Virtualization:
how far can it go

Chuck Yoo

Korea University

hxy@os.korea.ac.kr

2013.10.11

Contents

- **Background**
- **Applicable Fields of Virtualization**
  - **Automotive Software**
    - TrustZone based Automobile Virtualization Platform
  - **Storage**
    - SSD scheduler for VMs on cloud
  - **Network**
    - New virtual router: Xebra
- **Future**

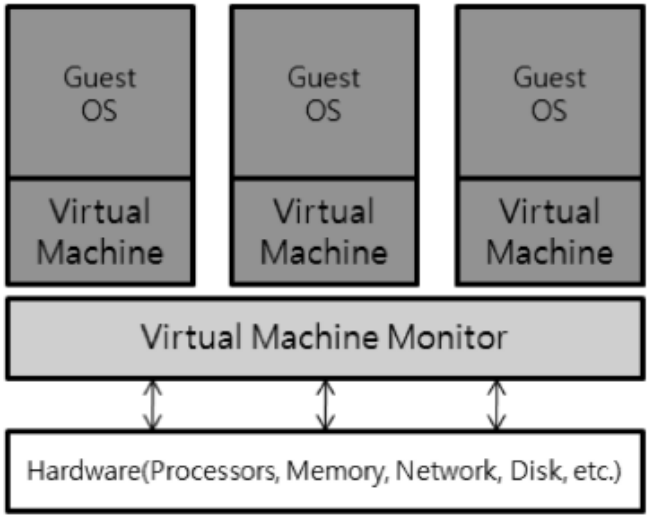
Background

Introduction

- **General meaning**
  - An abstraction of resources that provides a logical rather than an actual physical incarnation of those resources
  - Such resources are: CPU, storage, network, etc.
- **Introduced to fully utilize mainframes in 1960s**
  - Could run several projects on a single mainframe
- **Has been extended to several areas**
  - System Virtualization → Consumer Electronics, Automobile
  - Storage Virtualization → Cloud Computing
  - Network Virtualization → Future Internet

Virtualization Component

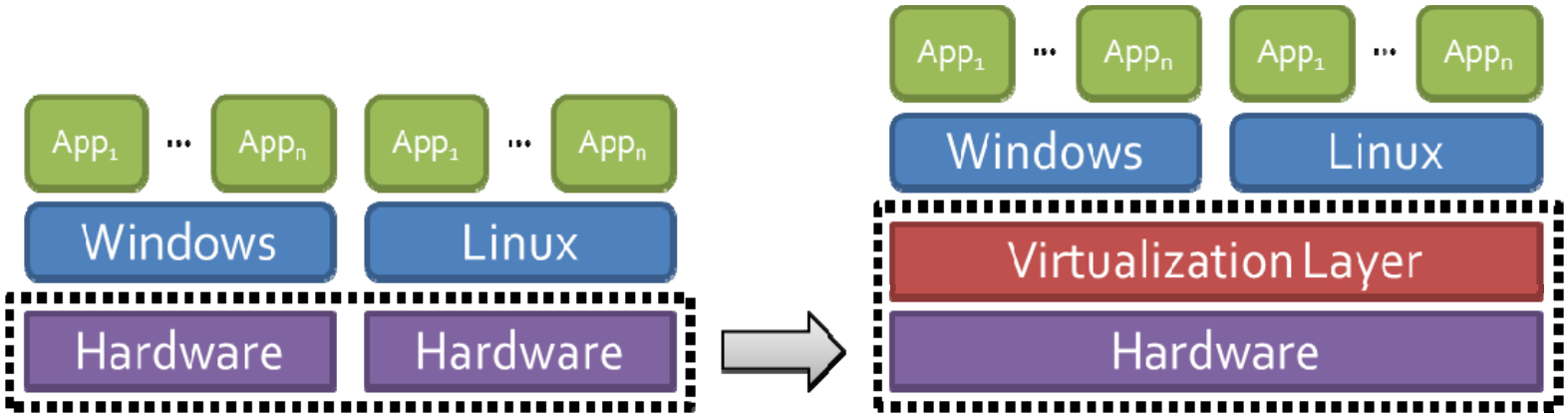
- **Virtual Machine (VM)**
  - Independent software execution environment
  - Guest OS is operated on VM



- **Virtual Machine Monitor (VMM) or Hypervisor**
  - Control physical hardware
  - Management system for VMs

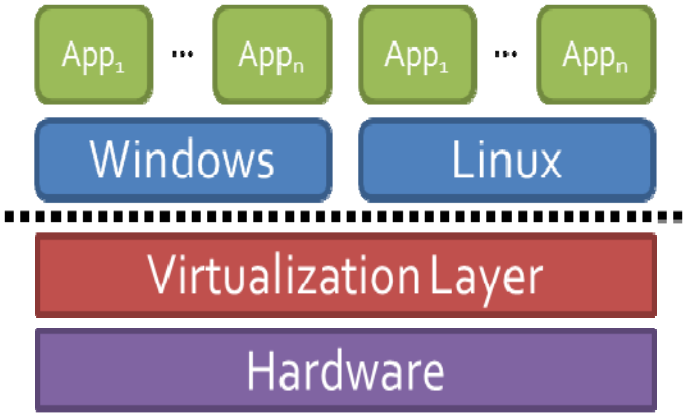
Advantage of Virtualization – Consolidation

- A physical machine can operate multiple logical systems
  - Reduce hardware cost
- e.g. Typical utilization of web server is about 15%
  - Can operate 6~7 VMs in a physical machine

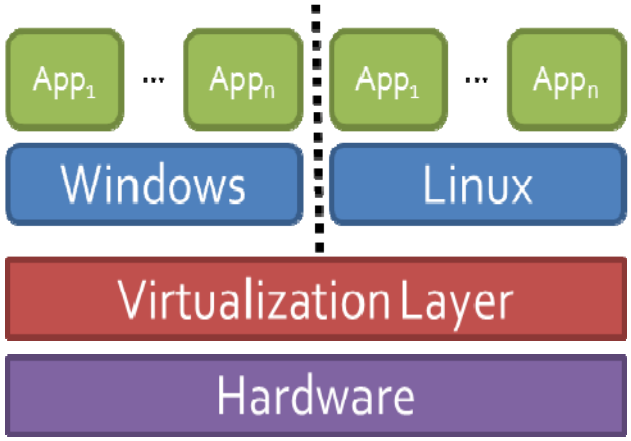


Advantages of Virtualization – Decoupling & Isolation

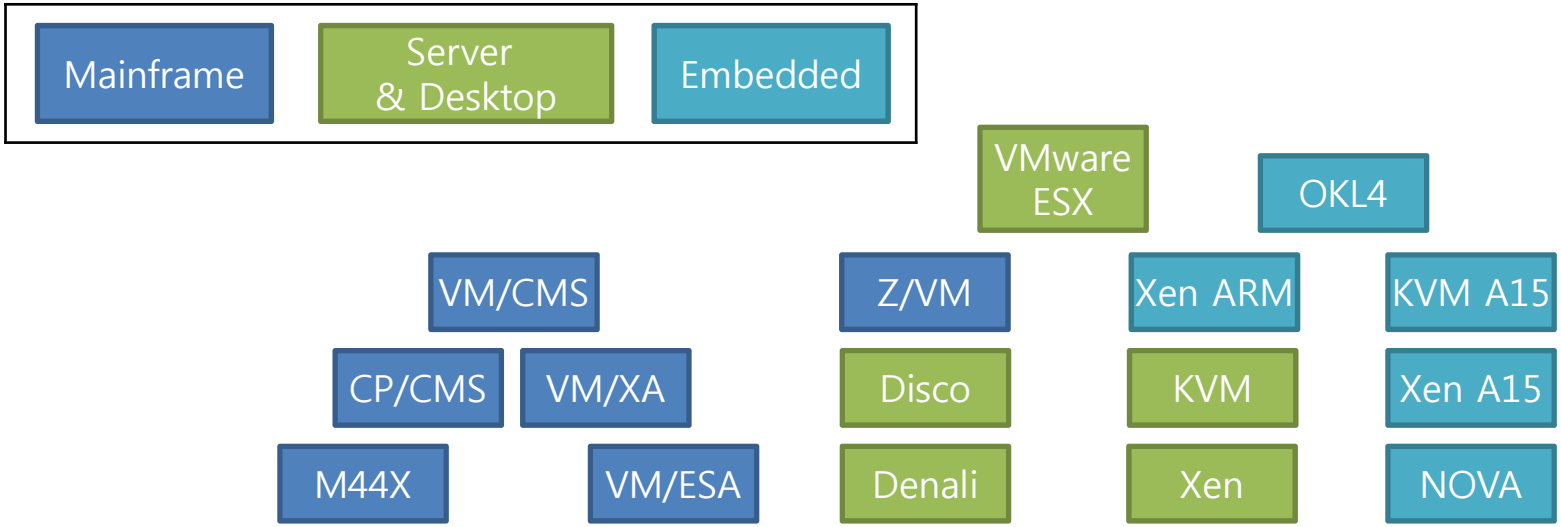
- **Decoupling**
  - Eliminate dependency between software and hardware
    → Increase software portability
    e.g.) Applications can run on new hardware without modification



- **Isolation**
  - Guest OS are isolated from each other
    → Increase system reliability
    e.g.) Device driver failure does not affect other VMs



History of hypervisor



Generation	1st	2nd	3rd	4th
Year	1960~1990	1990~2000	2000~2010	2010~2020
Main technology	• Early virtualization technology	• Supporting trap of sensitive instruction	• Paravirtualizing Intel/ARM processor • Binary conversion • Intel VT HW virtualization	• ARM A15 HW virtualization • Virtualizing I/O device (such as SR-IOV) • Virtualizing GPGPU

Applicable Fields

- **Embedded System Virtualization**
  - Support real-time service, low-power design
  - Mobile phone & automobile
- **Server Virtualization**
  - I/O virtualization is a key
  - Foundation for Cloud computing: OpenStack
- **Network Virtualization**
  - Support virtual networks of different traffics
  - Virtual router

Automotive Software

- TrustZone based Automobile Virtualization Platform-

Paradigm shift: Running software

- **Since 2008, the number of automobile company participating CES* continues to grow every year**
  - **CEO of Benz 2012 CES* Keynote address**
    - IT tech. is main stream of automobiles industry
  - **'Car and Digital Realm': focused on connectivity through Internet in a car.**

*CES: Consumer Electronics Show

- **Running Machine → Running Electric Devices → Running Software**
  - **A future automobile is the result of IT integration**
    - **Android-on-car**
    - **Google-car**

(*Google's Autonomous Car Takes To The Streets*, IEEE Spectrum, 2010)

Similar problems in Avionics

- **Problem**
  - **In AVIONICS*, huge size and increased complexity of hardware and software**
    - One-Function One-Computer
  - **Difficult maintenance and high cost**
- **Solution**
  - **Using consolidation technique of virtualization**
    - Software functions running on several H/W can be integrated on one hardware
    - Other software functions can be protected from a software fault by consolidation
  - **Using software reuse technique of virtualization**
    - Legacy software can run on virtual machine without modification
- **Example: AIRBUS(11. 2008)**
  - **Using PikeOS embedded virtualization solution**
    - The software of next-generation Airbus will be designed as Module enabling to operate on virtualization solution of PikeOS

*AVIONICS: Aviation Electronics Equipment

State of virtualization in automobile – Volkswagen, TOYOTA

- **Background**
  - **Need for connectivity of automobile software with outside network**
  - **Enhancing security of automobile software**

- **Oversee Project (Volkswagen)**
  - **Open vehicular secure platform**
  - **Providing software execution environment using virtualization**
    - Separate VM runs OS that support additional software
    - Guaranteeing secure access to internal network
    - Providing secure communication channel on virtualization layer between software

- **SafeG Project (TOYOTA)**
  - **Dual-OS monitor development project**
  - **Executing both legacy RTOS and new GPOS on single processor**
    - Using the ARM TrustZone technology
    - Providing separated software execution environment for each OS

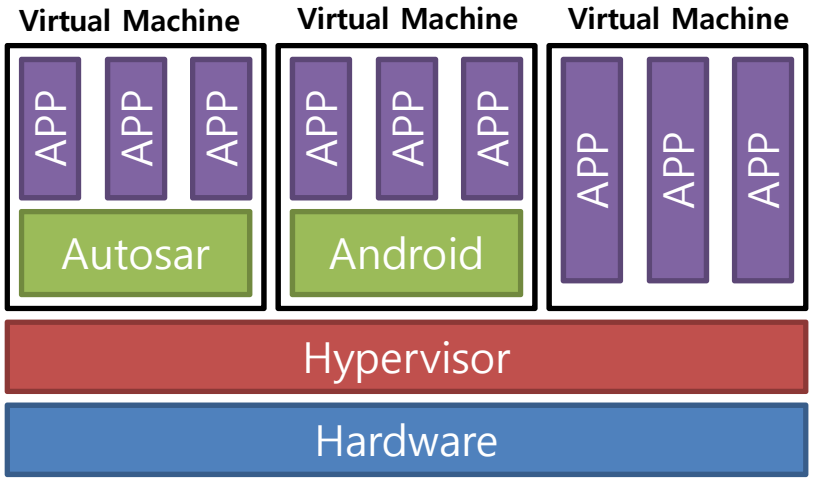
Automobile Virtualization Platform

Outline of automobile virtualization platform

- **Definition of automobile virtualization platform**
  - Virtualized software platform that supports electronic control software and infotainment software
- **Requirements of automobile virtualization platforms**
  - High reliability
  - Real-time support
  - High security
  - Support backwards compatibility and interoperability between software
- **Two approaches of automobile virtualization platforms**
  - Hypervisor based virtualization
  - ARM Trust-Zone based virtualization

Hypervisor based virtualization platform

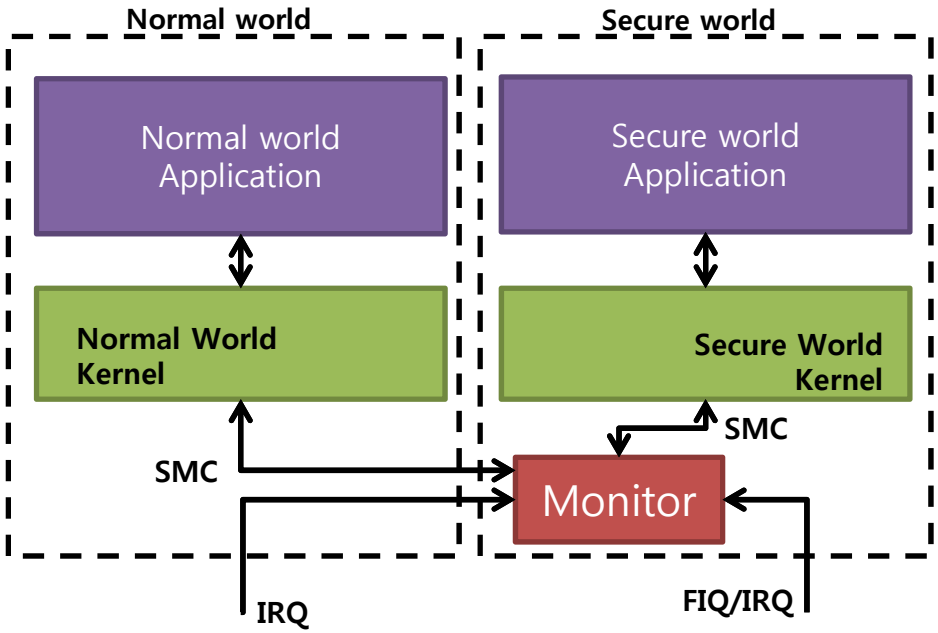
- **Executing guest OS, electronic control software and infotainment software on VM provided by hypervisor**



- **Design issues**
  - **Real-time support**
    - Both hypervisor and guest OS should support real-time
  - **additional privileged level on processor**
    - Para-virtualization increases software development cost
  - **Safety verification of hypervisor**
- **Commercial virtualization products**
  - **XenARM, KVM, OKL4, PikeOS and Redbend**

ARM TrustZone

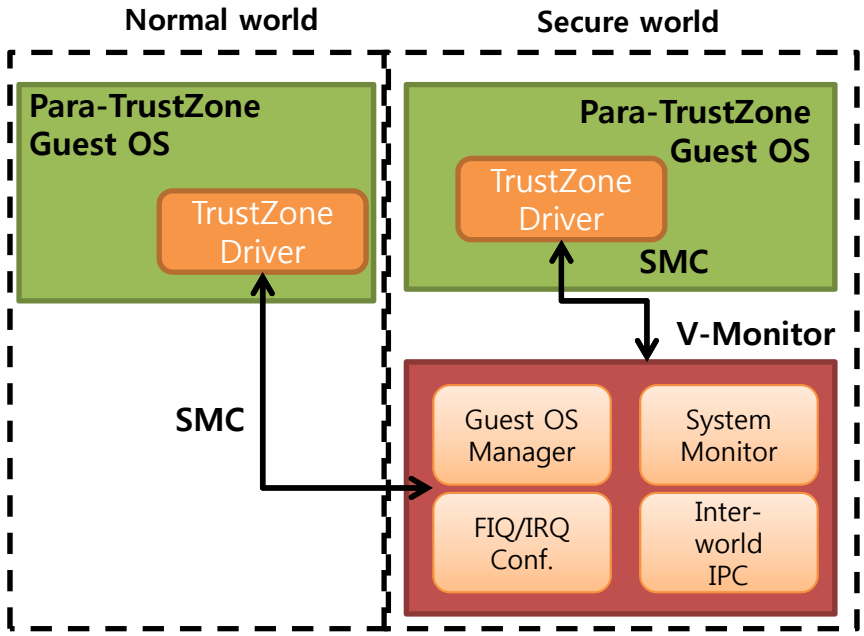
- **Divide processor into secure world and normal world**
  - The concept of world is orthogonal to User / Kernel mode of processor
- **SMC(Secure Monitor Call)**
  - Instruction switching the world where processor be executed (Secure ↔ Normal)
  - Used in implementing communication between two world
  - Only in the kernel mode, SMC can be executed
- **Monitor**
  - Manages processor context while world switching
  - Entering monitor by SMC, IRQ and FIQ



- **TrustZone provides function to partition computing resource on hardware level**

TrustZone based virtualization platform

- **Component of Virtualization platform**
  - "Para-TrustZone"- Guest OS
  - V-Monitor: Monitor supporting virtualization
- **"Para-TrustZone"- Guest OS**
  - Including TrustZone Driver
  - Lightweight than Para-Virtualization
    - No sensitive instruction modification
- **V-Monitor function**
  - Function to manage Guest OS (Guest OS Manager)
  - Support real-time Guest OS (FIQ/IRQ Configurator)
  - Function to monitor system (System Monitor)
  - IPC Service between the worlds (Inter-world IPC)



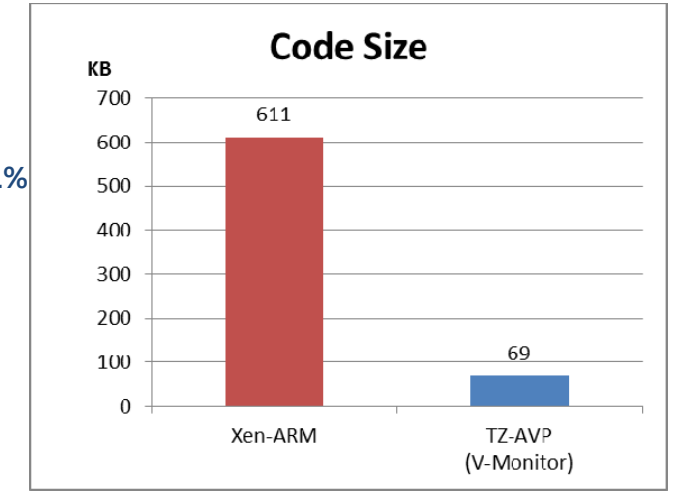
TrustZone based Automobile Virtualization Platform

TZ-AVP (TrustZone based Automobile Virtualization Platform)

- **TZ-AVP**
  - **H/W platform: Nvidia Tegra T30 Application Processor (ARM Cortex-A9 MPCore Architecture)**
- **V-Monitor**
  - **World scheduler (management of guest OS)**
    - Round Robin / Priority RT Scheduler
  - **Memory partition (management of guest OS)**
  - **FIQ/IRQ Configurator (Support real-time in Guest OS)**
    - FIQ → Secure world , IRQ → Normal world
  - **Monitor console (able to system monitoring)**
  - **SMC handler for IPC (IPC services)**
- **"Para-TrustZone" Guest OS**
  - **Secure world: uCOS-II (Single Core version)**
    - Add FIQ handler
    - Implement Secure TrustZone Driver (uCOS-II dependent)
  - **Normal world: Linux (SMP Version)**
    - Memory map modification
    - Mapping the address space of a permitted device to Normal world
    - Implementing TrustZone Driver (Linux dependent)
    - Supporting SMP

Advantages of TZ-AVP

- **Do not need hypervisor**
- **Code size of V-monitor**
  - Lower amount of code modification than that of Xen-ARM about 11%
  - Having advantage for monitor verification
- **Less modification of PT-guest OS**
  - Xen-ARM: Should modify own kernel to execute guest OS on CPU's User Mode
  - TZ-AVP: add only set-up routine related to Guest OS's memory map and TrustZone
- **Support real-time in V-Monitor**
  - TrustZone can define priority of interrupts toward each world
  - Secure world executes real-time guest OS such as AUTOSAR



Conclusion

- **Automobile virtualization platform**

→The core technology of future automobile which various IT services are grafted onto

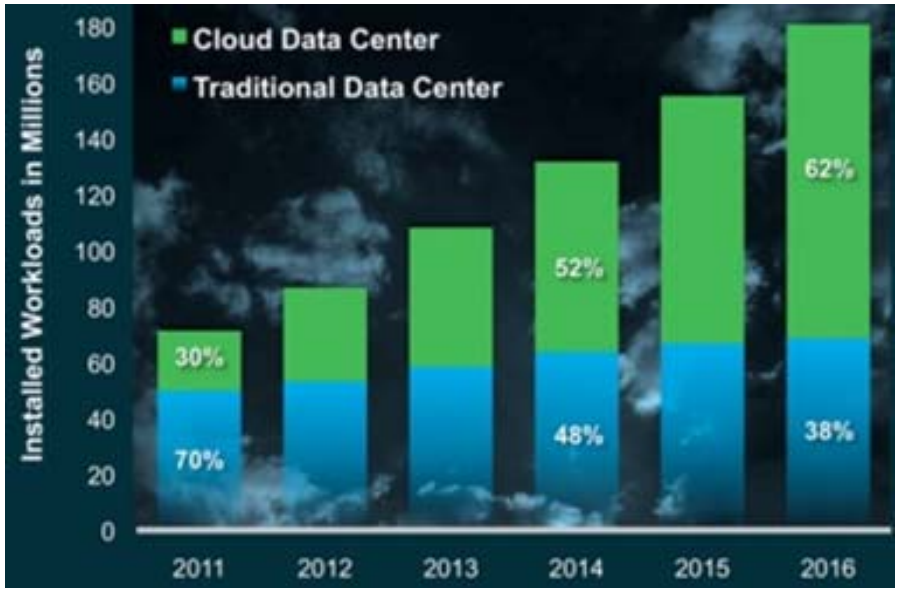
	Hypervisor based virtualization	TZ-AVP
Pros	• Available for most processor • High Flexibility (All virtualized functions can be implemented by software)	• Small size of V-Monitor • Easy to support real-time guest OS
Cons	• Larger size of hypervisor than TZ-AVP • Need to hypervisor optimization for real-time support	• Available for only ARM processor that implements TrustZone • Maximum two Guest OS allowed

Storage

-SSD scheduler for VMs on cloud-

Cloud Data Center (CDC)

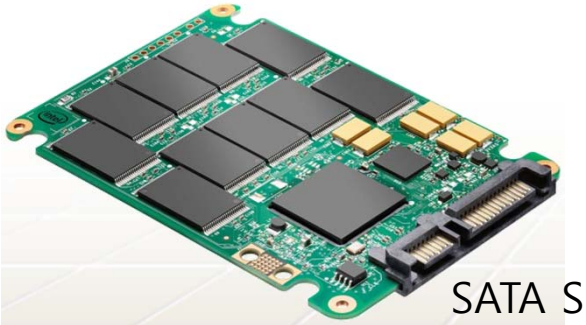
- **CDC based on virtualization**
  - **Advantages**
    - Low cost, high utilization, easy management
  - **Replacing traditional IDC (internet data center) rapidly**
    - 2011: 30% of entire workloads (0.7 zetta bytes = 0.7*10^21)
    - 2016: increase to 62%, 4.3 ZB



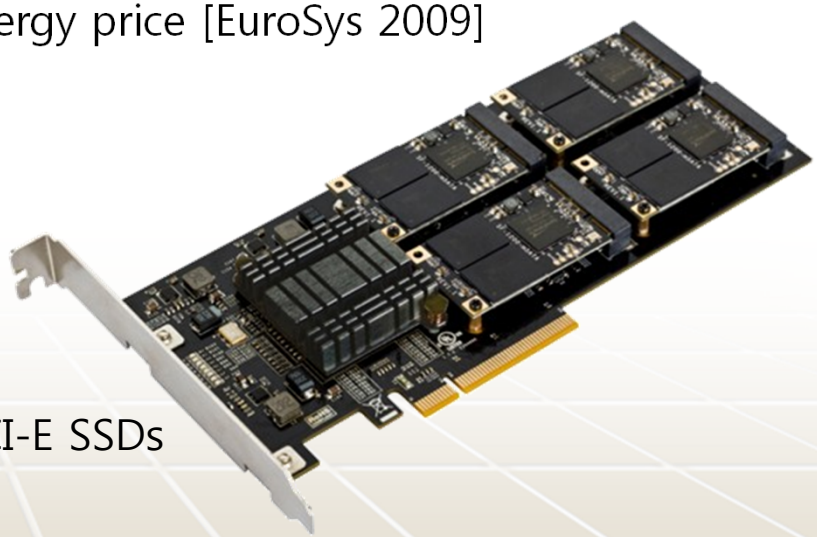
(Cisco Global Cloud Index, 2012)

Solid State Drives (SSDs)

- **Suitable for CDC servers**
  - **Advantages of SSDs**
    - Uniform and fast access time, high bandwidth, high IOPS (I/O per second)
    - Low heating
  - **High price, but low TCO (Total Cost of Ownership)**
    - High price is compensated by low energy price [EuroSys 2009]



SATA SSDs



PCI-E SSDs

SSDs Market Growth

News

AOL installs 50TB of SSD; boosts DB performance by 4X

SSD SAN installed to eliminate I/O bottlenecks caused by back-end storage

By Lucas Mearian

October 14, 2010 07:02 AM ET

5 Comments 좋아요 98 +1 0

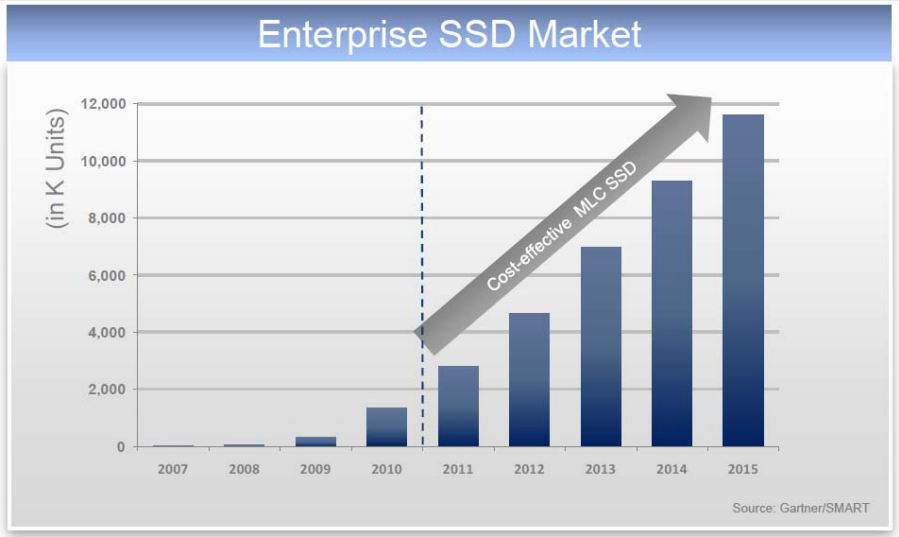
* Computer World, 2010

Projections – Enterprise SSDs – Revenues to SSD Manufacturer

Year	Revenue US$ (Millions)	Growth Rate %
2011	1,033	Base
2012	1,580	53
2013	2,386	51
2014	3,579	50
2015	5,368	50

* Storage strategies NOW

* Gartner/SMART



Market Growth Fueled by Cost-effective Enterprise Class SSD

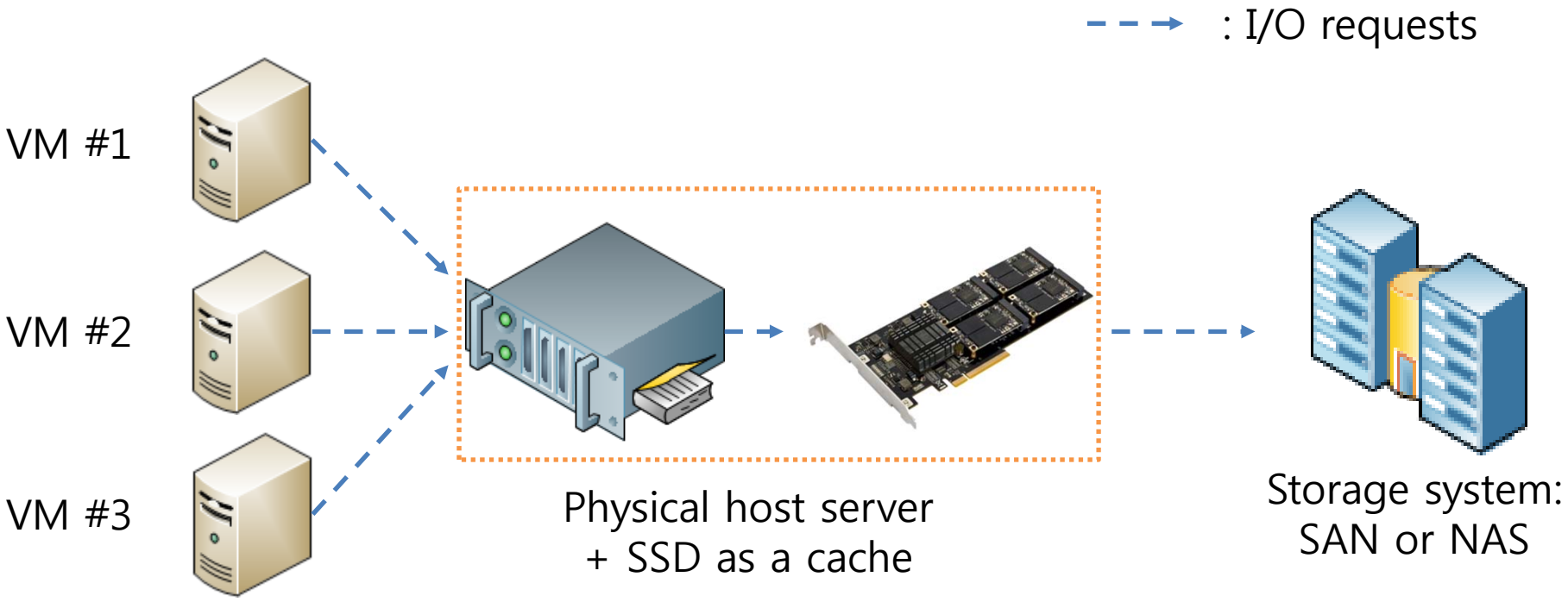
- **BIG IT industries adopt SSDs in their CDCs**
  - **AOL, Amazon, Google, and etc.**
- **Rapidly growing in market**
  - **Growth rates will be more than 50% since 2011**

Solid State Drives (SSDs) in CDC

- **Deployment architectures [IDC 2013]** * Jeff Janukowicz in NVM summit
  - **Hybrid Array**
    - Partially replacing HDDs as a cache or persistent storage
  - **Host based - Local to the server**
    - NetApp: Mercury [FAST 2011, MSST 2012]
    - EMC: XtremSW CACHE
    - FusionIO: ioCache
  - **All Flash Array - SSD only array**
    - Emerging architecture

Solid State Drives (SSDs) in CDC

- **Host based architecture: a host-side cache**
  - **Accelerates I/O processing in a host server**



Goals in cloud storage system

- **Goal 1: Exploiting SSDs to obtain the maximum performance**
- **Goal 2: QoS guarantees for several clients**
  - **QoS in CDC is defined as a service level agreement (SLA)**
    - A service contract where a service is formally defined
    - A technical definition of service in terms of throughput, response time, or similar measurable details
  - **SLAs with different metrics should be satisfied together**
    - Client A: Bandwidth more than 100 MB/s
    - Client B: Response time within 50 ms
    - Client C: Request processing more than 1000 IOPS

Conflicting goals

- **How to achieving both goals together?**
  - **Should provides QoS functionalities**
  - **Should utilizes SSD in maximum**

- **Need a new approach**
  - **I/O scheduler in hypervisor**
    - Directly manages a SSD
    - Arbitrate SSD usages among VMs

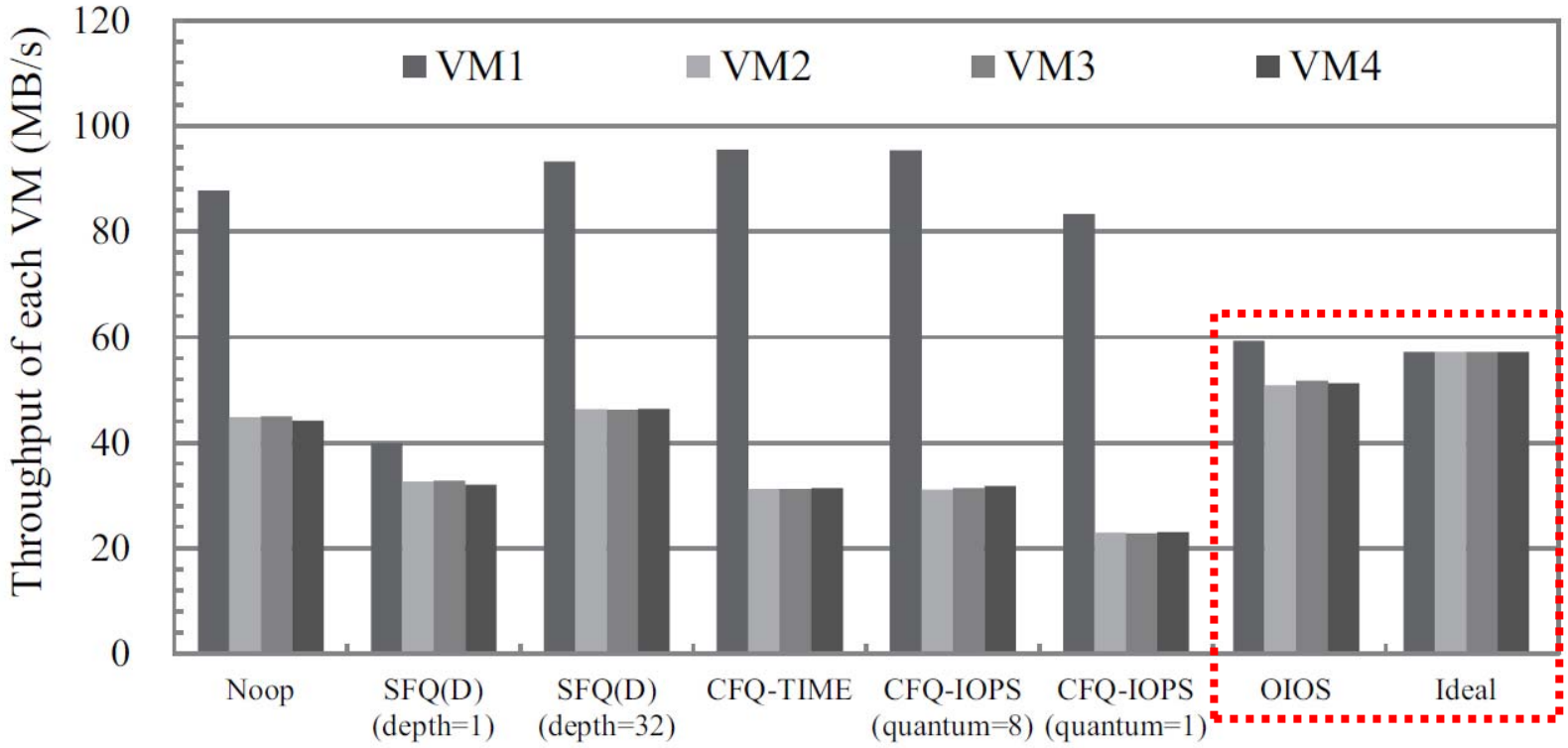
Related works

- **I/O scheduler on a hypervisor of host server**
  - **Roles**
    - Scheduling I/O requests from VMs in a host machine
  - **Existing schedulers**
    - NOOP: Simple FIFO scheduler. No merge or sort.
    - CFQ: Completely Fair Queuing Scheduler (Linux default)
      - Time-based : measurement of fairness is time
      - IOPS-based : measurement of fairness is IOPS
    - SFQ: Start-time Fair Queuing

Problems in existing I/O schedulers

- **Simple experiment to evaluate existing I/O schedulers**
  - **Question: "Do they achieve performance and QoS together?"**
  - **Experiment**
    - 4 VMs runs on 1 physical host with a SSD
    - VM1 tries to use SSD twice more than the others
    - QoS goal : complete fair sharing of SSD
  - **I/O schedulers**
    - Noop, CFQ-Time, CFQ-IOPS
    - SFQ(1) : SFQ with queue-depth 1. QoS-centric configuration
    - SFQ(32) : SFQ with queue-depth 32. Performance-centric configuration
    - OIOS: Our solution
    - Ideal case: a fair-share with maximum performance

Problems in existing I/O schedulers

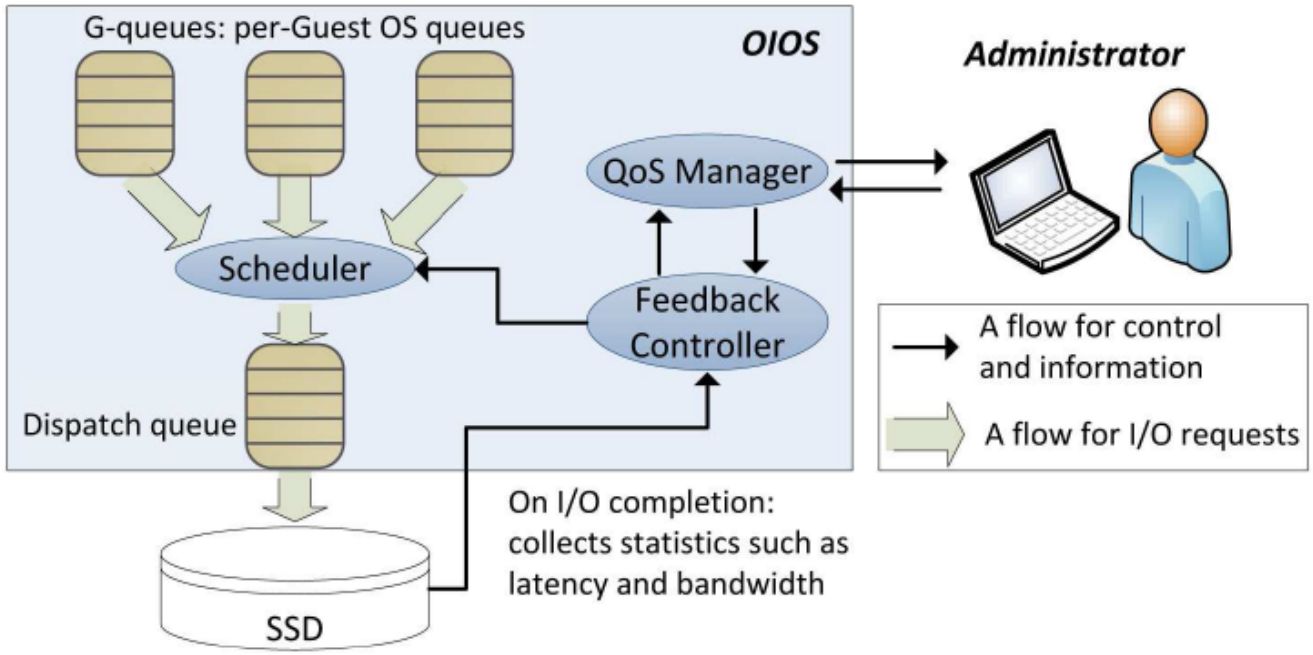


- **Existing schedulers fail to satisfy the performance and QoS together**
  - Performance only: SFQ(32), CFQ, NOOP
  - QoS only: SFQ(1)
- **OIOS shows fairly similar result with the ideal case**

Our solution: OIOS

- **The Opportunistic I/O Scheduler**
  - **QoS supporting**
    - Provides SSD sharing among VMs based on client SLA
    - QoS functionalities: Reservation, Limitation, Weight-based sharing
  - **Multi-metric supporting**
    - Clients can express their requirements based on multiple metrics
    - Supports 4 metrics: Bandwidth, latency, IOPS, and utilization
  - **No dependency for I/O workloads**
    - No need to examine I/O workload preliminary

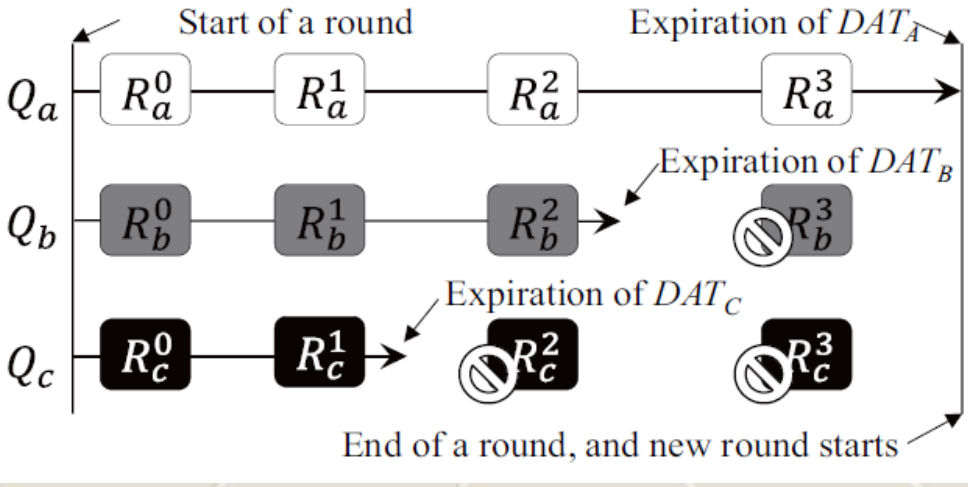
Architecture



- **Administrator**
  - Setups QoS requirements of VMs by QoS manager
- **Feedback controller**
  - Observes current statistics on SSD and notifies them to scheduling module
- **Scheduler**
  - Adjusts the SSD sharing parameters among VMs

Main idea

- **DAT scheduling (differentiated anticipation technique)**
  - **Performance differentiation mechanism**
    - Round-robin based scheduling
    - In a round, VMs gets different opportunities to dispatch their I/O requests to SSDs according to their weights
  - **Non-work conserving: OIOS fully exploits the SSD**
    - It always dispatches the I/O requests whenever they exists



Main idea

- **Feedback control for providing QoS**
  - **Collect the I/O statistics of each VM for several rounds**
    - Using different metrics: Bandwidth, latency, IOPS, and utilization
  - **Comparing the QoS achievement with QoS goal**
    - Less than QoS goal? Or more than QoS goal?
  - **According to the comparison, regulates DAT of each VM**
    - Regulate the opportunity of each VM
    - E.g. If a VM uses the shared SSD less than QoS goal, DAT will be enlarged to give more opportunity to use

Evaluation

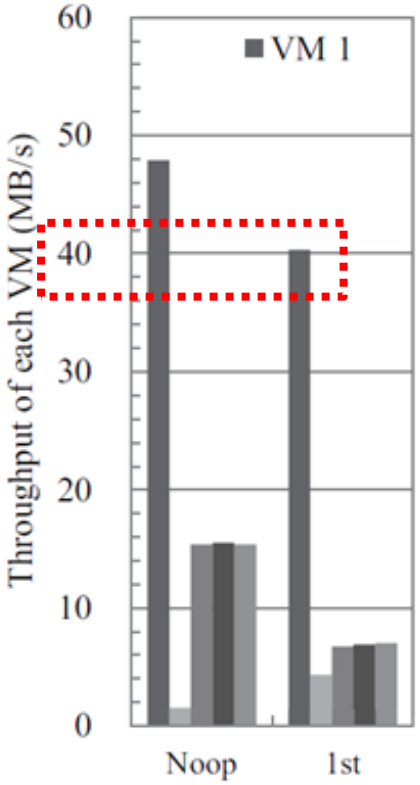
- **Setup**
  - **5 VMs runs on 1 physical host with a SSD**
  - **Different roles and workloads of VMs**

	VM1	VM2	VM3	VM4	VM5
Role	File server	DB server	Web server		
Workload	Filebench	Sysbench	Postmark		

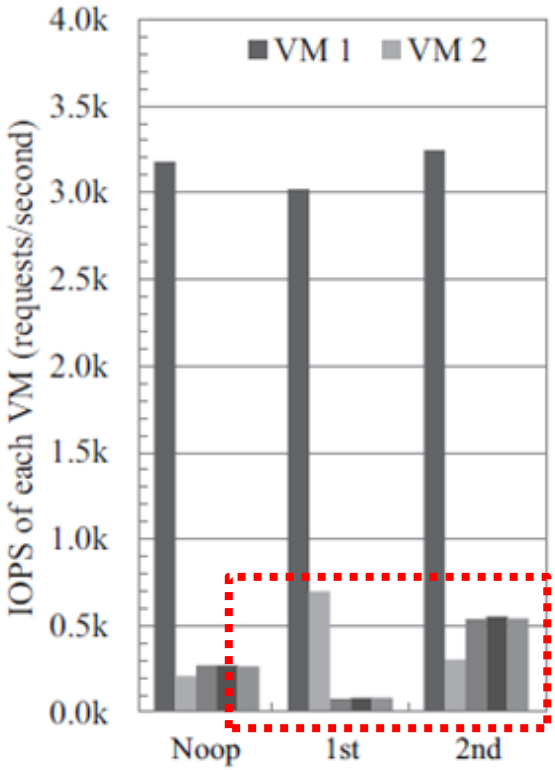
- **Scenarios**
  - **Accumulative QoS requirements for each scenario**
    - 1st : A reservation of 40 MB/s for VM1
    - 2nd : A limitation of 300 requests/s for VM2
    - 3rd : A reservation of 5 milliseconds for VM3 (latency less than 5 ms)
- **Noop scheduler is used for comparison**

Results

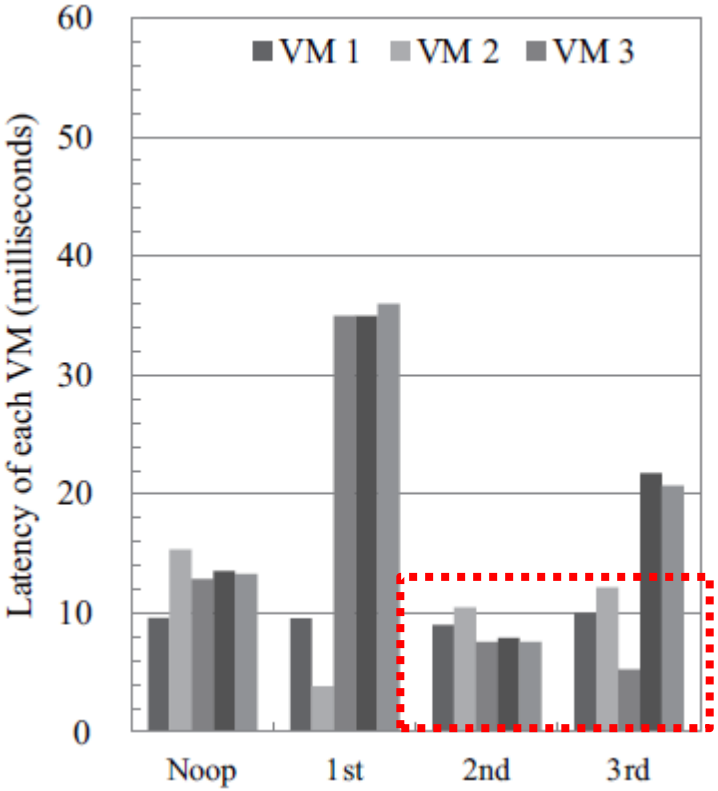
- 1st : A reservation of 40 MB/s for VM1
- 2nd : A limitation of 300 requests/s for VM2
- 3rd : A reservation of 5 milliseconds for VM3 (latency less than 5 ms)



(a) 1st scenario



(b) 2nd scenario



(c) 3rd scenario

Conclusion

- Storage virtualization focuses on SSDs
  - Virtualization enlarges CDC in data centers
  - SSDs are emerging devices in CDC
- The challenge is the QoS supporting with SSD
  - To guarantee different SLAs of clients
- OIOS: only solution for a host-based SSD architecture
  - I/O scheduler that supports QoS and utilizes SSD in maximum
  - Our evaluation demonstrates that OIOS satisfy different QoS goals with different metrics

Network

-New virtual router: Xebra-

Motivation

- **Future Internet**
  - **Requirements**
    - Various network protocols should coexist in Future Internet
    - Example: IPv4, content-centric network (CCN), cyber-physical system (CPS), routing protocol "plug-in"
  - **Key challenge**
    - How to isolate different networks

- **Network Virtualization is an alternative**
  - **Virtualizing network is a new approach for Future Internet**
    - Allowing multiple virtual networks on a physical network
    - Isolating virtual networks in router

Goals of Network Virtualization

- **Support virtual networks**
  - **Each network can have a different routing protocol**
  - **Network isolation**
    - Virtual networks(VNs) should not interfere each other
    - e.g. VN1 for IPv4 must not access packets of VN2 for CCN
- **Provide performance isolation**
  - **Should be able to control bandwidth allocated to virtual networks**
  - **Control CPU allocation as well**
- **Achieve performance comparable to what is in the market**

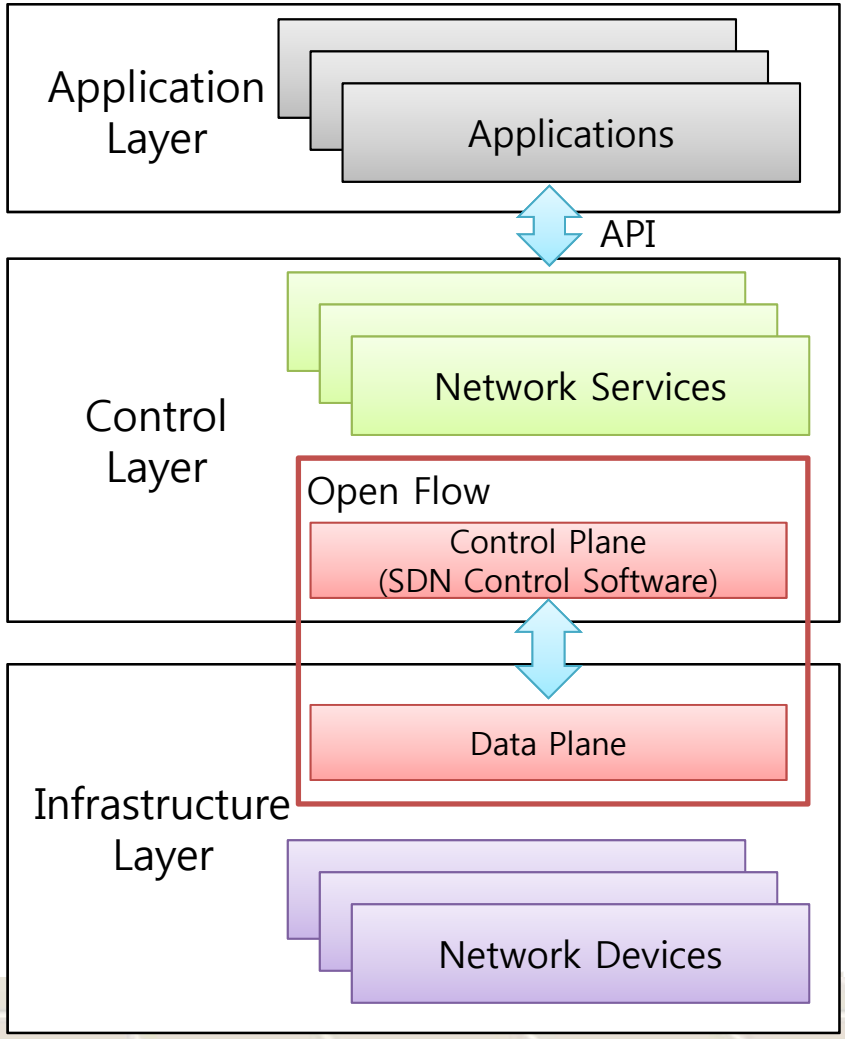
Approach for Network Virtualization

SDN (Software Defined Networking)

- **Network that is controlled by Software**
  - Implement network's functions and characteristics by software
  - Support centralized control by software
  - Separate the control of networks from the physical network
- **Advantage**
  - Protocols and services are upgradable at anytime
  - Business opportunity can be realized quickly

Architecture of SDN

- **Application Layer**
  - **Applications and services**
  - **Example : Video Streaming, Cyber Physical Systems, Cloud Robot**
- **Control Layer**
  - **Components to control entities in infrastructure layer**
    - Located on the machine isolated from data plane
    - Control and manage the network with the global view for the entire network's state
- **Infrastructure Layer**
  - **Hardware components for forwarding packets**
    - L2/L3 switching box for data transmission
    - Implemented by switches and routers



OpenFlow Overview

- **Well-known platform to implement SDN**
- **Separate the control plane and data plane**
  - **Support a communication protocol between the control plane and data plane**
  - **Control plane managed by controller and data plane is operated at switch**
- **Remove the dependency by device vendors**
  - **Software on an external machine decides packet's path regardless of device vendors**
  - **Support programmable open protocols which are independent of vendors**

OpenFlow vs. Classical Router

- **Classical router (CR)**
  - **Data plane and control plane coexist on the same device**
  - **Difficult and expensive to deploy new protocols**
    - Because classical router is black box

- **OpenFlow**
  - **Separate data plane and control plane**
    - Data plane reside on the switch and control plane moved to a separate controller
  - **By this method OpenFlow can slice traffic with controller and reduce network management cost**

SDN Trends

- Vendors develop their own proprietary versions of SDN and OpenFlow
  - Delivering hybrid and centralized control in vertically integrated fashion
  - Cisco, HP, IBM and Dell
- SDN and OpenFlow development also tie into OpenStack
  - Automatically provision network resources through SDN controller
  - Controlling OpenFlow enabled switches
- VMware vCloud
  - Cloud management platforms (CMP)
  - Nicira
  - Create the software defined datacenter that is fully automated

New Architecture : Xebra

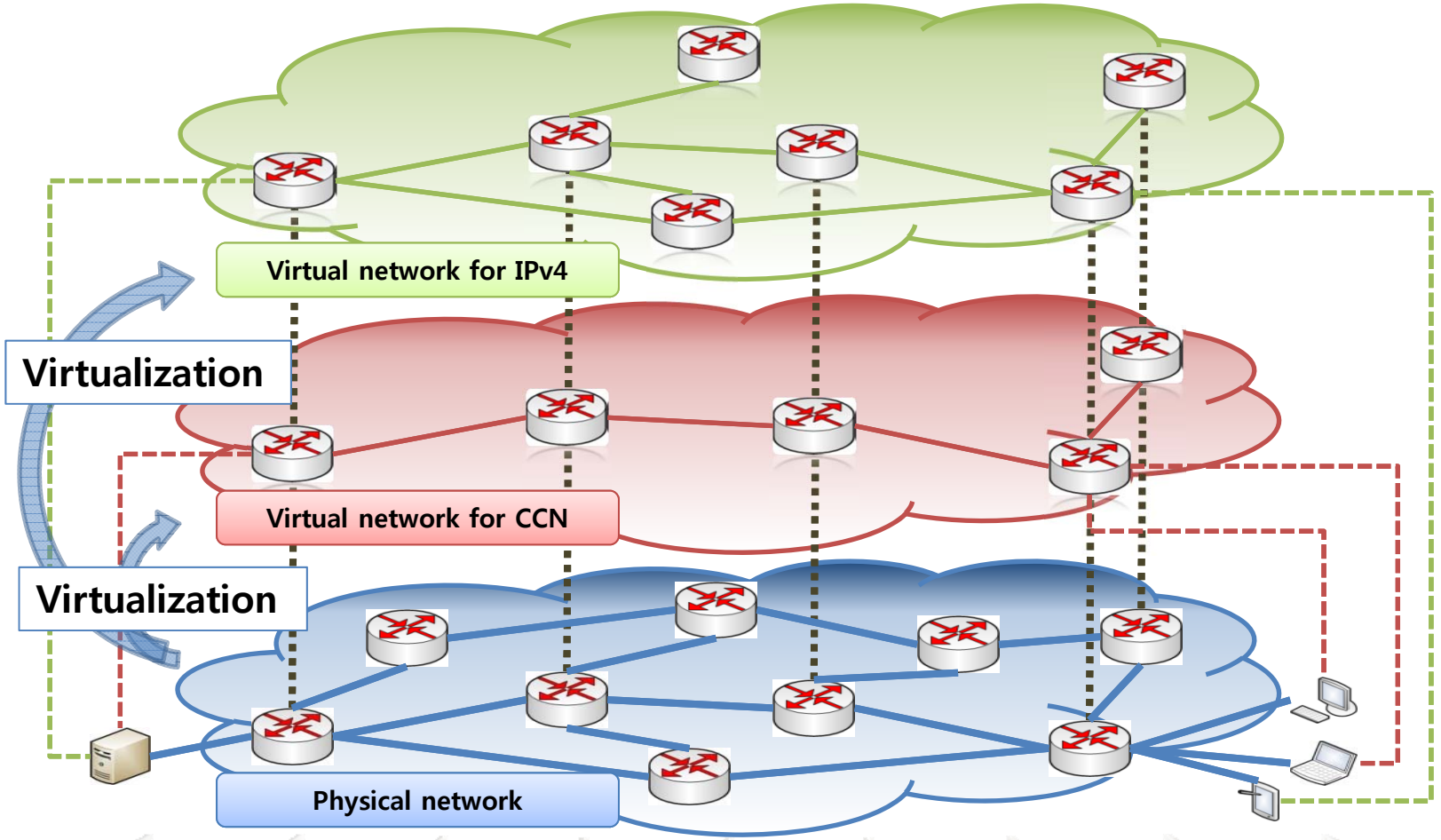
Xebra Overview

- **Xen based virtual router**
- **Apply virtualization to support virtual networks**
  - **VM(Virtual Machine) is an unit of isolation between virtual networks**
  - **Link isolation becomes a necessity**
  - **Together, network isolation is guaranteed**
    - Each VM contains both data plane and control plane
- **Software only design with commodity hardware**
- **Utilize features in commodity hardware**
  - **PCI SR-IOV with Intel VT architecture**

Goals of Xebra

- **Various virtual networks**
  - **Support several virtual networks simultaneously**
  - **Each network can have a different routing protocol**
  - **Need network isolation**
    - Virtual networks(VNs) should not interfere each other
    - e.g. VN1 for ipv4 must not access packets of VN2 for CCN
- **Provide performance isolation**
  - **Should be able to control bandwidth allocated to virtual networks**
  - **Control CPU allocation as well**
- **Achieve High performance**
  - **Comparable to what is in the market**

Eventually…


Virtual network for IPv4
Virtualization
Virtual network for CCN
Virtualization
Physical network

Xebra Status

- **Fully operational as router**
- **Redesigned and implemented network stack of Linux**
- **Built with commodity hardware**
  - **Intel XEON X5650 (2.67GHz, 6-cores) * 2**
  - **12GB physical memory**
  - **Intel 82599 NIC (10Gbps with SR-IOV support)**
- **Software environment**
  - **Xen 4.0.2**
  - **Guest OS Ubuntu 10.04 LTS with Paravirtualization (Kernel ver 2.6.37.1)**
  - **2 VCPUs**
  - **2GB memory**
  - **NIC**
    - Igbvf (Intel 82599 VF) 1.1.3 for SR-IOV
    - E1000 (Xen PV NIC model)
  - **Benchmark : pktgen**

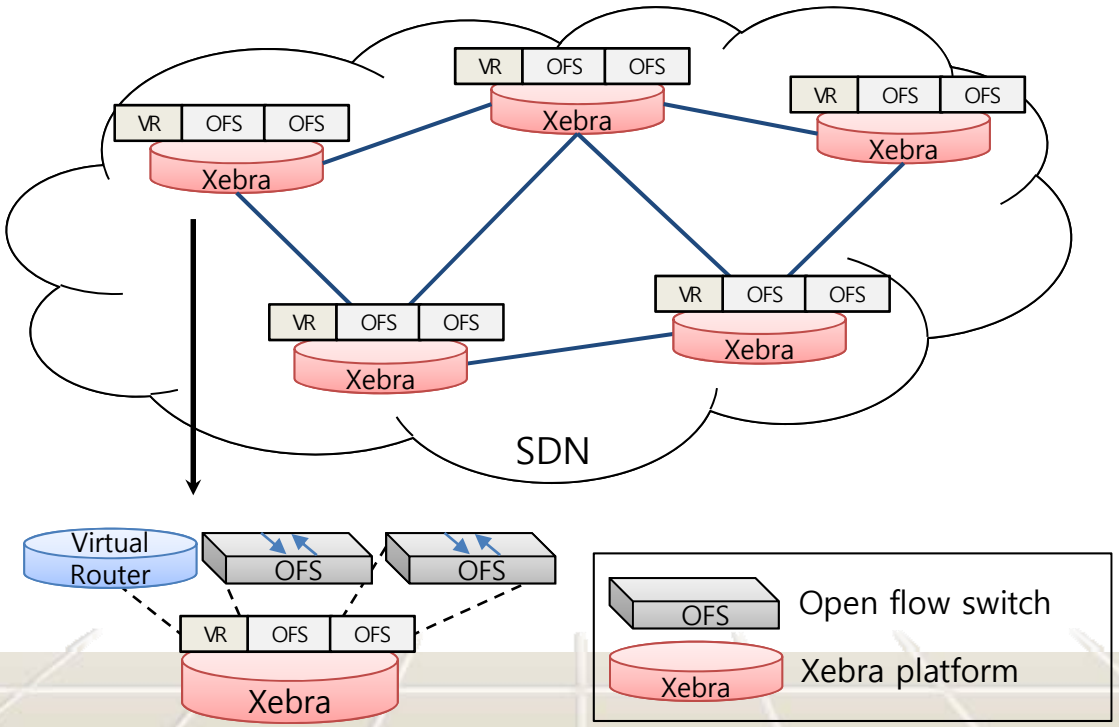
Performance Comparison

	Trellis	PEARL	VRouter	Xebra
Feature	Focused on flexibility, isolation	Designed for high performance, flexibility, and isolation using multicore CPU and netFPGA	Virtual router based on Xen	High performance router based on Xen
Virtualizaton Platform	Vserver	Linux-Based LXC	Xen	Xen
VN ID	MAC address for virtual network	Vlan tag	NA	UDP port number
Link Isolation (tunneling)	EGRE (Ethernet over GRE)	NA	NA	Mac-in-UDP
Performance per flow	1Gbps	3.9Gbps	3.6Gbps	38Gbps
Features	- Decapsulated in dom 0	- Use NetFPGA for routing table lookup - Support performance isolation through 2 levels of priority	- Separate control planes except forwarding plane - No isolation	- Control and data plane in VM - Address resolution by vARP

Achieves almost 9.7 times more performance than PEARL!

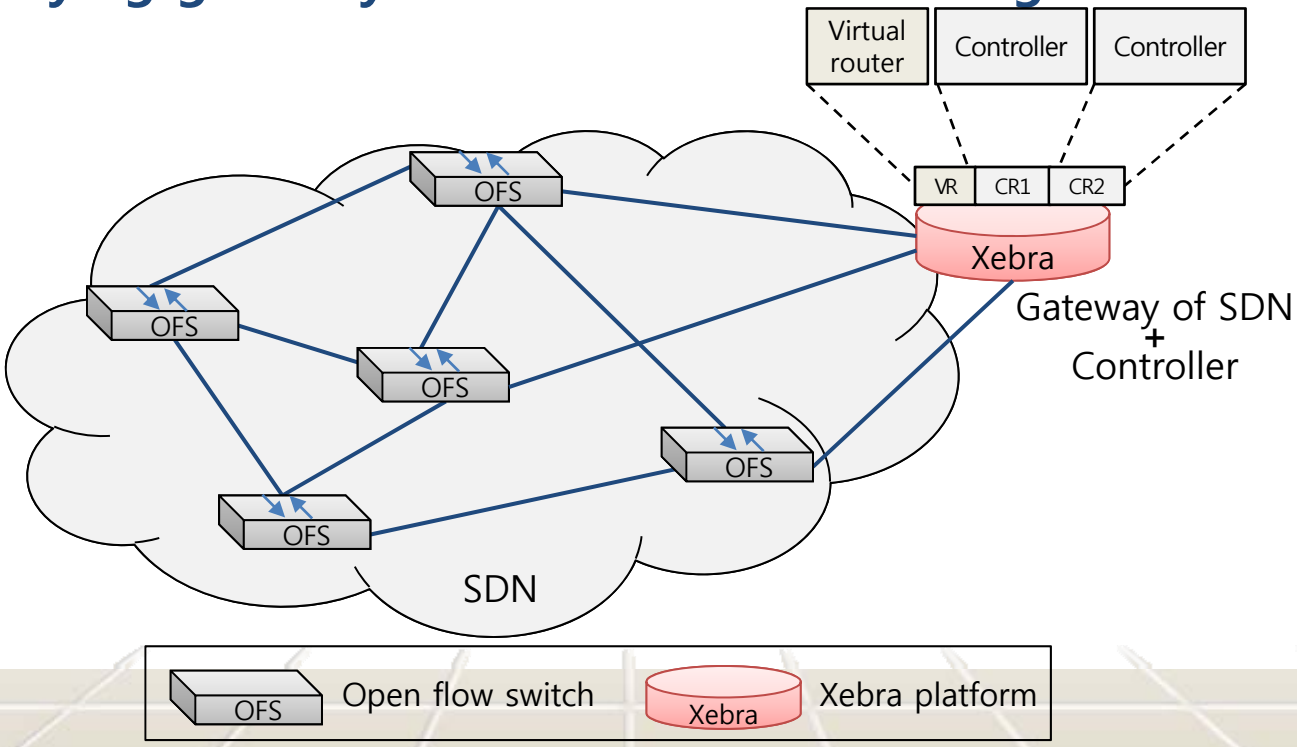
Xebra with SDN : Switch

- **Xebra platform as a OpenFlow switch**
  - **Xebra supports OpenFlow switches and virtual router simultaneously on a single machine**



Xebra with SDN : Controller

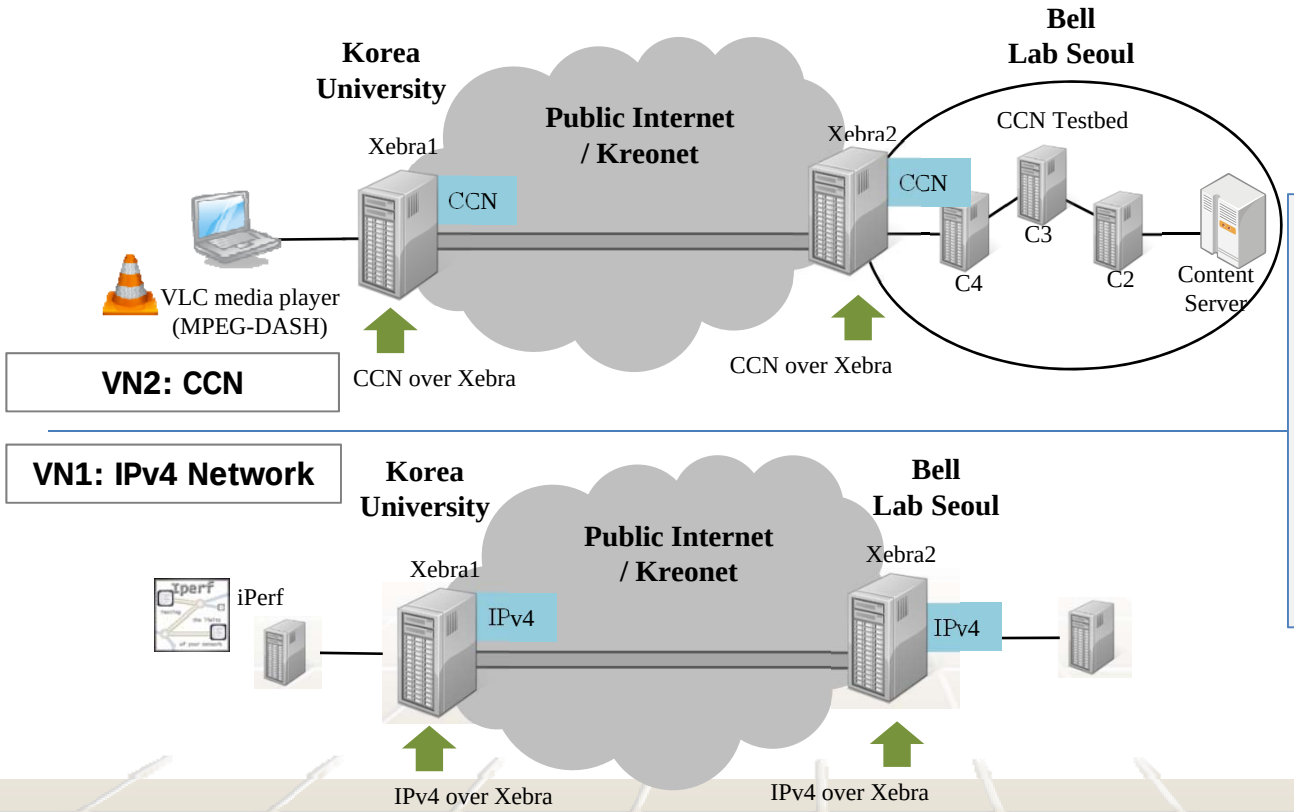
- **Xebra platform as a OpenFlow Controller**
  - **Multiple controllers on a single machine**
    - Substitute for FlowVisor
  - **Gateway for the network which consists of switches**
  - **Unifying gateway and controllers on a single machine**



Evaluation

CCN and IPv4 on Xebra

- **Xebra can support CCN and IPv4 concurrently**
  - **By isolating two virtual networks**
- **VN1 is IPv4 network and VN2 is CCN Network**



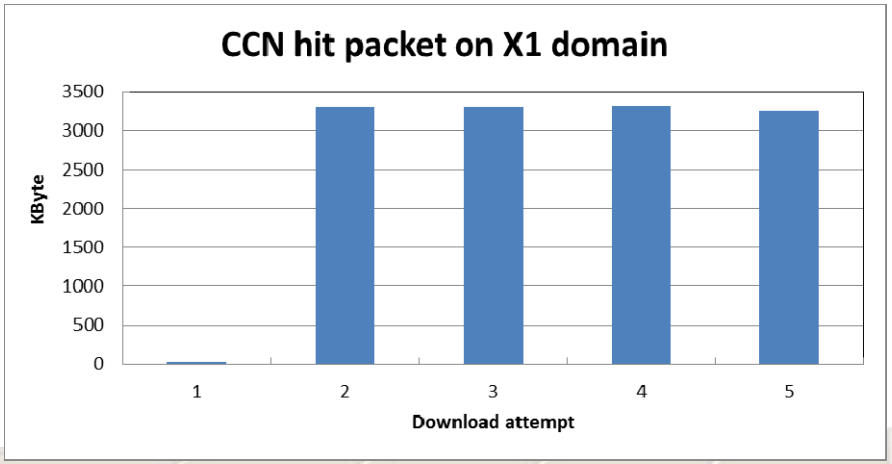
- Xebra1 and Xebra2 are Xebra routers
- C2, C3, and C4 are CCN nodes with data caching

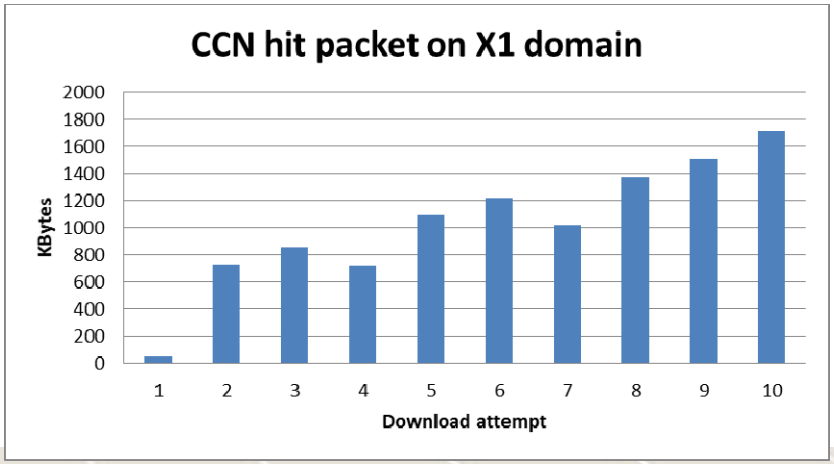
Cache effect by CCN

- **Traffic reduced by CCN cache**
  - **All traffics cached after first attempt**

- **Kreonet only**
  - **All CCN traffics hit by X1 CCN domain cache**
    (Download attempt 2~5)
    - There are no traffics between X1 and X3.

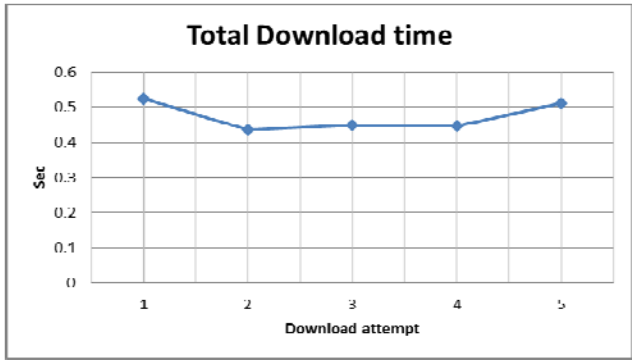
- **Public Internet**
  - **Hit CCN traffics changed by download attempts**
    - Selected bitrate is changed because of dynamic bandwidth states.





Download time by CCN

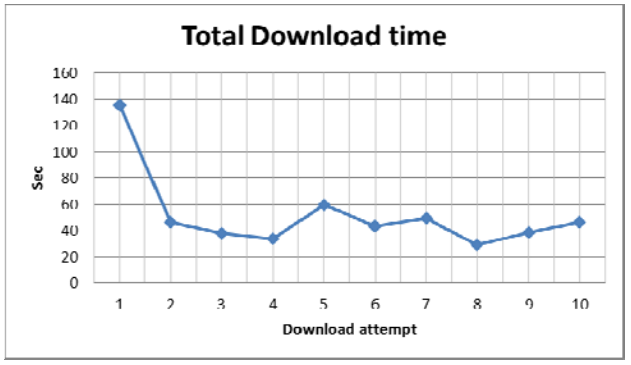
- **Kreonet only**
  - **Few changes of download time**
    - Wide and stable network bandwidth



  - **Selected bitrate**

	Download attempt				
chunk number	1	2	3	4	5
8	351	351	351	351	351
9	351	351	351	351	351
10	351	351	351	351	351
11	351	351	351	351	351
12	351	351	351	351	351
13	351	351	351	351	351
14	351	351	351	351	351
15	351	351	351	351	351
16	351	351	351	351	351
17	351	351	351	351	351
18	351	351	351	351	351
19	351	351	351	351	351
20	351	351	351	351	351
21	351	351	351	351	351
22	351	351	351	351	351
23	351	351	351	351	351
24	351	351	351	351	351
25	351	351	351	351	351
26	351	351	351	351	351

- **Public Internet**
  - **Huge changes of download time**
    - Dynamic network bandwidth



  - **Selected bitrate**

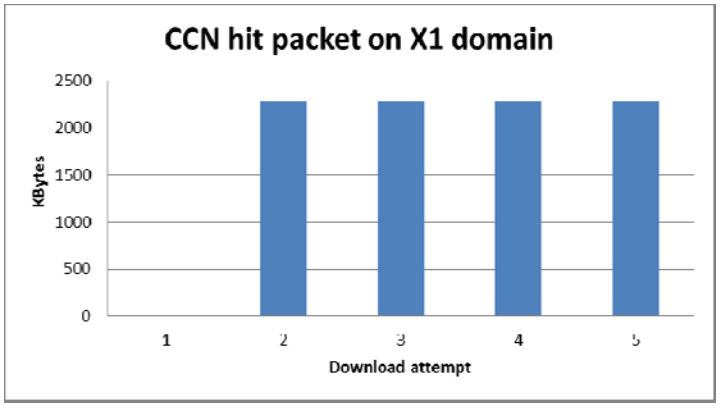
	Download attempt									
Chunk number	1	2	3	4	5	6	7	8	9	10
8	101	351	351	351	351	351	351	351	351	351
9	101	101	351	351	351	351	351	351	351	351
10	101	101	101	351	351	351	351	351	351	351
11	101	101	101	101	351	351	351	351	351	351
12	101	101	101	101	351	351	351	351	351	351
13	101	101	101	101	101	351	351	351	351	351
14	101	101	101	101	101	101	351	351	351	351
15	101	101	101	101	101	101	351	351	351	351
16	101	101	101	101	351	101	351	351	351	351
17	101	101	101	201	101	101	351	351	351	351
18	101	101	101	201	101	201	101	351	351	351
19	101	101	101	101	101	201	101	351	351	351
20	101	101	101	101	101	101	101	351	351	351
21	101	101	101	101	101	101	101	101	351	351
22	101	101	101	201	101	101	101	101	351	351
23	101	101	101	101	101	101	101	101	101	351
24	101	101	101	101	101	101	101	101	101	351
25	101	101	101	101	101	101	201	101	101	101
26	101	101	201	101	101	101	101	101	101	101

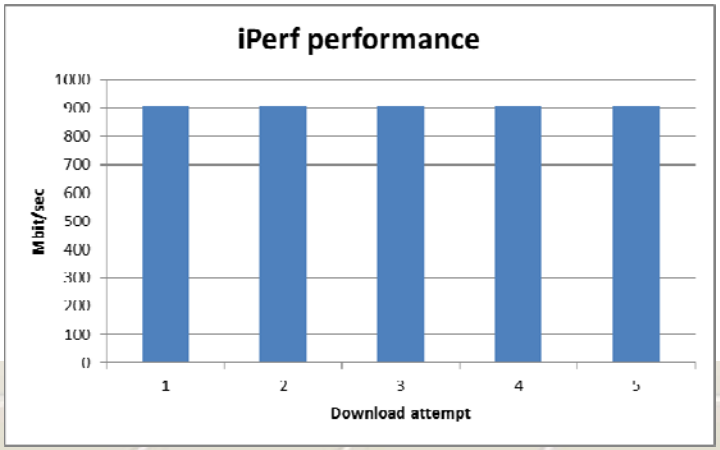
Not cached in X1

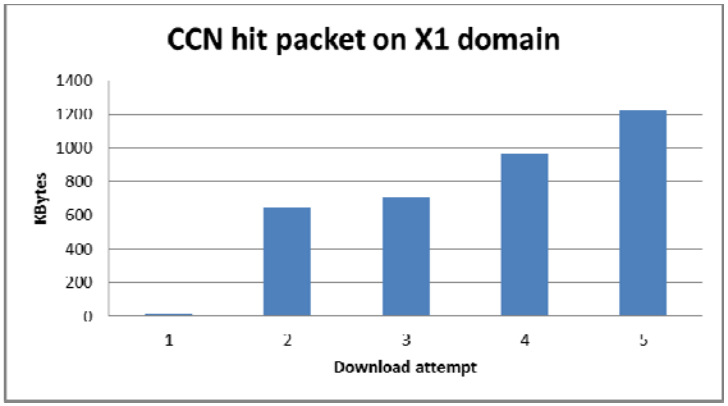
Cached in X1

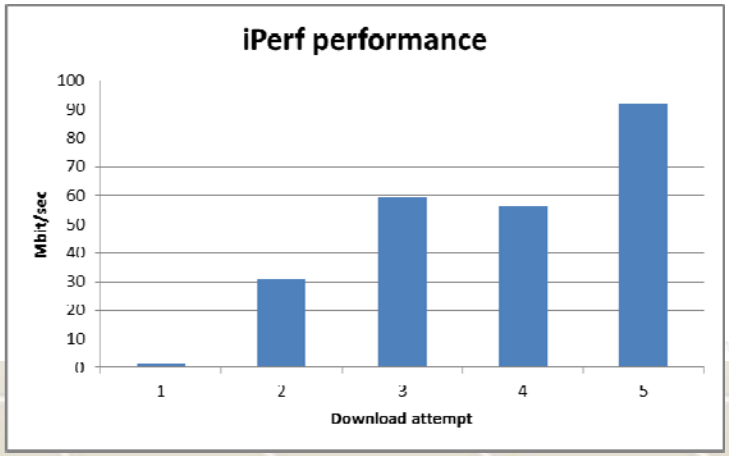
Performance by IPv4 & CCN

- **Kreonet only**
  - CCN and IPv4 traffics do not interfere because of large bandwidth
- **Public Internet**
  - More cache hit of CCN packets uses less network bandwidth
  - This causes more IPv4 bandwidth









Conclusion

- **Requirements of network virtualization**
  - **Support multiple virtual networks**
  - **Isolate virtual networks**
  - **Need more research**

- **Xebra**
  - **Router virtualization with commodity hardware**
    - Low cost, flexibility
    - High performance achieved
    - Scalable test bed running on the existing IP network
      - CCN is working together with IPv4

Future

Is there any future for virtualization ?

Q & A