002
- Became market leader in flash memory
- Achieved leading share of LCD panel market

2004
- Introduced mobile WiMAX technology (World's 1st)

2006
- Ranked #1 in TV market

2007
- Ranked #2 in global handset market

2010
- No.1 revenue in global electronics industry ($134B)

Global Top Tier

Business Divisions

Visual Display

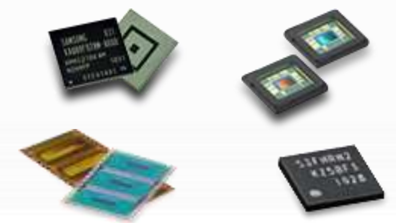
Mobile Comm.

Memory

IT Solutions

Telecomm.Systems / HME

System LSI



Digital Appliances

Digital Imaging

LCD

Recent Technology Leadership

Pioneering new technologies

2005	2006	2007	2008	2009	2010

World's largest TV
Sep 2005

World's first HSDPA phone
May 2006

World's first Blu-ray player
Jun 2006

World's first 30nm 64GB NAND
2007

World's first HSUPA phone
Apr 2008

World's slimmest LED TV
Jan 2009

World's first 30nm 2GB DDR DRAM
Jan 2010

DMC R&D Center Overview

Core R&D Domain (1/3)

1. NG Comm. & Networking

Conduct research for NG communication systems & connectivity solutions in advance

- NG mobile comm. system
- Wired/Wireless connectivity
- NG broadcast & service technologies

2. Advanced Media Processing

Create NG multimedia devices using innovative technologies

- NG display & audio solution (UHD, 3D, Amp, Speaker)
- NG video/audio codec
- Realistic graphics
- Medical imaging

Core R&D Domain (2/3)

3. Convergence & Platform Solutions

Build a new kind of ecosystem for multi-device convergence & improve platform competitiveness

- Multi-device convergence (AllShare[1]), Smart Home)
- Mobile S/W platform (SLP)
- Cloud service platform

4. Intelligent/Emotional Interaction

Create customized intelligent/emotional UX

- UI identity for SEC's device
- Multimodal interaction (Flexible & Ambient interface)
- NG UX (Context awareness)

1) AllShare : Integrated Service Solution of SEC (IT/Smart CE/Non-IT Devices)

Core R&D Domain (3/3)

5. Differentiated Device Solutions

Differentiate mobile device through innovative module solution & sensor application

- Camera SoC (DSC/CAM common)
- Mobile camera module
- Sensor application
- New function module (EMR[1)] pen)

1) EMR: Electro Magnetic Resonance

6. Eco-friendly Solutions

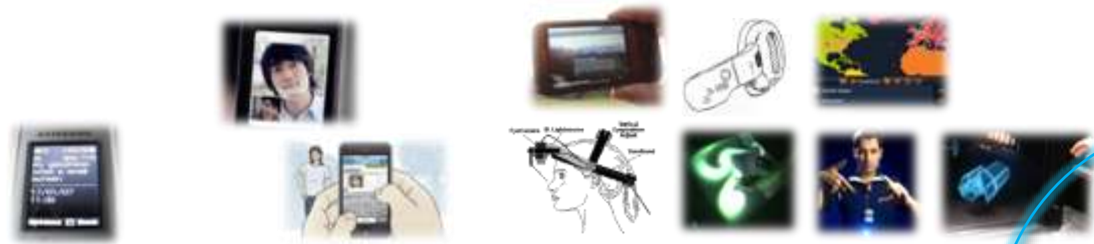
Develop eco-friendly core technologies & create new business opportunities

- Energy management (HEMS, BEMS)
- Energy saving (printer, air conditioner)
- Life-care solution
  (Water/Air care, u-Health, etc.)
- Clean material

Xen ARM Virtualization

Future Computing Trends

Changes in Computing



Closed
Centralized
Correct Info.
Stationary

▪Keyboard/Mouse
▪Voice Call, SMS

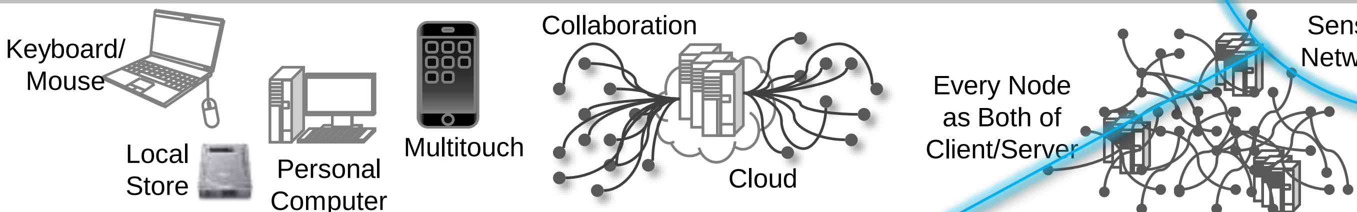
▪Multitouch
▪Video Call, MMS

▪Centeralized/Concentrated
▪Known Comm. Entities

▪Augmented Reality ▪Gesture ▪Interactive 3D UI
▪Eye-Tracking ▪Manytouch ▪Realtime Web

▪Distributed/Scattered
▪Unknown/Utrusted Comm. Entities

Open
Distributed
Correct+Timely Info.
Mobile



Embedded	Single-core	Multi-core	Many-core
IT	Single-core	Multi-core	Many-core

▪UC Berkeley
Sensornet Chip
(TI MSP430 8MHz
core, 10KB RAM)

[2009]
▪Tiger 1GHz Single-Core
▪Dunnington 3GHz 6-core

[2012]
▪ARM 2GHz 4-core
▪Intel 4GHz 32-core

[2017]
▪ARM 3GHz 8-core
▪Intel 6GHz 128-core
▪SensorNet Chip
(128MHz core, 160KB RAM)

"Privacy"

"Realtime"

SAMSUNG

Industry Trends

- **Introduction of Virtualization Technology in Embedded Devices**
- **Strengthening of Smartphone Features**

OS / Middleware

Ubiqitous Instant Boot (Android quick boot:

Wind River Acquisition (VxWorks, RTLinux)

Symbian OS Open source (2010.02)

Google Andriod

Linux based mobile OS (2010.01)

MS Widnows Phone 7 ('2010 4Q)

Google Chrome OS ('2010 4Q)

Virtualization

Trango Acquisition (2008,11)

XenDesktop / XenApp
Desktop/App. Virtualization

VMWare MVP (2009.01)

VirtualLogix VLX for ARM RTOS. Mpcore (2010. 02)

Nirvana Phone (Virtual Desktop w/ Phone(2011)

System Security

Apple iOS Sandbox

Google Andriod Sandbox & Permission-based Access Control

Google Chrome Browser Sandbox & Renderer Process Isolation

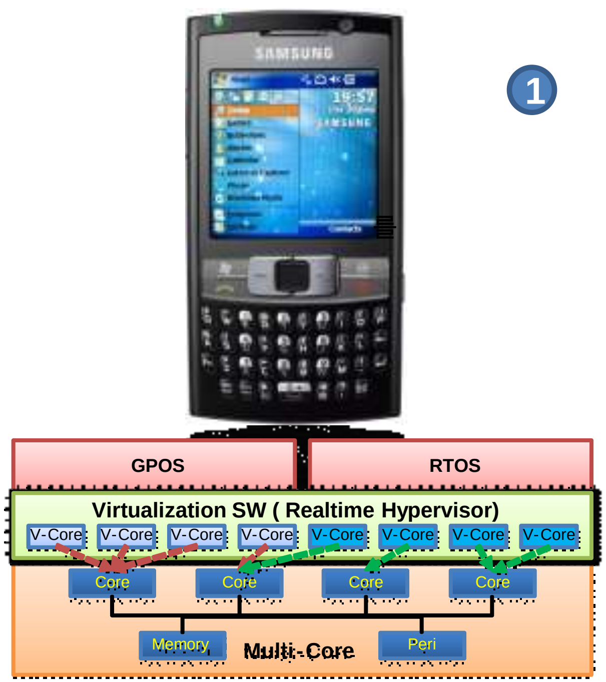
* RTM : Root of Trust Measurement

Why CE Virtualization?

1 – **HW Consolidation:** AP(Application Processor) and BP(Baseband Processor) can share multicore ARM CPU SoC in order to run both Linux and Real-time OS efficiently.

2 – **OS Isolation:** important call services can be effectively separated from downloaded third party applications by Xen ARM combined with access control.

3 – Rich User Experience: multiple OS domains can run concurrently on a single smartphone.



AP SoC +BP SoC -> Consolidated Multicore SoC

Secure Smartphone

Rich Applications from Multiple OS

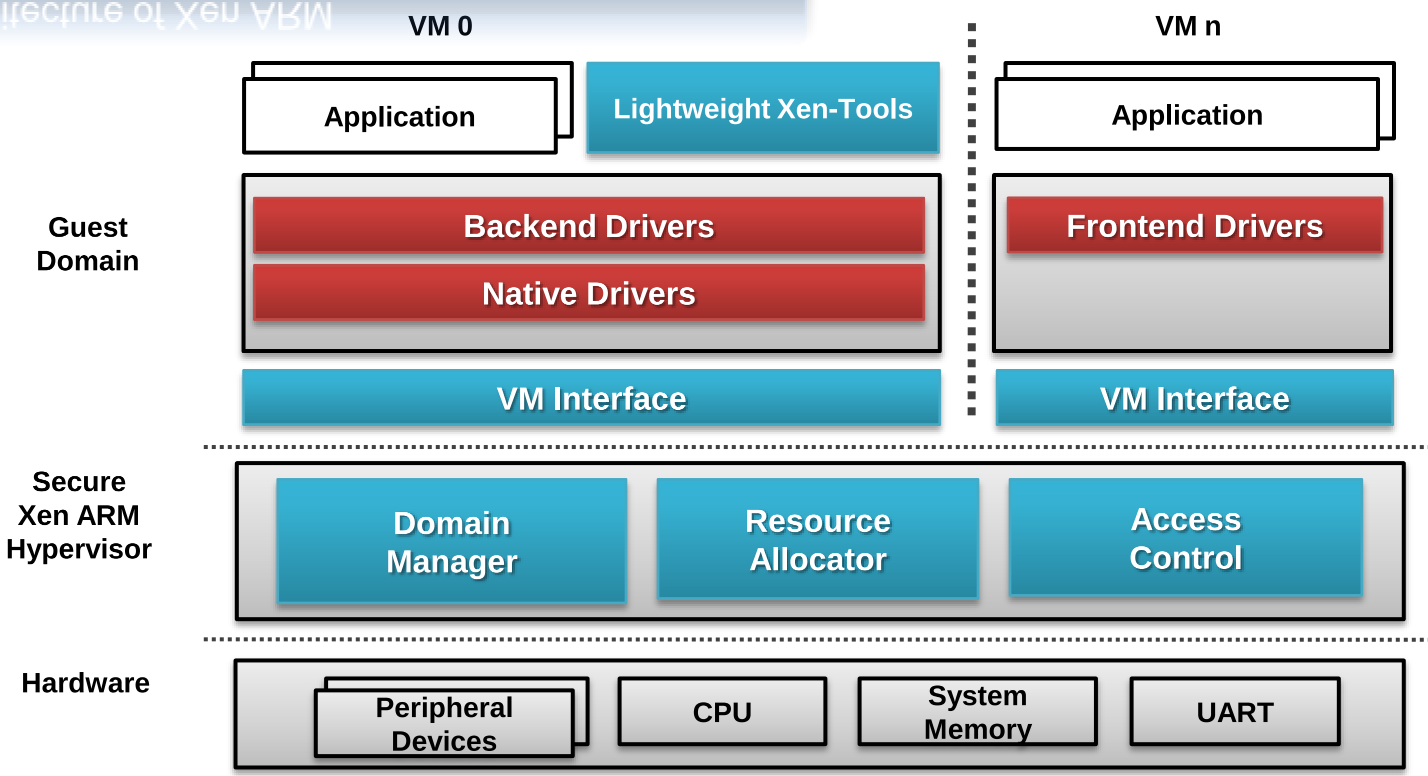
SAMSUNG

Xen ARM Virtualization

Goals

- Lightweight virtualization for secure 3G/4G mobile devices
  - High performance hypervisor based on ARM processor
  - Fine-grained access control fitted to mobile devices

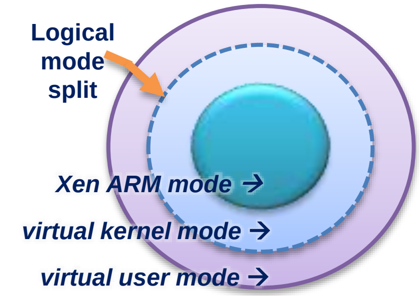
Architecture of Xen ARM



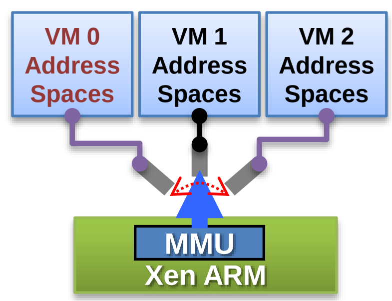
Xen ARM Virtualization

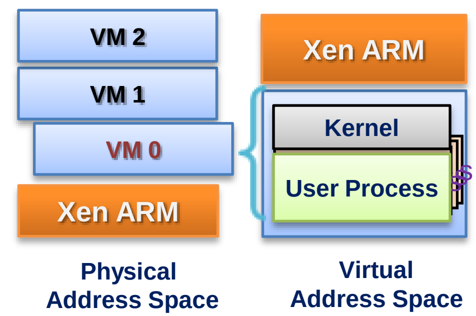
Overview

- **CPU virtualization**
  - Virtualization requires 3 privilege CPU levels, but ARM supports 2 levels
    - Xen ARM mode: supervisor mode (most privileged level)
    - Virtual kernel mode: User mode (least privileged level)
    - Virtual user mode: User mode (least privileged level)

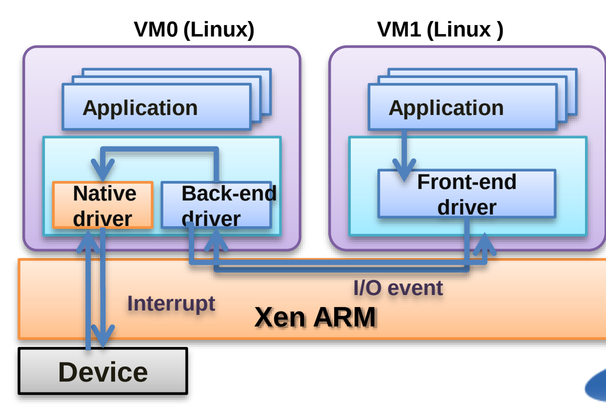


- **Memory virtualization**
  - VM's local memory should be protected from other VMs
    - Xen ARM switches VM's virtual address space using MMU
    - VM is not allowed to manipulate MMU directly





- **I/O virtualization**
  - Split driver model of Xen ARM
    - Client & Server architecture for shared I/O devices
      - Client: frontend driver
      - Server: native/backend driver



Performance Evaluation

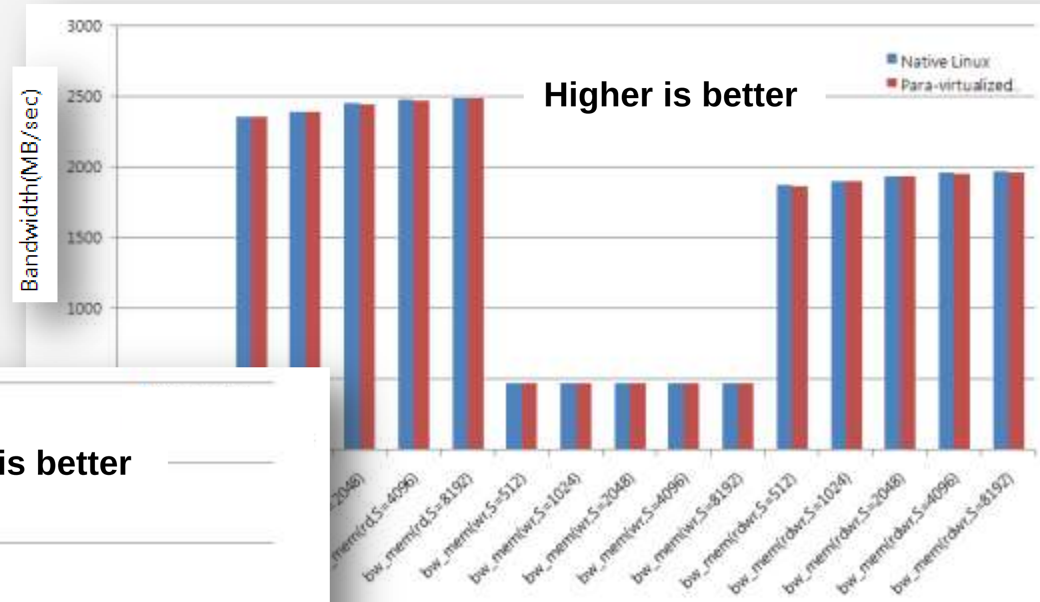
Virtualization Overhead

Micro-benchmark Results

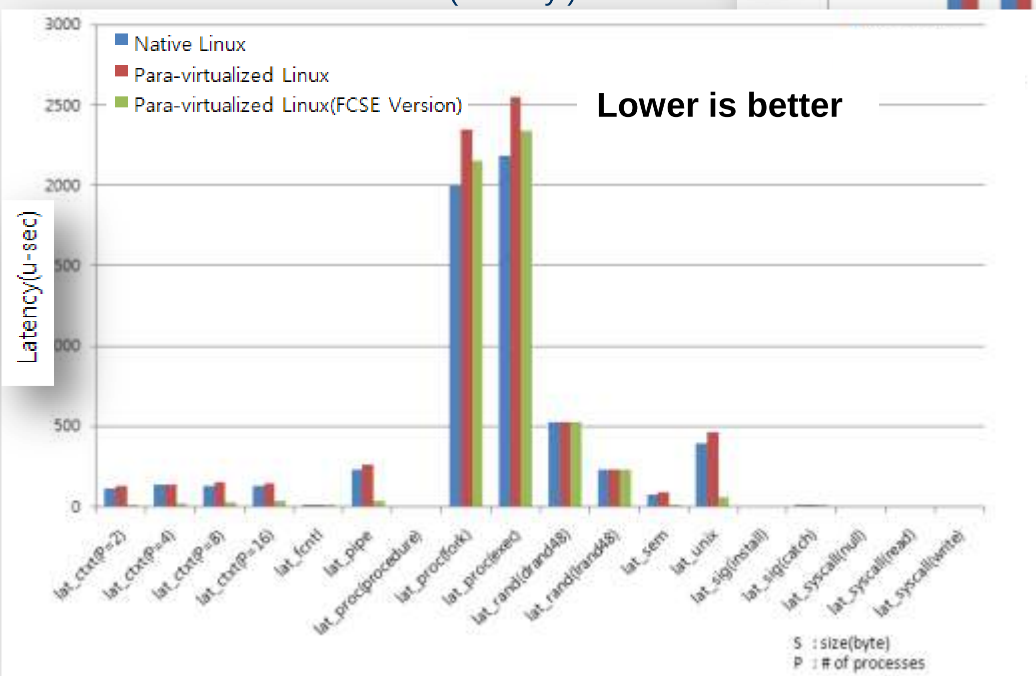
Micro-benchmark Results

- **Evaluation Environments : Samsung Blackjack Phone**
  - CPU : Xscale PXA310, 624MHz
  - L1 Cache : 32KB + 32KB
  - L2 Cache : 256KB (Disabled)
  - Memory : 128MB
  - Guest OS: Linux-2.6.21

LMBENCH Micro Benchmark (Bandwidth)



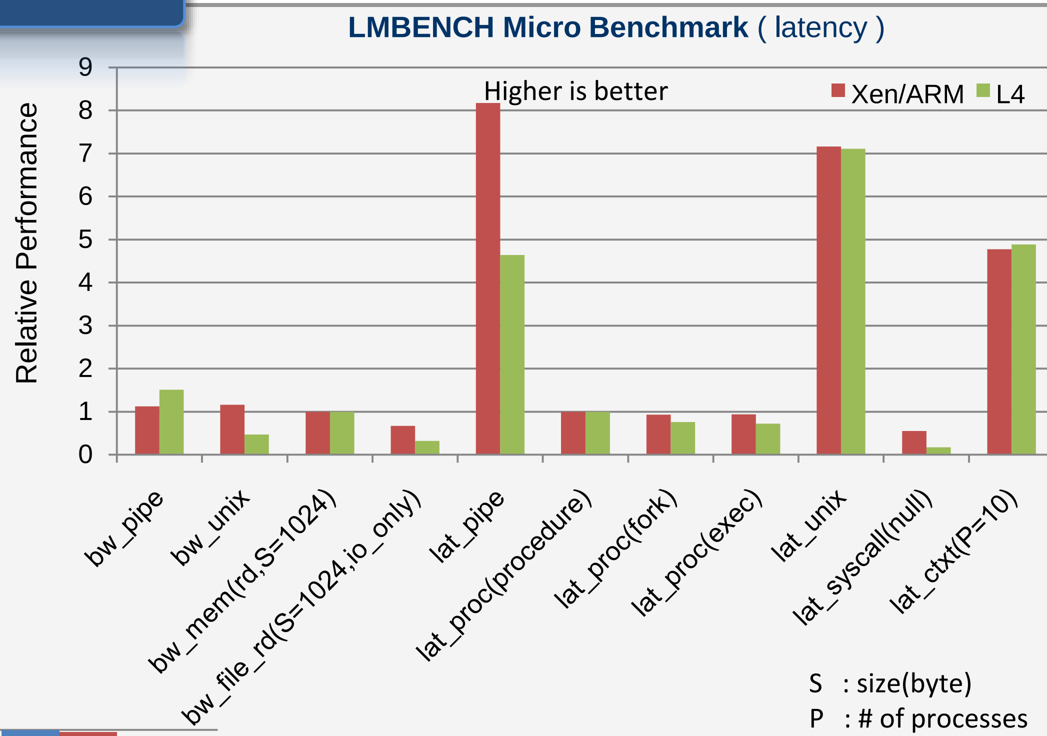
LMBENCH Micro Benchmark (latency)

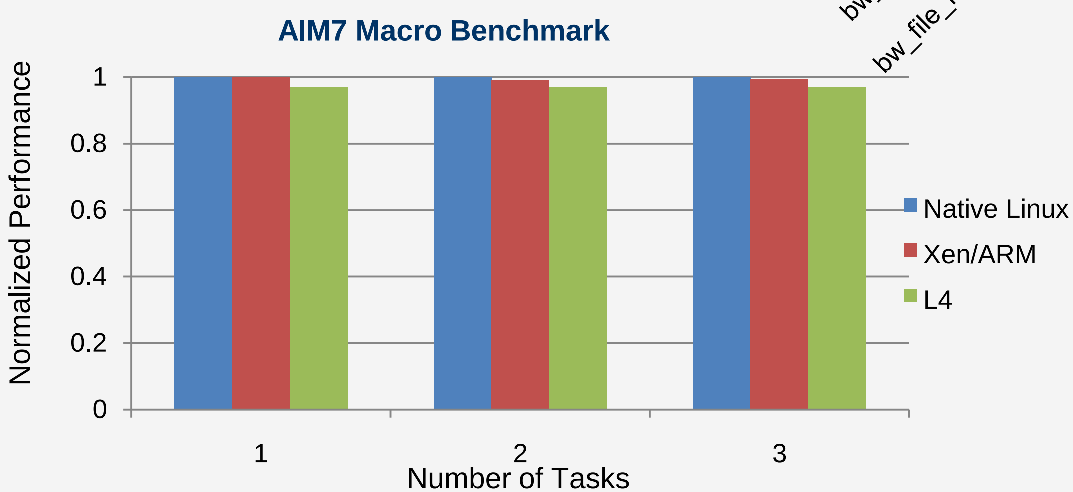


Virtualization Overhead Comparison

Benchmark Results

- **Evaluation Environments : Samsung Blackjack Phone**
  - CPU : Xscale PXA310, 624MHz
  - L1 Cache : 32KB + 32KB
  - L2 Cache : 256KB (Disabled)
  - Memory : 128MB
  - Guest OS: Linux-2.6.21



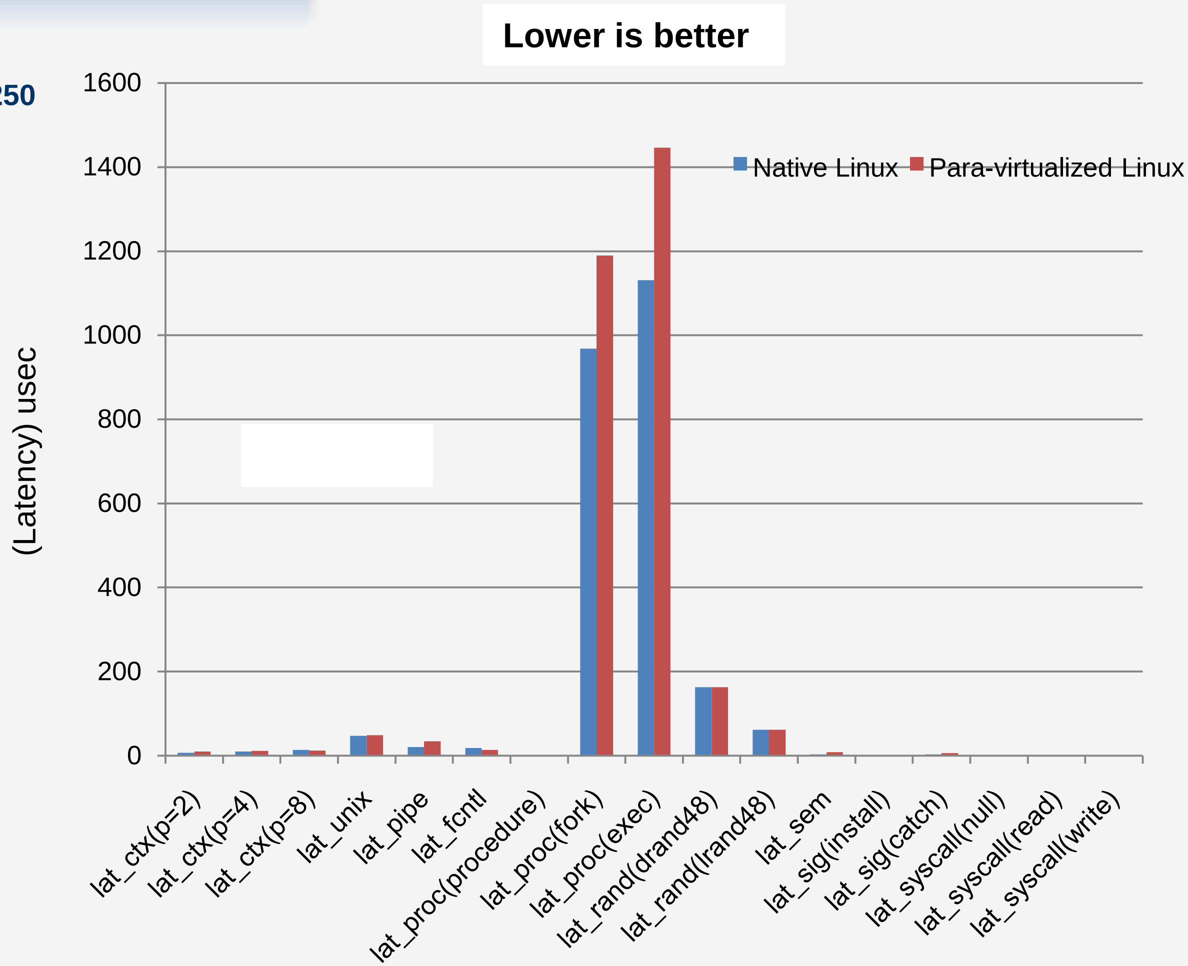


Performance Comparison

Micro-benchmark Results

Micro-benchmark Results

- **Evaluation Environments : nVidia Tegra250**
  - CPU : Cortex-A9 1GHz Dual Core
  - L1 Cache : 32KB + 32KB
  - L2 Cache : 1MB
  - Memory : 1GB
  - Guest OS: Linux-2.6.29



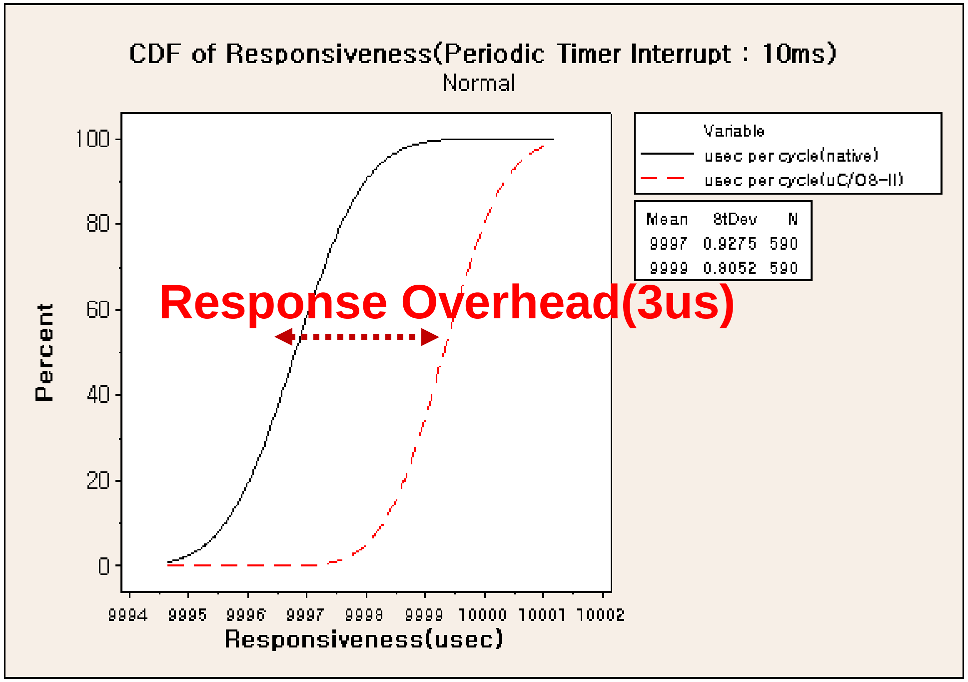
LMBENCH Micro Benchmark (latency)

Real-time Performance

- **Evaluation Environment**

Category		Description
H/W (Tegra250)	CPU	Cortex-A9 / 1GHz / Dual Core
	RAM	1GB
S/W	Hypervisor	Xen ARM
	Guest OS (DOM0)	Linux-2.6.29 (Running Busy Loop Task)
	Guest OS (DOM1)	uC/OS-II (Running RT Task : Cyclictest benchmark)

- **Cyclictest benchmark repeats**
  1. RT task sleeps for 10ms
  2. Timer interrupt will occur after 10ms
  3. Timer interrupt wakes up the RT domain(uC/OS-II)
  4. uC/OS-II preempts Xen ARM
  5. RT task is scheduled
  6. RT task logs timestamp

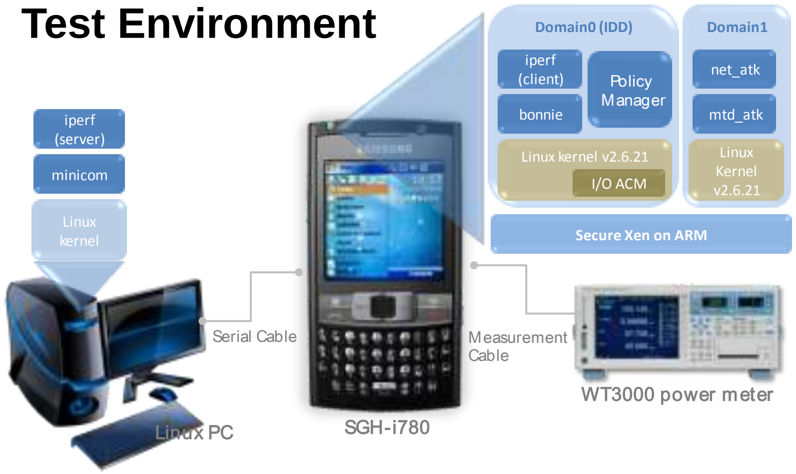


Native(uC/OS-II)		
Min	Avg	Max
9995	9996.810169	10000
Xen ARM(uC/OS-II)		
Min	Avg	Max
9996	9999.327119	10001

Unit : usec

Effectiveness of Access Control

Test Environment

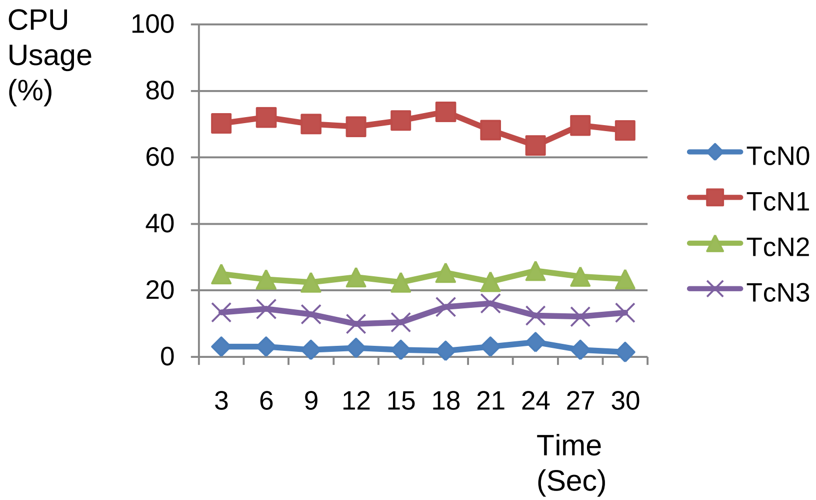


net_atk: UDP packet flooding (sending out UDP packets with the size of 44,160 bytes every 1000 usecs)
mtd_atk: overwhelming NAND READ operations (scanning every directory in the filesystem and reading file contents)

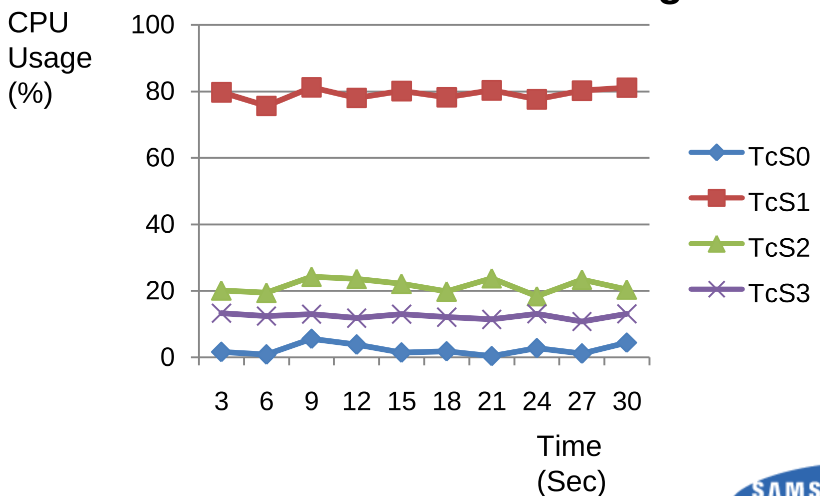
Test Cases

	Network I/O Test Cases	Storage I/O Test Cases
No Attack	TcN0	TcS0
Under Attack (No I/O ACM)	TcN1	TcS1
Under Attack (20% I/O ACM Policy)	TcN2	TcS2
Under Attack (10% I/O ACM Policy)	TcN3	TcS3

CPU Utilization: Network

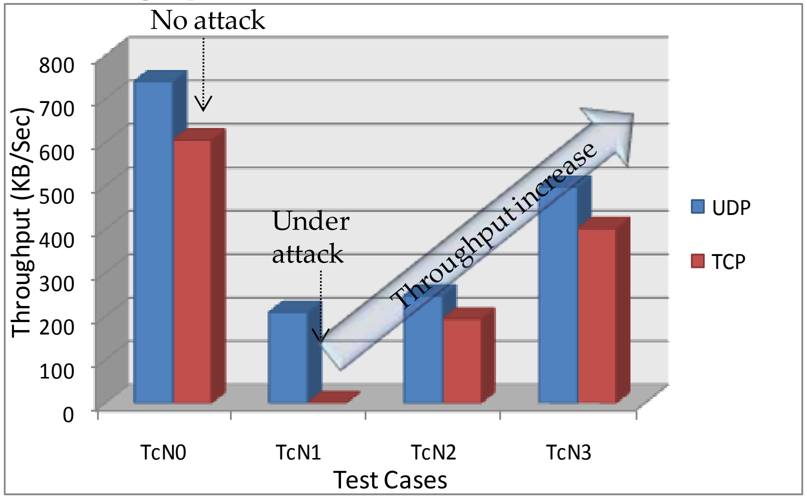


CPU Utilization: Storage

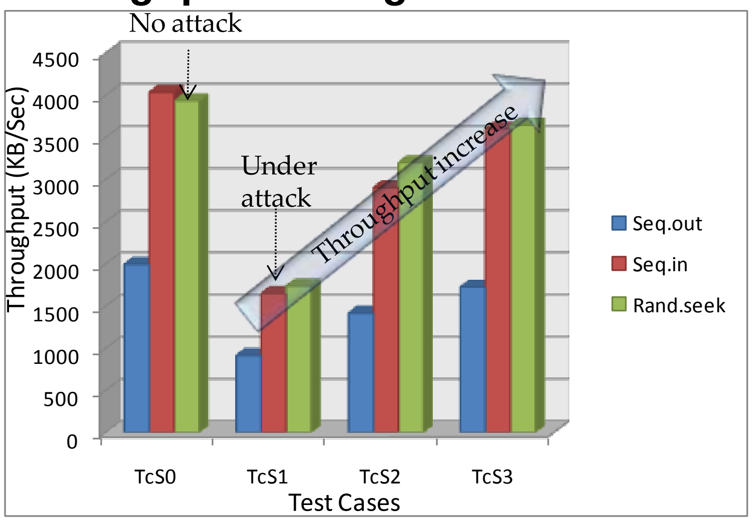


Effectiveness of Access Control

Throughput: Network

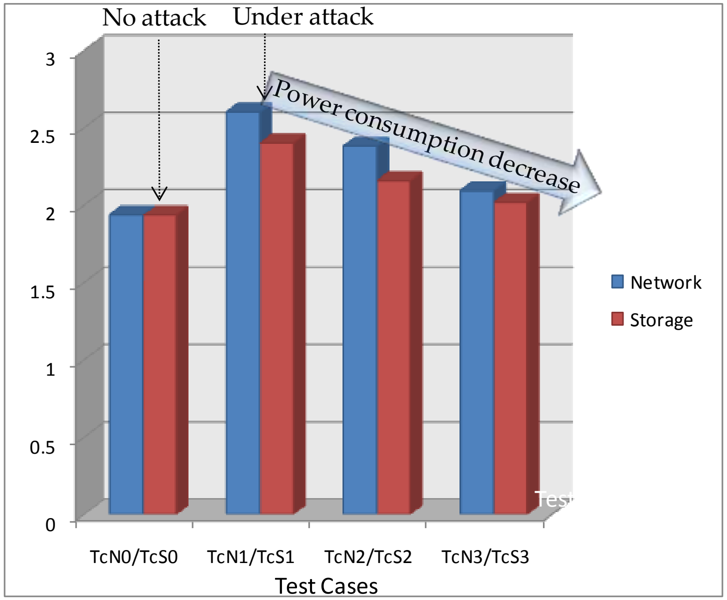


Throughput: Storage



- **Effectiveness of our access control: throughput increase and power consumption decrease even under malware attack**

Power Consumption



History of Xen ARM

- **'04** — **x86 Xen Hypervisor Release** (Cambridge University)
- **'08** — **Xen ARM 1st Release:** ARM9 Xen Hypervisor, Mini-OS (Samsung)
- **'09** — **Xen ARM 2nd Release:** Paravirtualized Linux kernel (v2.6.24), Xen tool (Samsung)
- **'10** — **Xen ARM 3rd Release:** ARM11MPCore Support (Samsung)
- **'11** — **Xen ARM 4th Release:** Performance Optimization (Samsung)
- **Xen ARM 5th Release:** Cortex-A9 MPCore Support (Samsung)

Xen ARM Open Source Community

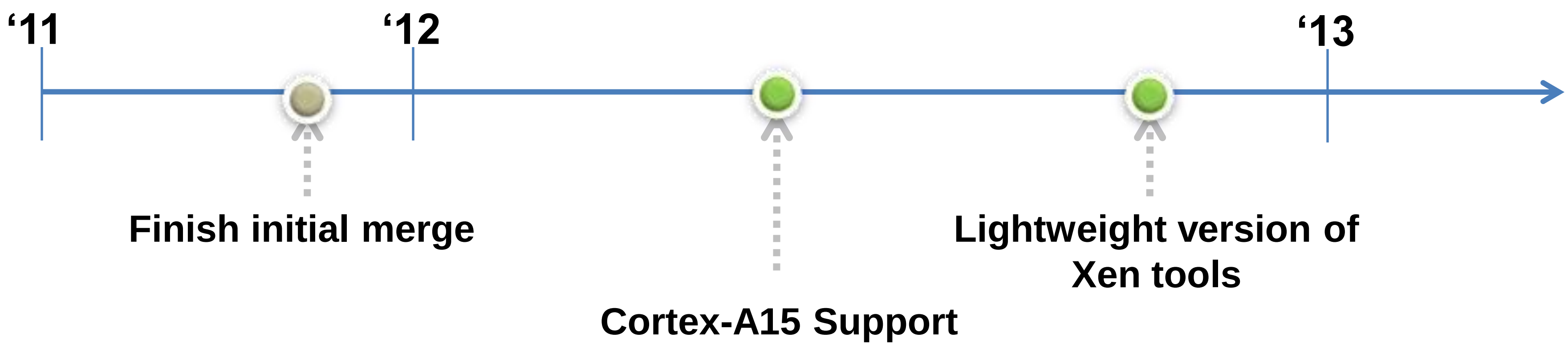
- http://wiki.xensource.com/xenwiki/XenARM

Supported Hardware & Guest OS(Stand-alone Version)

- **ARM926EJ-S (i.MX21, OMAP5912)**
- **Xscale 3rd Generation Architecture (PXA310, Samsung SGH- i780)**
- **ARM1136/ARM1176(Core Only)**
- **Goldfish (EQMU Emulator)**
- **Versatile Platform Board**
- **ARM11MPCore (Realview PB11MP)**
- **Tegra250**

- **Linux v2.6.11, v2.6.18, v2.6.21, v2.6.24, v2.6.27 (multicore supported)**
- **uC/OS-II**

Future Roadmap of Xen ARM

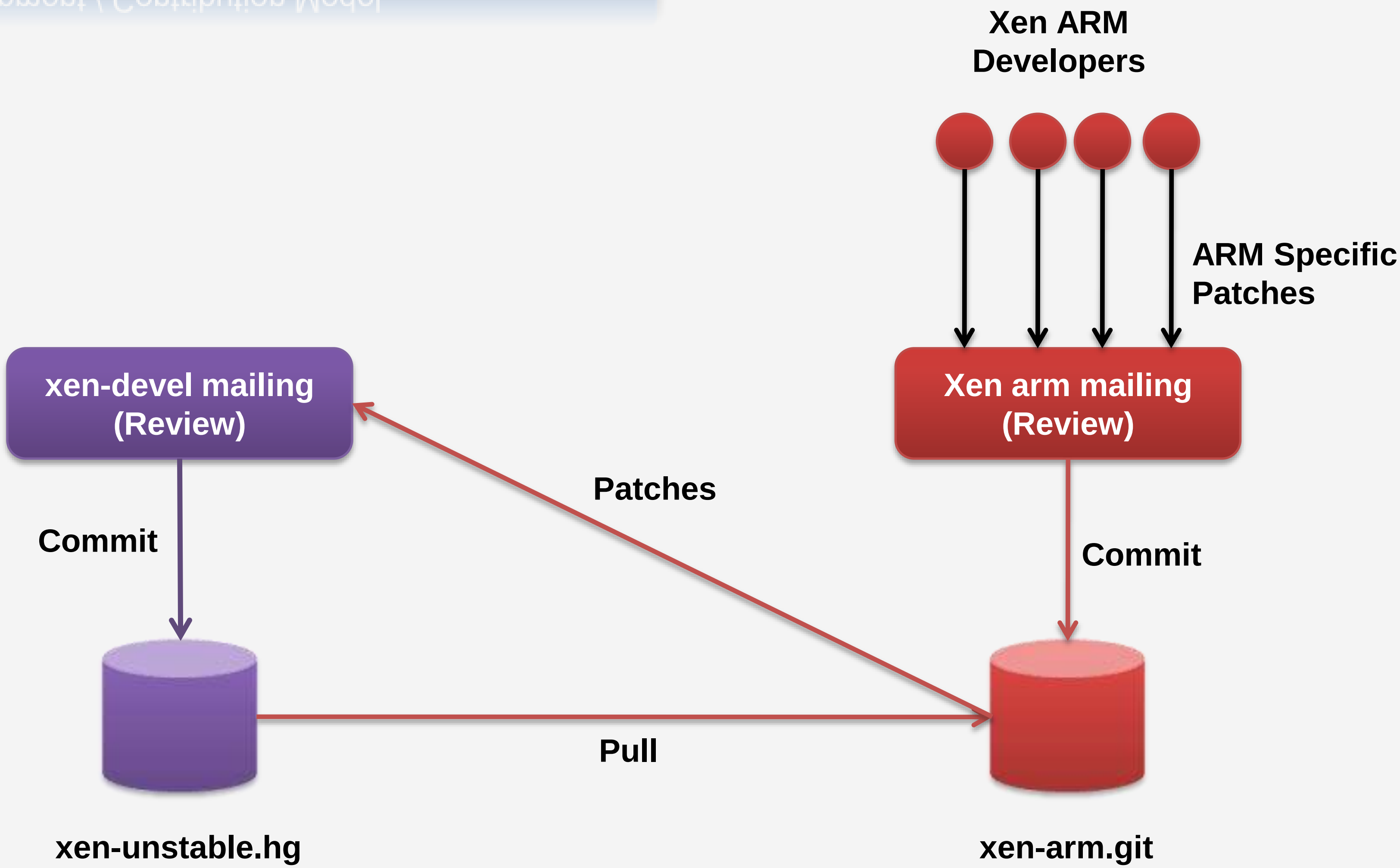


Mainline Merging

- **Integration of Xen ARM with mainline (80% completed)**
  - Rebased on the recent xen-unstable.hg
  - Many parts of the Xen ARM has been rewritten for the integration.
- **Dynamic domheap allocation**
  - Support of "pseudo-physical to machine translation" is ongoing.
- **Dynamic xenheap expansion**
  - Xenheap could be expanded on demand
    - Initially Xen ARM reserves 1MB(1 Section) of memory for heap

Xen ARM Development / Contribution Model

Development / Contribution Model



Issues

- **Xen-Tools**
  - Porting to ARM architecture is required
    - Currently libxc does not support ARM architecture.
- **Real-time**
  - Implementing Real-time Scheduler
    - How does the VMM knows which domain requires real-time scheduling?.
  - Implementing VMM Preemption
    - How to minimize interrupts and event latency within the view of VM? (for VM perspective)
- **Access Control**

Thank You !

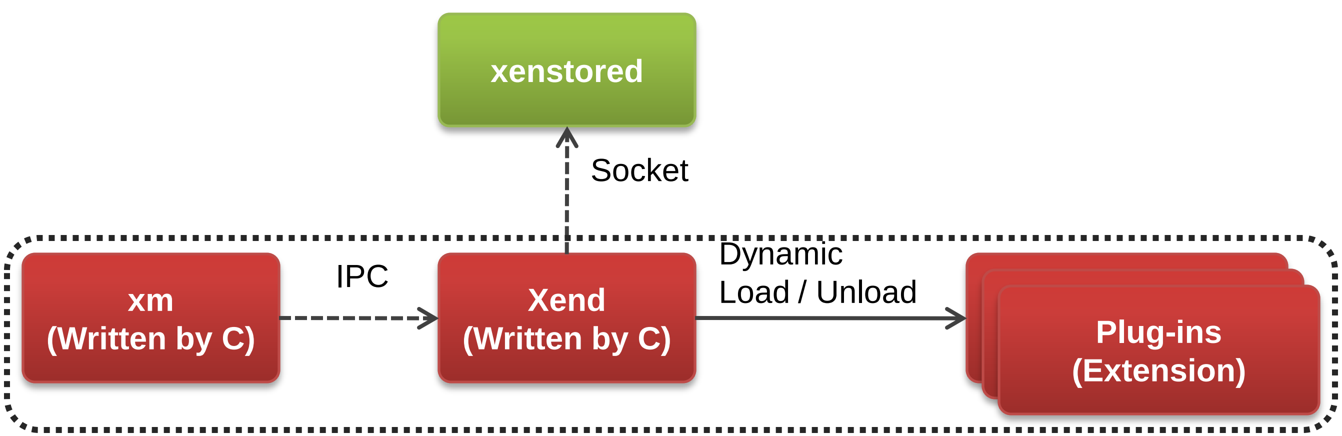
Issue: Xen-Tools

Lightweight version of Xen-tools

- Python-based xend/xm too heavy for small devices.

- Lightweight version of xend/xm for embedded devices
  - Adopt Plug-in architecture
    - To avoid re-compilation when new virtual device introduced.

	Python-based Xm/Xend	
Memory Usage	Several tens of MB	Several hundreds of KB.
Latency	Several seconds	< 1 second

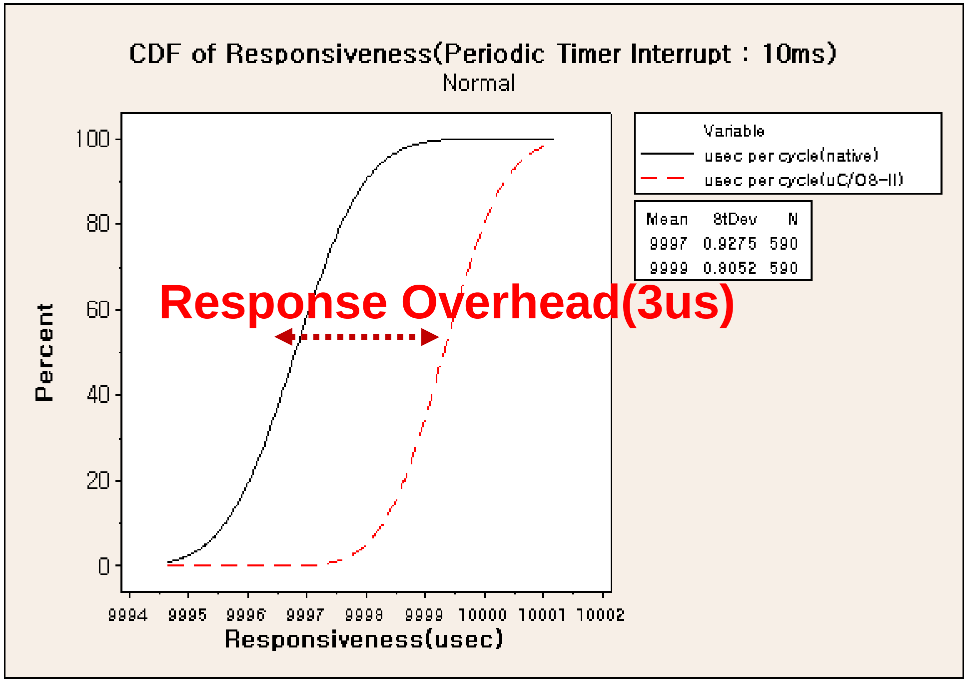


Issue: Real-time vs. Throughput

- **Evaluation Environment**

Category		Description
H/W (Tegra250)	CPU	Cortex-A9 / 1GHz / Dual Core
	RAM	1GB
S/W	Hypervisor	Xen ARM
	Guest OS (DOM0)	Linux-2.6.29 (Running Busy Loop Task)
	Guest OS (DOM1)	uC/OS-II (Running RT Task : Cyclictest benchmark)

- **Cyclictest benchmark repeats**
  1. RT task sleeps for 10ms
  2. Timer interrupt will occur after 10ms
  3. Timer interrupt wakes up the RT domain(uC/OS-II)
  4. uC/OS-II preempts Xen ARM
  5. RT task is scheduled
  6. RT task logs timestamp



Native(uC/OS-II)		
Min	Avg	Max
9995	9996.810169	10000
Xen ARM(uC/OS-II)		
Min	Avg	Max
9996	9999.327119	10001

Unit : usec

Issue: Access Control

sHype, XSM and our ACM

sHype, XSM and our ACM

	sHype[SAI05]	XSM [COK06]	Xen ARM ACM
Access Control Policies	Flexible based on Flask(TE and Chinese Wall)	Flexible based on Flask(TE and Chinese Wall, RBAC, MLS, and MCS)	Flexible based on Flask(TE and proprietary policy)
Objects of Access Control	Virtual resources and domain management	Physical/virtual resources and domain management	Physical/virtual resources and domain management
Protection against mobile malware-based DoS attacks	N/A	N/A	Memory, battery, DMA, and event channels are controlled by ACM
Access control to objects in each guest domain	Enforced by ACM at VMM	Enforced by ACM at VMM	Enforced by ACM at each domain(for performance reason)
Etc			Xen ARM specific hooks

Comparison of ARM vs. x86 Virtualizability

Comparison

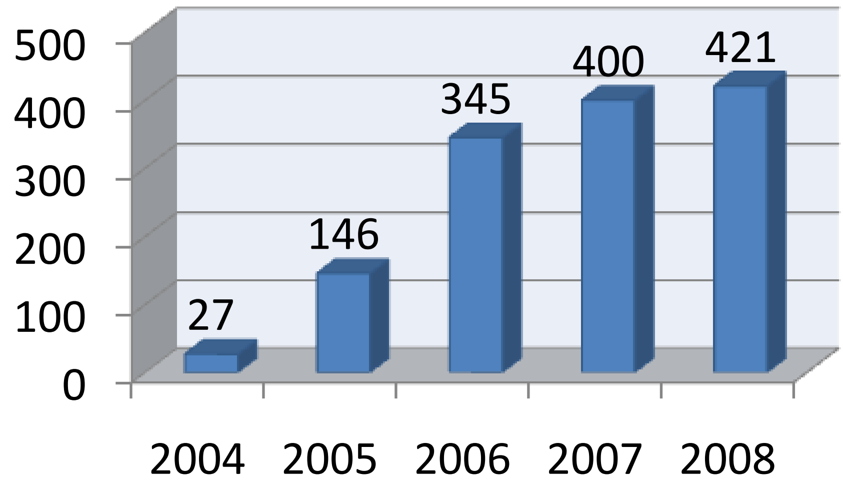
Comparison

	x86	ARM
Ring Compression (Protection mechanisms)	Segmentation and Paging	Paging and Domain Protection
Cache Architecture	PIPT	VIVT / VIPT / PIPT
I/O	I/O Instructions + memory-mapped I/O	Only memory-mapped I/O
# of privilege levels	4	2

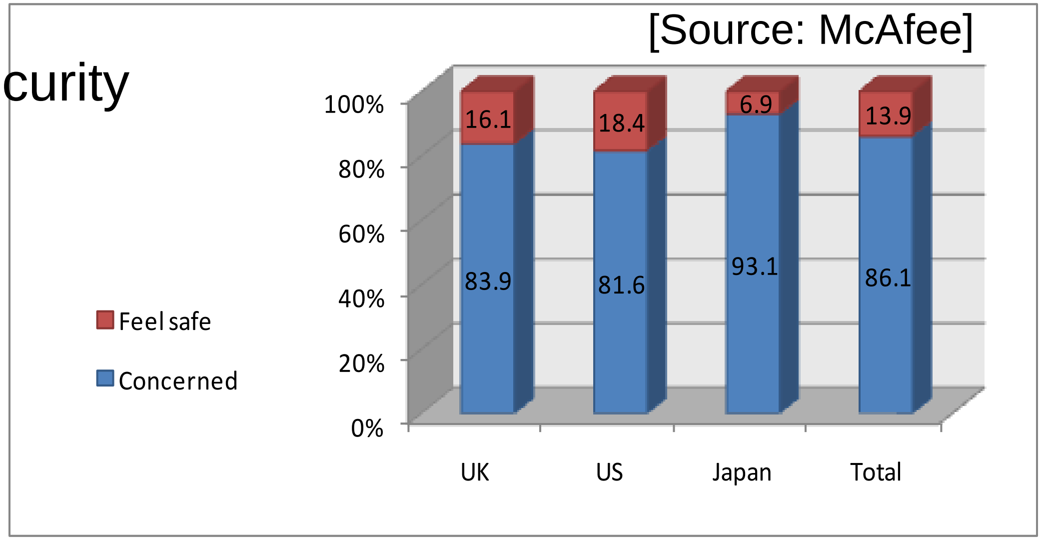
Mobile Malware

- Number of mobile malware
  - More than 420 mobile phone viruses (2008)
  - Tens of thousands of infections worldwide

[Source: F-Secure]



- Concerns about mobile phone security – by market



Current Status of Xen ARM

Changeset

Common files which have been modified

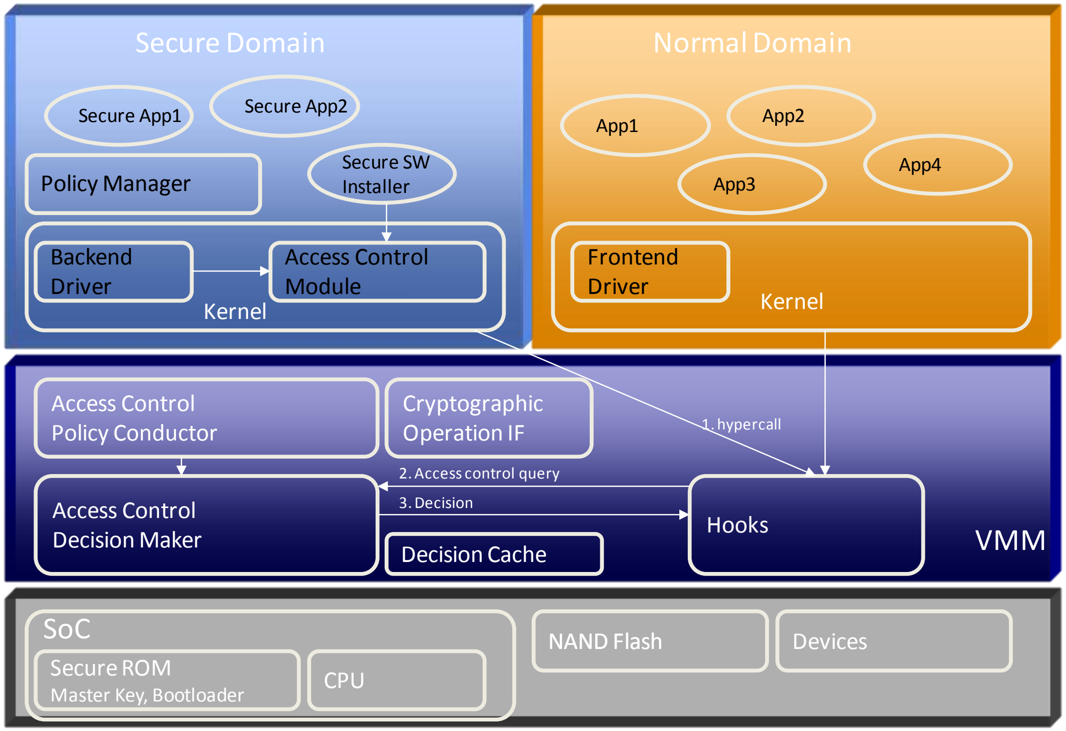
Directory	File	Comment
xen	Rules.mk	- override TARGET_SUBARCH := $(XEN_TARGET_ARCH) **+ override TARGET_SUBARCH := $(XEN_TARGET_SUBARCH)**
xen/common	page_alloc.c	Add reserve_boot_pages() function
xen/drivers	Makefile	Exclude x86 dependent device drivers when Xen is built for ARM architecture
xen/include/public	Xen.h	Add preprocessor macros to include arch-arm.h header file.
xen/include/xen	libelf.h	Add preprocessor macros to support ARM architecture.

New files

- We wrote xxx files for ARM architecture

Xen ARM Access Control

- Protect unauthorized access to system resources from a compromised domain



- **37 access control enforcers in hypercalls**
- **Flexible architecture based on Flask**
  - **Currently, 5 access control models supported (TE, BLP, Biba, CW, Samsung Proprietary)**
- **Access control of the resources**
  - **Physical resources (TE, Samsung Proprietary)**
    - **Memory, CPU, I/O space, IRQ**
  - **Virtual resources (TE, BLP, Biba)**
    - **Event-channel, grant table**
  - **Domain management (CW)**
    - **Domain creation/destroy**