Next-Generation Cloud/Big Data Infrastructure (and what we do @ POSTECH)

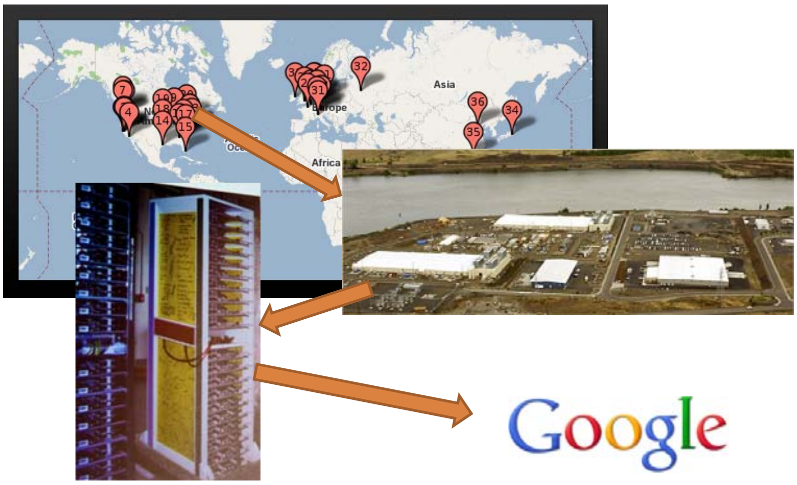
Jangwoo Kim

Oct 16, 2015

E-mail: jangwoo@postech.ac.kr

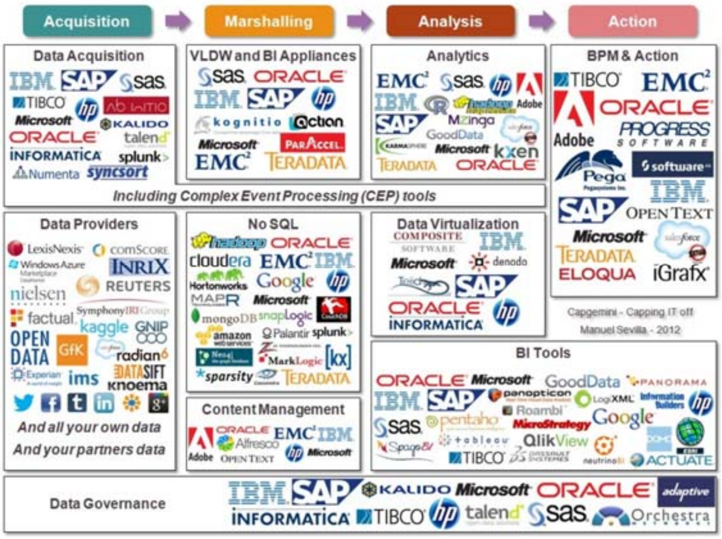
High Performance Computing Lab
Department of Computer Science & Engineering
Pohang University of Science and Technology (POSTECH)

Cloud companies run datacenters



Cloud IT company = “datacenter” company

Large companies process big data



Acquisition → Marshalling → Analysis → Action

Data Acquisition

VLDW and BI Appliances

Analytics

BPM & Action

Including Complex Event Processing (CEP) tools

Data Providers

No SQL

Data Virtualization

Capgemini - Capping IT off
Manuel Sevilla - 2012

BI Tools

And all your own data
And your partners data

Content Management

Data Governance

Smart companies make smart devices

Smart company = “smart device” company

“Smart devices + Cloud + Big Data”

a new computing engine?

Outline

- Introduction
- **Little more history**
- Issues in cost-effective cloud
- Cost-effective cloud @ POSTECH
- Summary

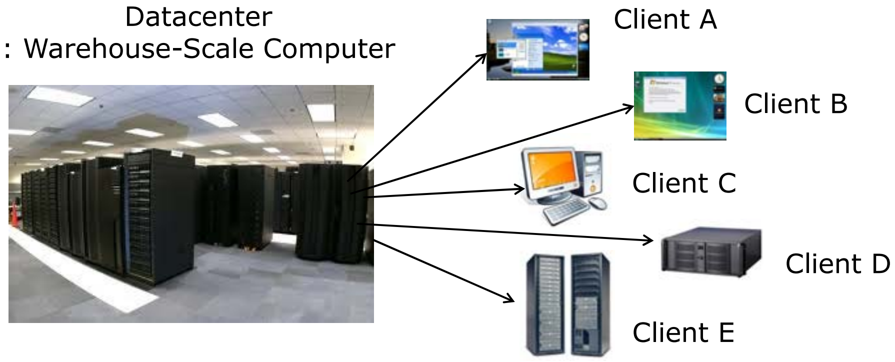
The “birth” of cloud computing

- **How did it start?**
  - Major IT companies (e.g., Google, Amazon, MS, etc.)
    - We have too many computers, but mostly idle computers
    - **How can we make more money?**
  - Rest of the world (e.g., anyone using computers)
    - We don’t want to maintain expensive computers.
    - **How can we reduce our costs?**

Let’s sell/buy computing as an on-line service!

Cloud computing: a simple picture

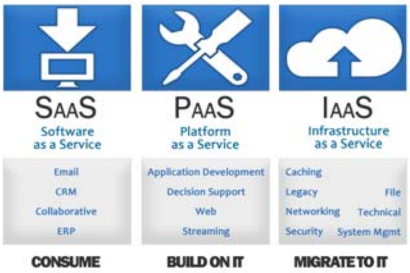
- **Datacenter provides SW/HW as a service**



Datacenter + Virtualization + Application
→ Cloud Computing

‘Three’ cloud service models

- **SaaS (Software as a Service)**
  - Applications, typically available via the browser e.g., Goggle Apps, MS Office365
- **PaaS (Platform as a Service)**
  - Application environment for building cloud apps e.g., Google App Engine, MS Azure
- **IaaS (Infrastructure as a Service)**
  - Providing utility-computing data center e.g., Amazon EC2, KT Ucloud



[source: Microsoft]

The “birth” of big data

- **How did it start?**
  - BIG, BIG, BIG data…
  - Existing things are not working any more
    - **Scalability**
      - o Can handle increasing amount of data?
    - **Fault tolerance**
      - o Can work with failed storage?
    - **Read & Write**
      - o Can access the data as we used to do?
    - **Processing data**
      - o Can work on the large data on many nodes?

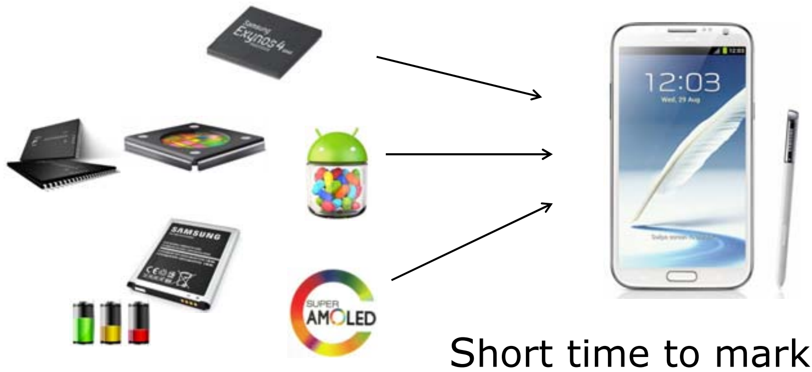
The “birth” of smart devices

- **We carry small, but powerful computers**
  - Mostly “standardized” components
    - ARM CPU, Moderate GPU, SSD, ..
    - Google Android, Apple iOS, Microsoft Windows 8, ..
    - HD camera, scripting tool, ..
    - App market (well, mostly games?), ..
    - **How to become a market leader?**
      (or how to differentiate your devices?)

Well, let’s pack all these things fast and nicely!

Smart device: a simple picture

- **Assemble standard components fast and nicely**

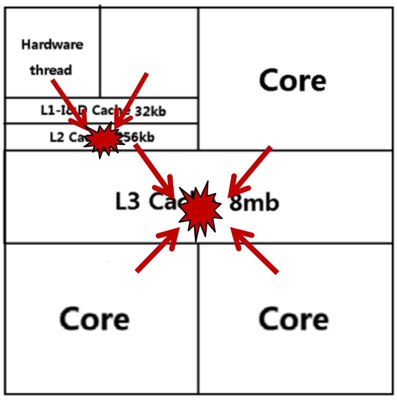


Short time to market!
High performance!
Long battery life!

Outline

- Introduction
- Little more history
- **Issues in cost-effective cloud**
  - **Performance**
  - Power
  - RAS
  - Analysis
  - Storage
  - Mobile
- Cost-effective cloud @ POSTECH
- Summary

Quality of Performance (1/2)



[Example 4-core CPU]

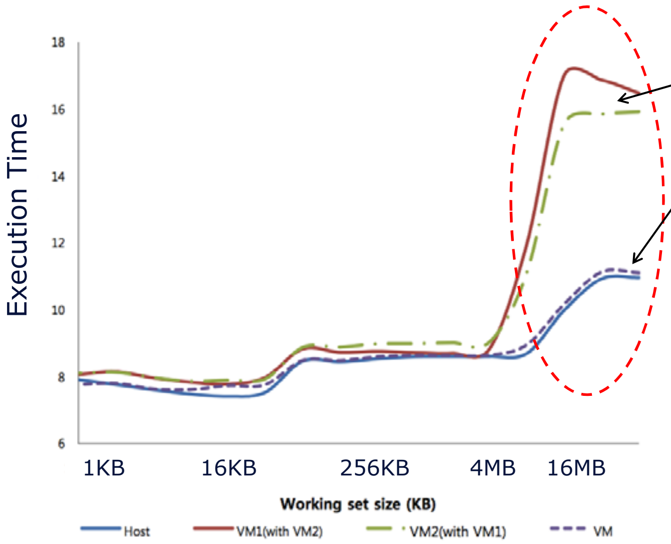
No true ‘**multi**’ thing!

Virtual machine is only ‘virtual’

> Resource contention exists
ALUs,
caches
I/O devices
Net bandwidth

> Difficult performance analysis

Quality of Performance (2/2)



Two programs on 2 VMs / 2 Cores

One program on 1 VM / 1 Core

Service you wanted
!=
Service you obtained

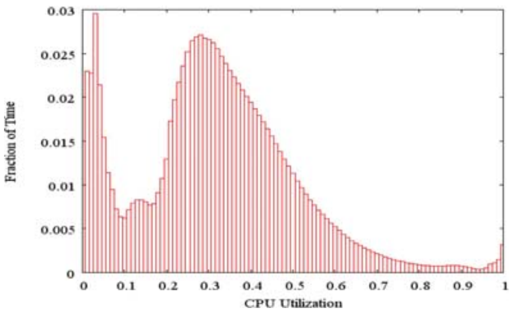
→ QoS failure

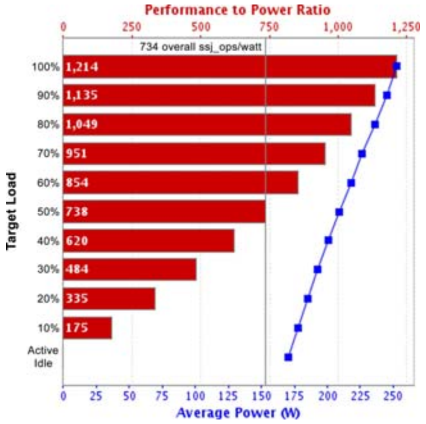
Outline

- Introduction
- Little more history
- **Issues in cost-effective cloud**
  - Performance
  - **Power**
  - RAS
  - Analysis
  - Storage
  - Mobile
- Cost-effective cloud @ POSTECH
- Summary

Bad news for power: no work

[Example Google server]





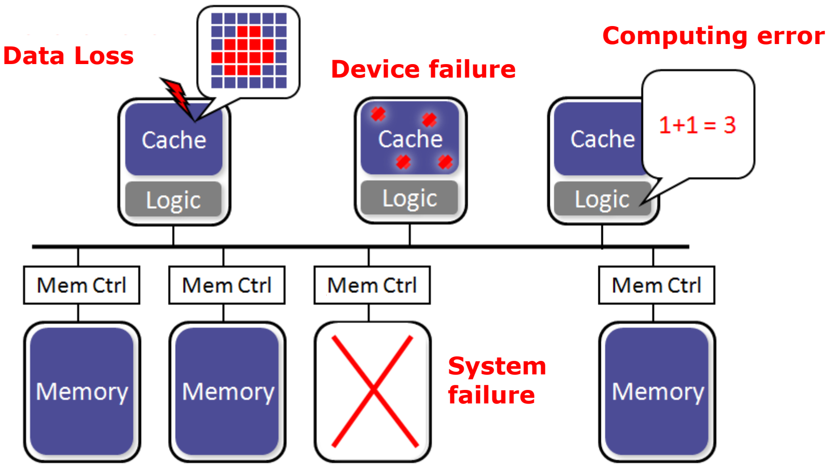
Datacenter is **usually idle** Still consumes **huge power**

We need 100% busy servers & 100% power-off servers for some periods

Outline

- Introduction
- Little more history
- **Issues in cost-effective cloud**
  - Performance
  - Power
  - **RAS**
  - Analysis
  - Storage
  - Mobile
- Cost-effective cloud @ POSTECH
- Summary

Systems do fail in field!



RAS is one of key SLA for large-scale clusters

Datacenter can fail anytime, anywhere!

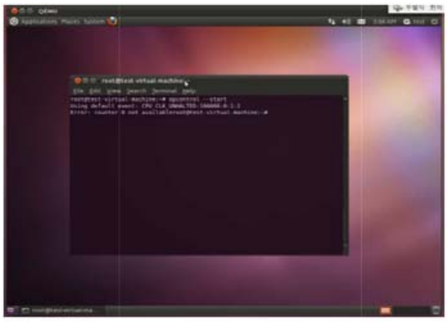
- **Failure rate of datacenter (case of soft error)**
  - Mean Time Between Failures (MTBF)
    - Average times between two failures per system.
  - For a single system,
    - **After applying** all the existing reliability techniques, we can achieve MTBF = 30 years = 10,000 days.
  - For a data center, however,
    - If we have 10K servers, our MTBF = 1 day.
    - → **At least one system in the center fails everyday.**

More redundancy?
Too expensive for a datacenter.

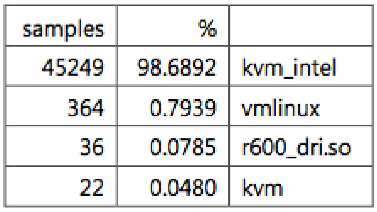
Outline

- Introduction
- Little more history
- **Issues in cost-effective cloud**
  - Performance
  - Power
  - RAS
  - **Analysis**
  - Storage
  - Mobile
- Cost-effective cloud @ POSTECH
- Summary

VM-level analysis is TOO LIMITED



```
root@test-virtual-machine:~# opcontrol --start
Using default event: CPU_CLK_UNHALTED:100000:0:1:1
Error: counter 0 not availableroot@test-virtual-machine:~#
```



samples	%	
45249	98.6892	kvm_intel
364	0.7939	vmlinux
36	0.0785	r600_dri.so
22	0.0480	kvm

Difficult to analyze the performance of apps on VM

Modeling & Simulation: what is it?

- **Modeling is the key of system R&D.**

Before making a real datacenter
or running an application

- Design exploration
- Design modeling
- Design evaluation
  - Performance
  - Power
  - Reliability
  - …
- Design feedback

Must be able to model & simulate datacenters

Timing simulation is TOO SLOW

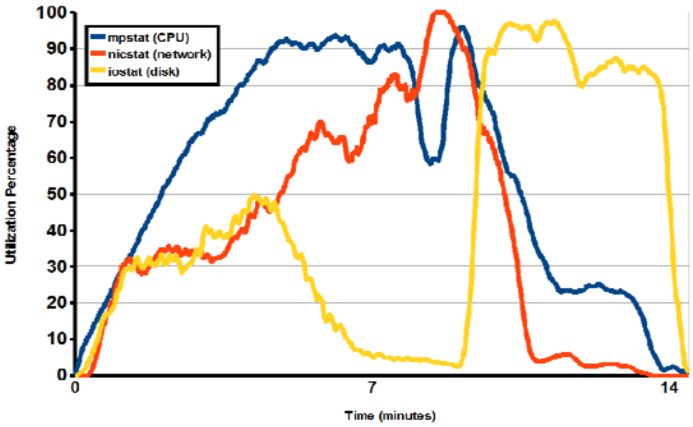
- **Typical simulation speed**
  - Speed granularity as Instruction Per Second (IPS)

▪ Real machine (e.g., Intel CPU)	:	1,000,000,000 IPS
▪ Same ISA/arch functional VM simulator	:	500,000,000 IPS
▪ Different ISA/arch functional VM simulator	:	1,000,000 IPS
▪ Timing simulator (e.g., Flexus)	:	1,000 IPS

  - 1 min on real machine → >1 year on timing simulator
  - However, real-world workloads require long-period benchmarking (e.g., hours for TPCC on database engine)

How to do cloud-level performance simulation?

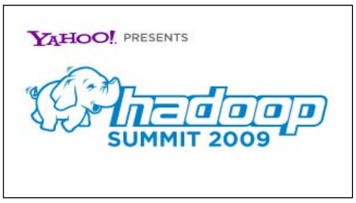
System-level analysis is TOO DIFFICULT



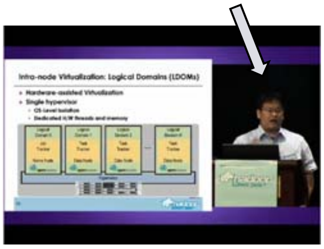
[150GB sort using 640-thread CPUs using **Hadoop** @ Sun Microsystems]

Various performance bottlenecks exist.

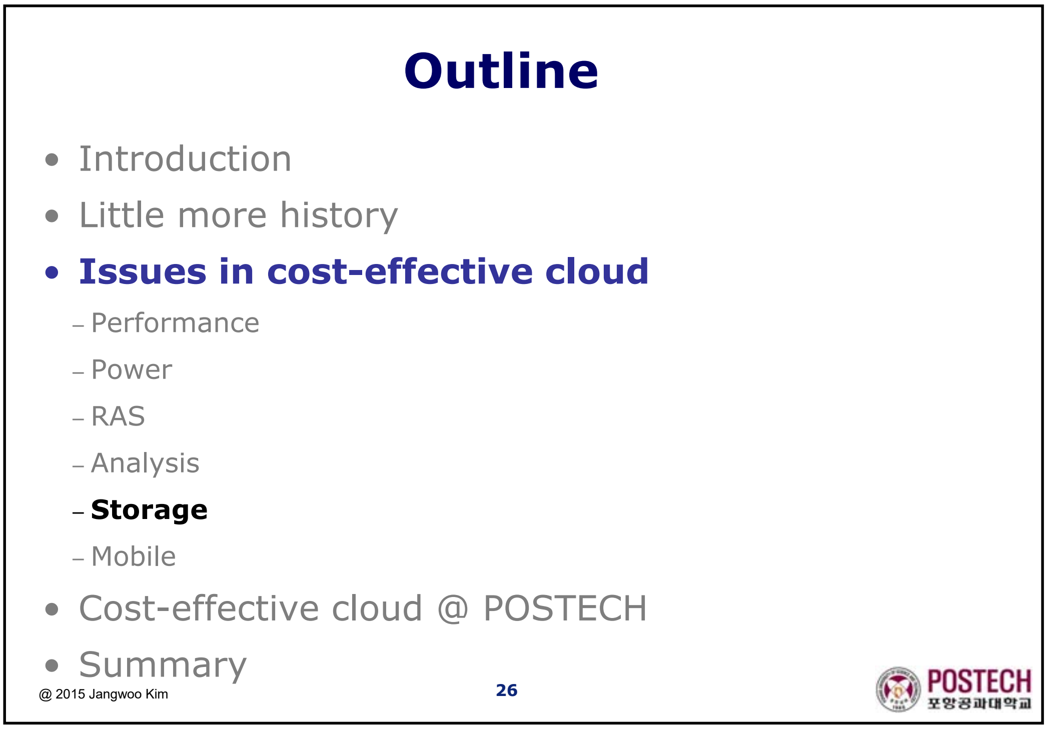
Performance analysis of Hadoop



Me!

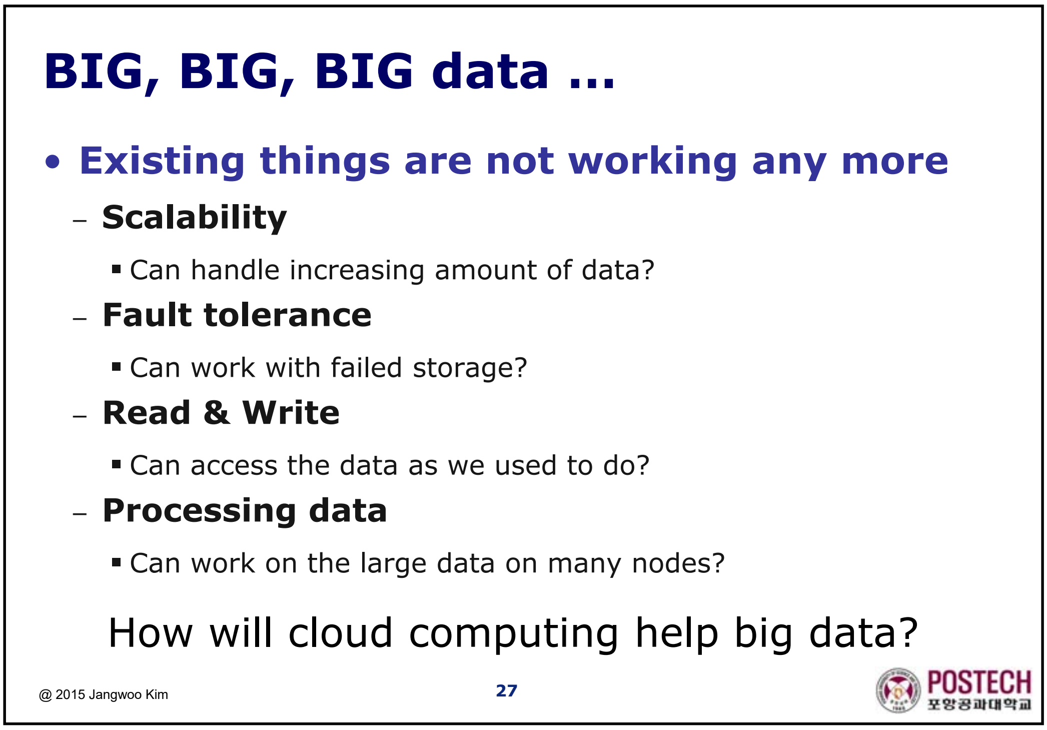

Intra-node Virtualization: Logical Domains (LDOMs)
Hardware-assisted Virtualization
Single hypervisor
OS-level isolation
Dedicated H/W threads and memory

700+ attendees looking for performance analysis method



Outline

- Introduction
- Little more history
- **Issues in cost-effective cloud**
  - Performance
  - Power
  - RAS
  - Analysis
  - **Storage**
  - Mobile
- Cost-effective cloud @ POSTECH
- Summary



BIG, BIG, BIG data …

- **Existing things are not working any more**
  - **Scalability**
    - Can handle increasing amount of data?
  - **Fault tolerance**
    - Can work with failed storage?
  - **Read & Write**
    - Can access the data as we used to do?
  - **Processing data**
    - Can work on the large data on many nodes?

How will cloud computing help big data?

Big Data: storage (i.e., file system)

- **Performance vs Replication**
  - Big data stored in many disk drives
    - Adding or removing disk drives → Horizontal elasticity
    - Separating control node and data objects (often unstructured)
      - meta node : know the physical locations of data
      - object node : store the data object
    - Must maintain replications → Slow performance
  - Solutions with different tradeoffs

    (e.g., how to manage meta node and object node, global view, ..)
    - File system: Google FS (GFS), Hadoop FS (HDFS), Amazon Dynamo
    - Cloud storage: Amazon S3, OpenStack Swift

Big Data: access (i.e., database)

- **SQL vs NoSQL**
  - Conventional RDBMS systems cannot be scaled

    without sacrificing its lock/log-based ACID

    (Atomicity, Consistency, Isolation and Durability)
  - Solutions supporting only parts of ACID

    (e.g., "key-value" access/column oriented /unstructured data)
    - Google Bigtable: atomicity on single keys
    - Yahoo PNUTS: serialized single-key writes → **timeline consistency**
    - Amazon SimpleDB: asynchronous writes → **eventual consistency**

Example big data stack

Google "specific" View

Open-Source Implementations from Apache/Yahoo

- New file system
  - **Google File System (GFS)**
    - Object-based distributed file system
- New database
  - **Google Big Table**
    - Column-based, key-value based data access
- New data processing
  - **Google Map & Reduce**
    - Distribute/collect jobs based on key-value grouping

Hadoop File System (HDFS)

Hadoop Database (HBase)

Hadoop Map & Reduce

Amazon, MS, OpenStack, all have different views

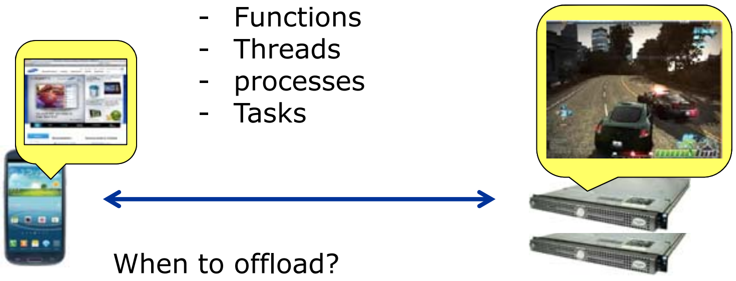
Outline

- Introduction
- Little more history
- **Issues in cost-effective cloud**
  - Performance
  - Power
  - RAS
  - Analysis
  - Storage
  - **Mobile**
- Cost-effective cloud @ POSTECH
- Summary

Why mobile cloud computing?

- **High performance**
  - Let’s borrow the server-scale power
    - E.g., 3D HD game using GPU @ datacenter
  - Let’s take advantage of cloud-scale information
    - E.g., Amazon Silk “cloud” browser’s predictive web caching
- **Longer battery life**
  - Let’s offloading mobile work to datacenter
    - E.g., Intel CloneCloud, Microsoft MAUI
- **Early time to market**
  - Let the cloud handle development-tricky things

Mobile cloud: task offloading



What to offload?
- Functions
- Threads
- processes
- Tasks

When to offload?
- High performance
- Long battery life

These are ongoing research issues..
BTW, is the cloud free?

Outline

- Introduction
- Little more history
- Issues in cost-effective cloud
- **Cost-effective cloud @ POSTECH**
  - **Compute Cloud**
  - Storage Cloud
  - Mobile Cloud
  - Cloud Workload
- Summary

PosCloud: Advanced open-source based cloud/big data system @ POSTECH

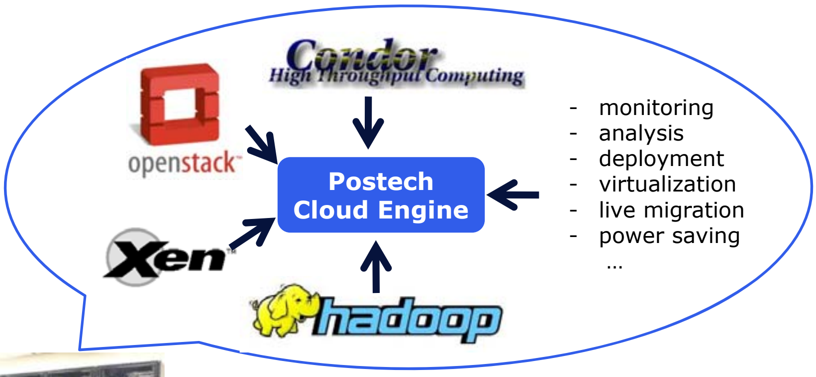
Commercial cloud solutions are VERY expensive

- **Can't use immature open-source solutions**
  - Lack of key features
    - e.g., monitoring, migration, RAS, backup, ...

- **Can't afford commercial solutions**
  - costs up to 1000s of dollars per CPU + licensing fees (for advanced management features)

Price for 1,000~10,000 nodes?
How to modify commercial engines?

PosCloud: open-source implementation



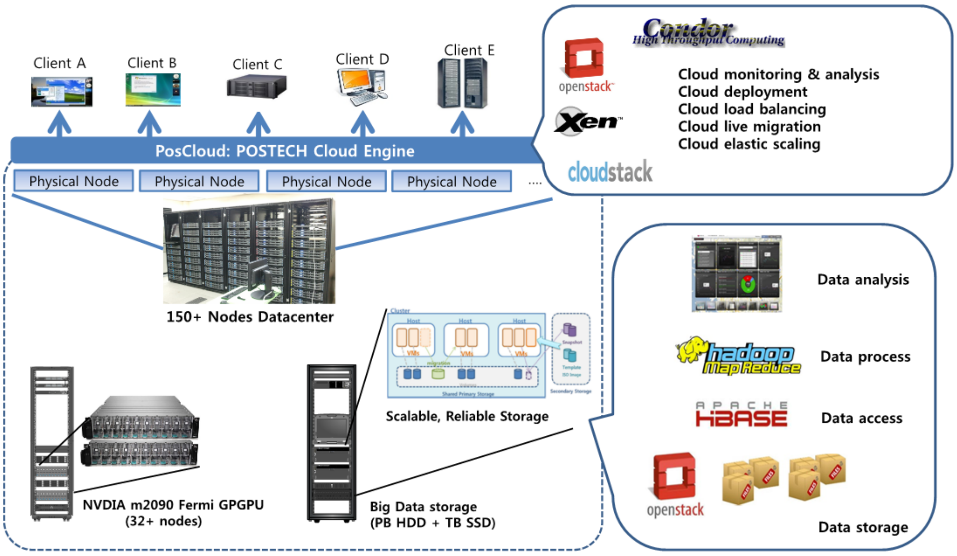
Postech Cloud Engine

- monitoring
- analysis
- deployment
- virtualization
- live migration
- power saving
...

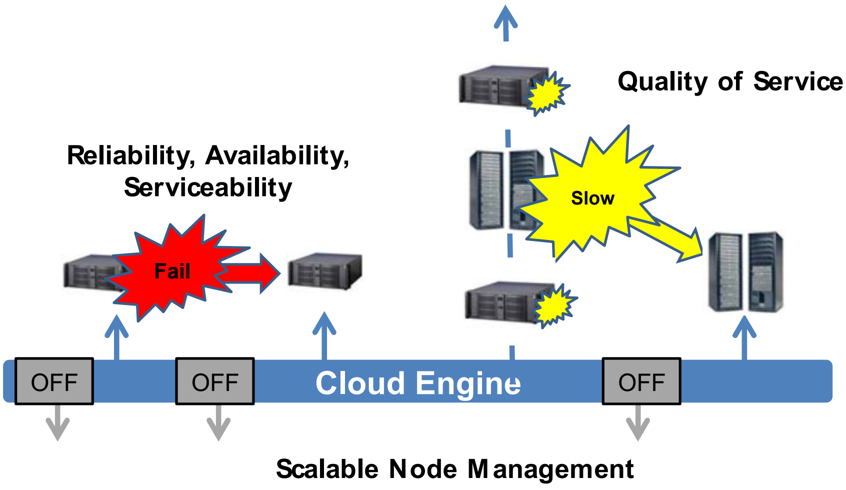
POSTECH Datacenter (100+ nodes)

Tools	Function
OpenStack	Infrastructure as a Service (IaaS)
Condor	Workload scheduling
Hadoop	Scalable file system
Xen	Virtualization

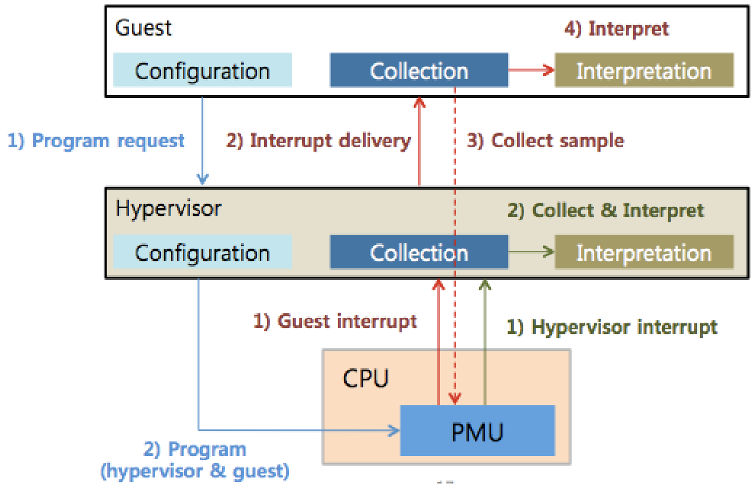
PosCloud: a big picture



PosCloud: dynamic resource management



PosCloud: system-wide monitoring



PosCloud: cost-effectiveness

Service Quality		Typical Open-source	Typical Commercial	PosCloud
IaaS Service		√	√	√
Cloud Computing Management	Performance	-	√	√+
	Power	-	√	√+
	Recovery	-	√	√+
	Availability	-	√	√+
	Other Features	-	?	√
Open-source Platform		-	-	√
S/W costs		~$0	1000s of $ per CPU	~$0

Commercial-level services at near zero prices!

Outline

- Introduction
- Little more history
- Issues in cost-effective cloud
- **Cost-effective cloud @ POSTECH**
  - Compute Cloud
  - **Storage Cloud**
  - Mobile Cloud
  - Cloud Workload
- Summary

The importance of scalable storage

- **Representative enterprise cloud storages**

Google File System

Amazon S3 (Dynamo)

Facebook Haystack

Yahoo Walnut / HDFS

And many more…

Evaluating cloud storage

- **Do existing solutions work well?**
  - Is it really fast?
  - Is it really scalable?
  - Is it really reliable?
- **What we are focusing on**
  - Identifying the performance bottleneck of a system
  - Using architectural support to improve existing storage system

Many research challenges for system architects

Our storage cloud system

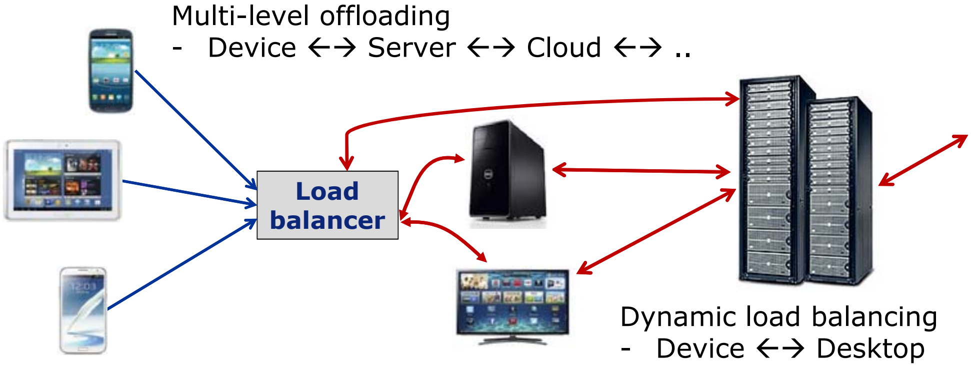
- **OpenStack Swift**
  - Popular open-source object storage system for cloud environments
  - Similar to enterprise storage system's architecture (Amazon's Dynamo)
  - Scalable and Reliable
- **State-of-the-art cluster**
  - Intel™ SandyBridge Xeon processors
  - 500TB storage capacity (SSD/HDD)

30+ storage servers

Outline

- Introduction
- Little more history
- Issues in cost-effective cloud
- **Cost-effective cloud @ POSTECH**
  - Compute Cloud
  - Storage Cloud
  - **Mobile Cloud**
  - Cloud Workload
- Summary

Mobile cloud: task offloading



Multi-level offloading

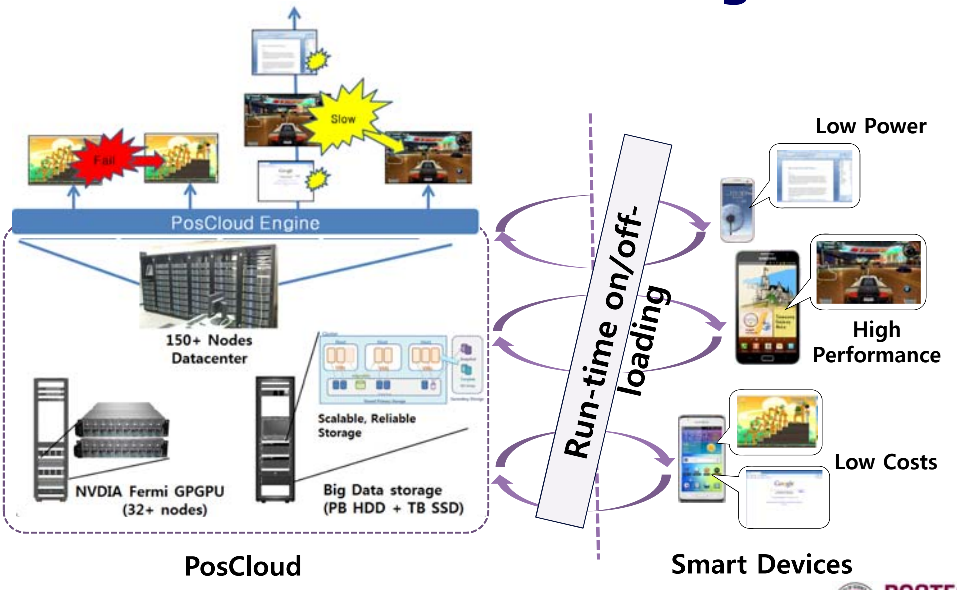
- Device ←→ Server ←→ Cloud ←→ ..

Dynamic load balancing

- Device ←→ Desktop
- Desktop ←→ Server
- Server ←→ Cloud

Dynamic offloading & balancing required!

PosCloud: mobile offloading



Outline

- Introduction
- Little more history
- Issues in cost-effective cloud
- **Cost-effective cloud @ POSTECH**
  - Compute Cloud
  - Storage Cloud
  - Mobile Cloud
  - **Cloud Workload**
- Summary

Making realistic cloud workloads

- **What a new workload should have**
  - Support for new infrastructure
    - Cloud services are based on distributed, scalable system
  - Realistic dataset
    - Cloud applications handle massive dataset, ..
    - No SQL, Map-Reduce, Web server, Mail server, Multimedia, ..
  - Evaluating virtualization performance
    - Cloud vendors use virtualization techniques to better utilize hardware resources

Two workloads on PosCloud

- **CloudSuite [from EPFL]**
  - Benchmark suite consists of scale-out applications
    - Covers broad range of applications: 6 different categories
- **SPECvirt [from www.spec.org]**
  - Performance evaluation of a single datacenter server
    - Covers all system components: hardware, virtualization platform, virtualized guest OS and application software

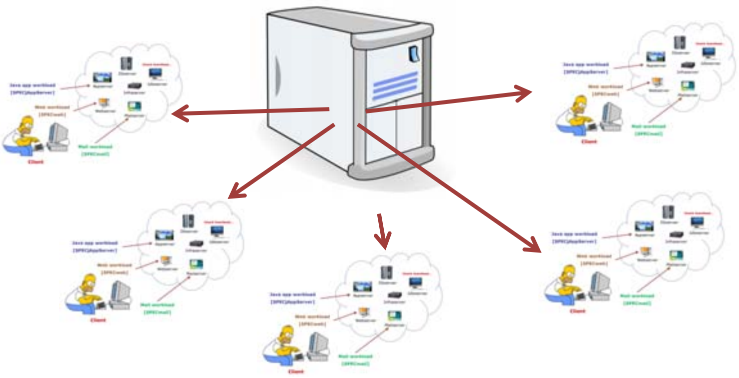
PosCloud: Real-world workloads

- **CloudSuite**
  - **Data Serving**
    - Serving data queries in a scalable noSQL storage system
  - **MapReduce**
    - Scalable machine learning library on Hadoop
  - **Media Streaming**
    - RTP/RTSP streaming server
  - **Software Testing**
    - Automated real-world software testing
  - **Web Serving/Search**
    - Search-oriented dynamic Web serve

PosCloud: Real-world workloads

- **SpecVirt**
  - OS : Centos 5.6
  - Virtualization : KVM-83
  - Webserver : Apache 2 with PHP 5
  - Infraserver : Apache 2 with fast-cgi
  - Appserver : Oracle Glassvish v2
  - Mailserver : Dovecot 1.2.17
  - DBserver : PostgreSQL 8

SPECvirt : Throughput + QoS test



Metric: How many VMs can be run,
while maintaining target QoS?

Research projects under PosCloud

- Mobile offloading
  - Representative real-world mobile applications
  - Potential workload reduction
- Datacenter performance monitoring
  - Performance counter virtualization
  - Resource contention identification
- Fast, live migration of virtual machines
  - Quality-of-Service guarantee
  - Load balancing for power re-cycling
- Big data management
  - Scalable object-oriented storage engine
  - SSD-HDD hybrid storage

Summary

- **Next computing platform = Big Future**

  (but, must be cost-effective!)

- **What we do @ POSTECH**
  - PosCloud
    - Mobile workload offloading
    - Quality-of-Service guarantee
    - Performance monitoring
    - VM, process, function migration
    - Cloud/Big Data workloads

Question?

Thank You!

Jangwoo Kim
e-mail: jangwoo@postech.ac.kr
http://hpc.postech.ac.kr/~jangwoo