POSTECH 포항공과대학교

Computer System Research @ POSTECH HPC

Jangwoo Kim

Feb 2, 2013

E-mail: jangwoo@postech.ac.kr

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

POSTECH

Cloud computing: a simple picture

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

Datacenter
: Warehouse-Scale Computer

Client A

Client B

Client C

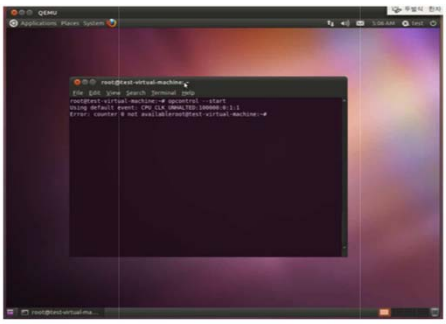
Client D

Client E

Datacenter + Virtualization + Application
→ Cloud Computing

POSTECH 포항공과대학교

VM-level analysis is TOO LIMITED

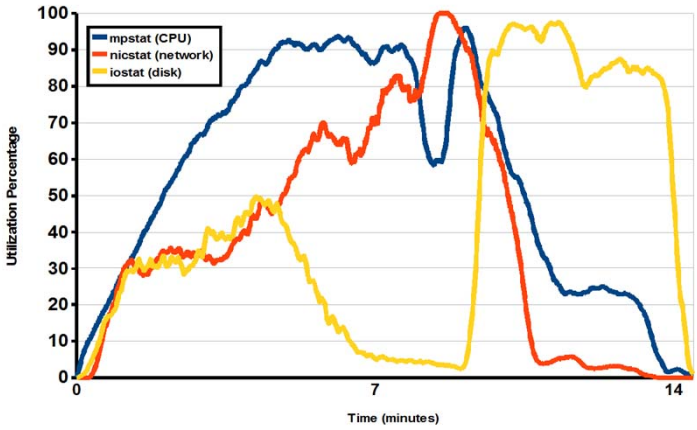


samples	%	
45249	98.6892	kvm_intel
364	0.7939	vmlinux
36	0.0785	r600_dri.so
22	0.0480	kvm

Difficult to analyze the performance of apps on VM

POSTECH 포항공과대학교

System-level analysis is TOO DIFFICULT



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

Various performance bottlenecks exist.

POSTECH 포항공과대학교

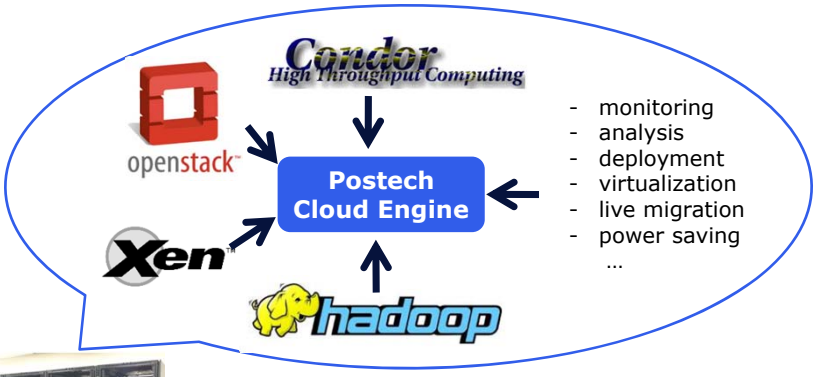
Commercial cloud solutions are TOO expensive

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

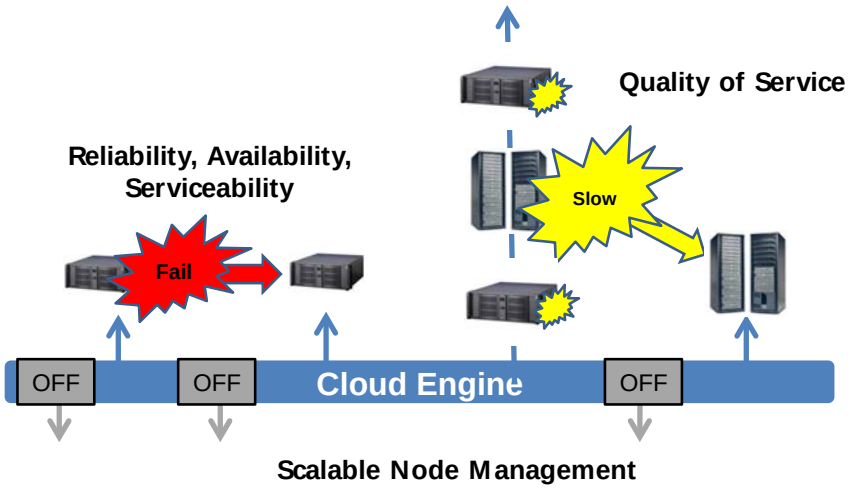
PosCloud: open-source implementation



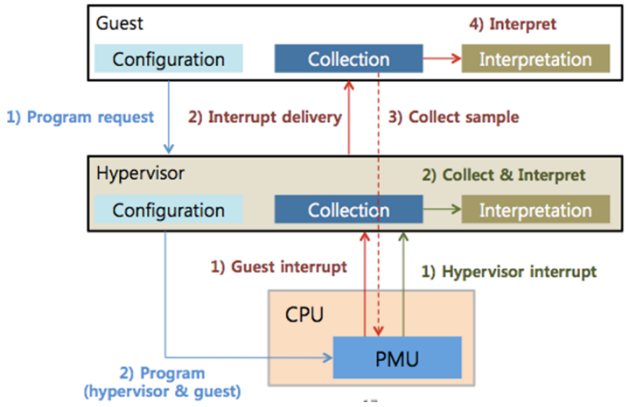
POSTECH Datacenter (100+ nodes)

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

PosCloud: dynamic resource management
(submitted to USENIX ATC'13)



PosCloud: VM-aware monitoring



PosCloud: Real-world workloads

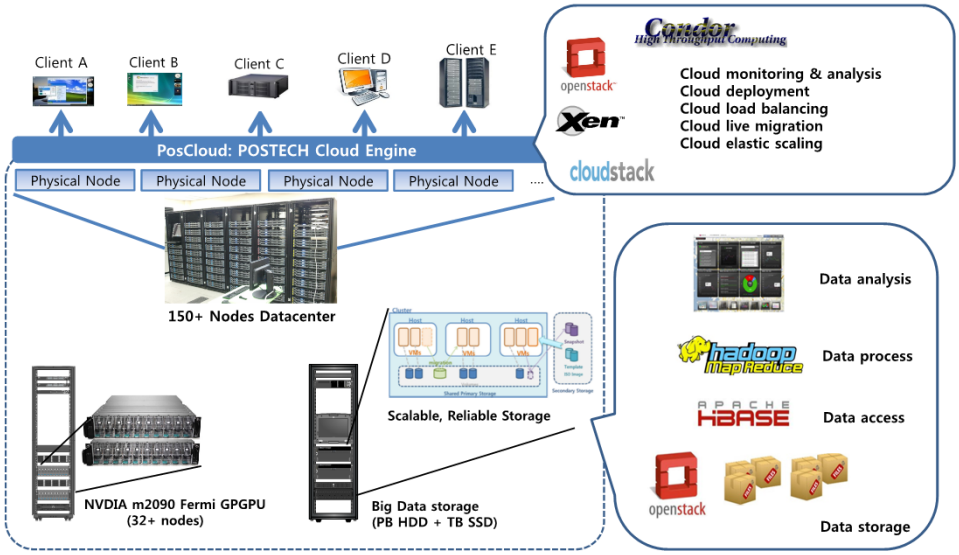
- **CloudSuite**
  - **Data Serving**
    - Serving data queries in a scalable noSQL storage system
  - **MapReduce**
    - Scalable machine learning library on Hadoop
  - **Media Streaming**
    - RTP/RTSP streaming server
  - **Software Testing**
    - Automated real-world software testing
  - **Web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

Cassandra mahout hadoop Darwin Streaming Server YouTube php Lucene nutch

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

PosCloud: a big picture



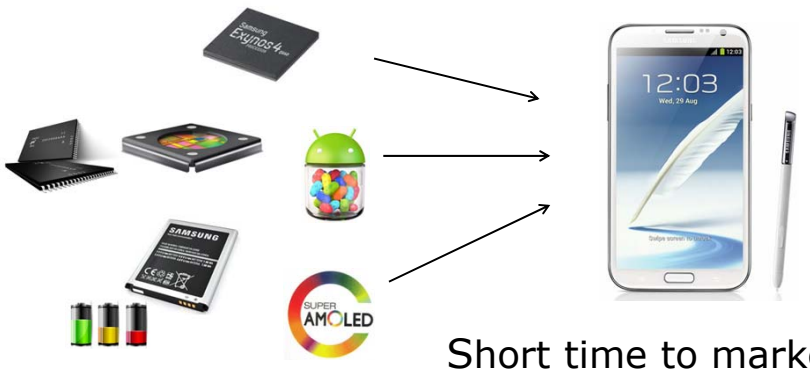
PosCloud: cost-effectiveness

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!

Smart device: a simple picture

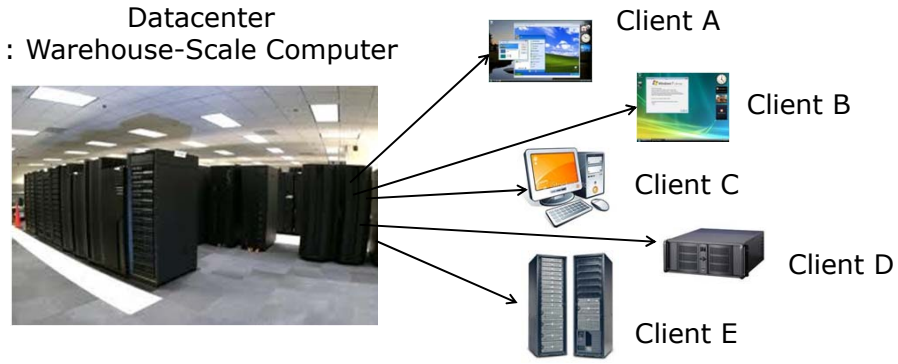
- **Assemble standard components fast and nicely**



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

Cloud computing: a simple picture

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

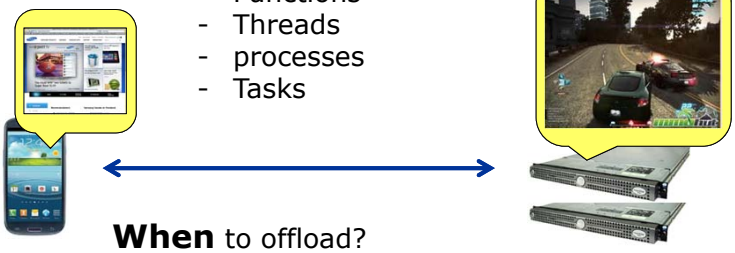


Datacenter + Virtualization + Application
→ Cloud Computing

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?

The importance of scalable storage

- **Representative enterprise cloud storages**

Google — Google File System

amazon — Amazon S3 (Dynamo)

facebook — Facebook Haystack

YAHOO! — Yahoo Walnut / HDFS

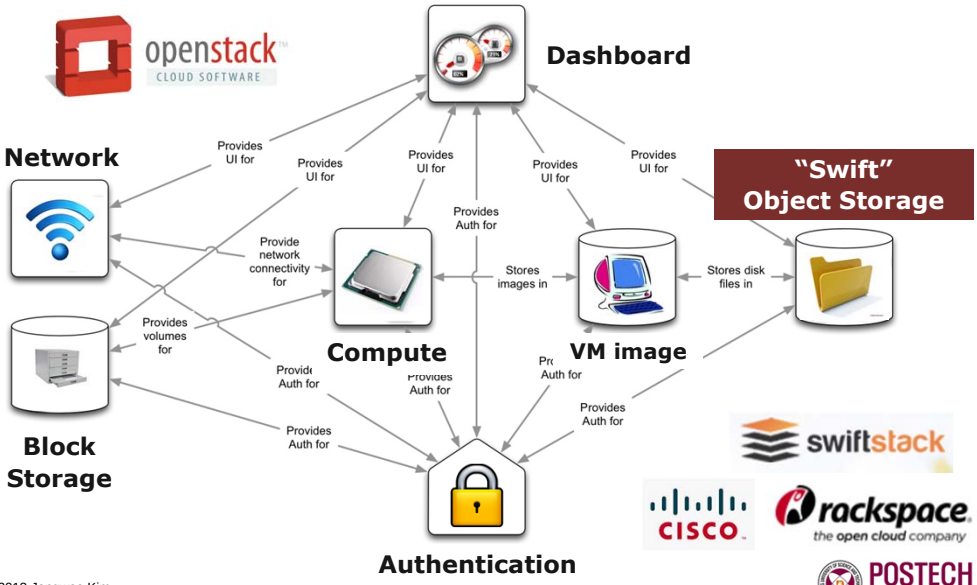
And many more…

Evaluating cloud storage system

- **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

OpenStack: cloud software framework



Our storage system testbed

- **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)

500TB storage servers

CPU Performance Analysis

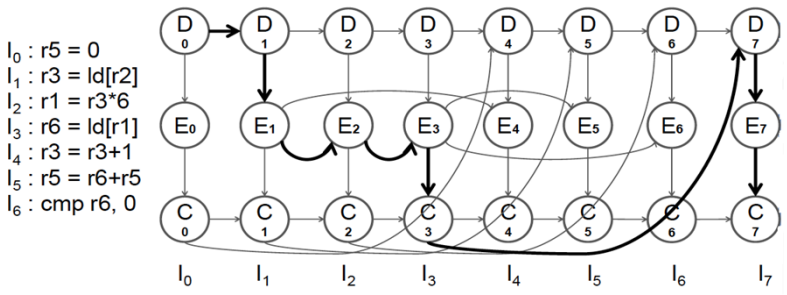
- **Where did CPU cycles go?**
  - Cache misses, branch mispredictions, data dependencies…
  - **Programmer's** aspects
    - Is my code optimal?
    - Why does this function take too long?
  - **Architect's** aspects
    - Which parts should be improved?
    - Do we need larger caches?
    - Can we reduce the issue width?

What people have been doing?

- **Performance Counter**
  - Counts various architectural events
  - Very fast, but **inaccurate**
- **Architecture model analysis**
  - Find critical instructions based on arch. knowledge
  - Quite accurate, but **difficult to apply**
- **Simulator**
  - Simulates every events of CPU and memory in cycles
  - Very accurate, but **extremely slow**

E.g., critical path theory

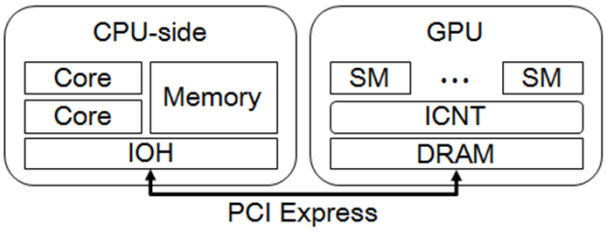
- **Typical example of arch. mode analysis**



Performance impact of modifying a critical path?
But, too many cases for too many design choices

GPU Research

- **Programming/optimizing GPUs is hard!**
  - **SIMD** programming model
  - **Hardware-dependent** optimization techniques
  - **Programmer-dependent** performance variation
  - **Program-dependent** optimal performance
  - …



Large-Scale GPU Programming

- **Difficult code modification**

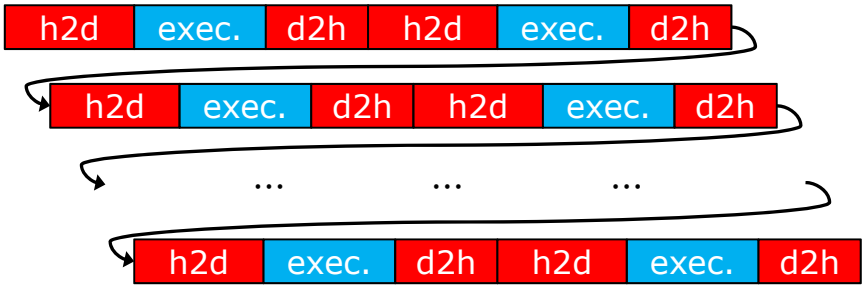
Small-scale	Large-scale

`// Perform C = A * B on GPU`
`MemcpyCPUtoGPU(matA);`
`MemcpyCPUtoGPU(matB);`
`CalcMatMultAll(matA, matB, matC);`
`MemcpyGPUtoCPU(matC);`
	`// Perform C = A * B on GPU`
	`stream_t strm[rows * cols];`
	`for (i = 0; i < rows; i++) {`
	`  for (j = 0; j < cols; j++) {`
	`  int idx = i * cols + j;`
	`  CreateStream(&streams[idx]);`
	`  MemcpyCPUtoGPU(matA[i][...], strm[idx]);`
	`  MemcpyCPUtoGPU(matB[...][j], strm[idx]);`
	`  CalcMatMultOne(matA, matB, matC, i, j, strm[idx]);`
	`  MemcpyGPUtoCPU(matC[i][j], strm[idx]);`
	`  PerformStream(strm[idx]);`
	`  }`
	`}`
	`for (i = 0; i < rows; i++)`
	`  for (j = 0; j < cols; j++)`
	`  SynchronizeStream(strm[i * cols + j]);`

Large-Scale GPU Programming

- **Huge performance degradation**
  - **Small-scale**

  - **Large-scale**



Summary

- **What we do @ POSTECH**
  - CPU performance modeling
  - Future GPU platform
  - PosCloud + PosMCloud
    - 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://www.postech.ac.kr/~jangwoo