

시스템 개발 방법론 @ 포항공대

김장우

2012년 7월 26일

포항공대 고성능 컴퓨팅 연구실

성공적인 시스템 개발 방법?



Google



목차

- 성공적인 시스템
- 성공적인 시스템의 개발 방법
 - 시스템 레벨 디자인
 - 시스템 모델링 및 시뮬레이션 기술
 - 시스템 워크로드 기술
- 시스템 개발 방법론 @ POSTECH

태권V를 성공적으로 개발하려면?



- 개발기간?
- 개발비용?
- 전투대상?
- CPU? 운영체제? 메모리?
- 철강재료?
- 부품 연결기법?
- 조종사 및 조종방법?
- 태권도? 검도?
- 전력공급?
- 물 속에서 싸워야 할까?
- 전투 중 오작동이 나면?
- ..

HW
OS
Time-to-market
Budget
Power
Devices
Control
Workload
Environment
Costs

도대체 무엇이 문제일까?

모델링/시뮬레이션을 통해
다양한 하드웨어와 소프트웨어들의 조합으로 이뤄진
복잡한 시스템의 성능 (또는 전력, 안정성..) 을
정확하게 분석하고 예측하는 것은 매우 어렵다.

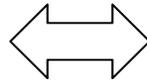
목차

- 성공적인 시스템
- 성공적인 시스템의 개발 방법
 - 시스템 레벨 디자인
 - 시스템 모델링 및 시뮬레이션 기술
 - 시스템 워크로드 기술
- 시스템 개발 방법론 @ POSTECH

시뮬레이션: 아키텍트의 주요 무기

시스템 개발용 시뮬레이터는 C/C++ 등의 언어로 이뤄진 대규모 소프트웨어

- 디자인 모델링
 - 성능/전력/안정성
 - 새로운 기술 적용
- 디자인 비교
 - 다양한 스펙/부품
 - 다양한 어플리케이션
- 디자인 평가
- 디자인 확정



**고급 시뮬레이션 기술이 없으면
경쟁력 있는 제품 개발 불가능**

시뮬레이션: 목표

- **정확한 시뮬레이션**
 - 목표 디자인과의 일치성 // 개발 정확도
(e.g., CPU vs 스마트폰 시뮬레이션)
- **빠른 시뮬레이션**
 - 목표 정확도의 시뮬레이션을 빨리 할 수 있는가? // 개발기간 단축
(e.g., 1일 vs 1달)
- **쉬운 시뮬레이션**
 - 시뮬레이션을 쉽게 할 수 있는가? // 개발비용 절감
(e.g., 시뮬레이터 개발비용? 인력확보?)

**고급 시뮬레이션 기술이란?
정확하고 빠르며 손쉬운 시뮬레이션 기술**

정확하고 빠른 시뮬레이션의 어려움 (1/2)

예: 고급 스마트 폰

▪ HW의 복잡도 증가

- 멀티코어/멀티쓰레드 CPU/..
- 메모리/플래쉬/..
- GPU/디스플레이/...



▪ SW의 복잡도 증가

- 운영체제/시스템소프트웨어/..
- 멀티미디어/영화/게임/..
- 소셜네트워크/클라우드/..



이 모두를 프로그램으로 정확히 구현하면
시뮬레이터의 느린 속도로 인해 개발적용 불가능

정확하고 빠른 시뮬레이션의 어려움 (2/2)

예: 멀티코어 CPU

- 실제 CPU 대비 속도저하도

- 단순 에뮬레이션 모드 < 2배
- SW 개발용 시뮬레이터 ~1000배
- 시스템 (HW/SW) 개발용 시뮬레이터 ~1000,000배

“실제 CPU의 1분 → 1년의 시뮬레이션 기간”

성공적인 시스템 개발을 위해 선택적 시뮬레이션,
워크로드 축약 등의 고급 시뮬레이션 기술 필요

목차

- 성공적인 시스템
- **성공적인 시스템의 개발 방법**
 - 시스템 레벨 디자인
 - 시스템 모델링 및 시뮬레이션 기술
 - 시스템 워크로드 기술
- 시스템 개발 방법론 @ POSTECH

워크로드 개발 및 운영의 어려움

예: 어떤 워크로드를 대상으로 개발을 해야 할까?

- 풀 시스템 워크로드 (full system workload)

- 사용자가 바라보는 워크로드
 - 3D 게임, 고화질 동영상, 끊어지지 않는 화상통화 등
- 시뮬레이터에서 사용하는 워크로드
 - 서킷 시뮬레이션, 단순계산 프로그램, 메모리 테스트 프로그램

“성공적인 모델링 =

성공적인 시뮬레이션 + 의미있는 워크로드”

**풀 시스템 워크로드를 모델링/시뮬레이션이
가능하게 축약하는 고급기술 필요**

목차

- 성공적인 시스템
- 성공적인 시스템의 개발 방법
- 성공적인 시스템 개발 방법론 @ POSTECH
 - CPU 분야 연구
 - GPU 분야 연구
 - Cloud/Big Data 분야 연구
- 요약

Our approach: Critical-Path Aware Fast Simulation

- One smart simulation only, and theory tells everything
 - E.g. “ 35% loss due to L2 d-cache miss
30% loss due to branch misprediction
20% loss due to floating-point division
15% loss due to data dependency “
- Key advantages
 - Algorithm needs to analyze only in-flight instructions
 - Post-analysis tools can be used to reduce # of simulations
 - Statistical data-correlation methods can be applied
 - Sim VS Sim comparison is easy (e.g., RTL vs Timing)

목차

- 성공적인 시스템
- 성공적인 시스템의 개발 방법
- 성공적인 시스템 개발 방법론 @ POSTECH
 - CPU 분야 연구
 - GPU 분야 연구
 - Cloud/Big Data 분야 연구
- 요약

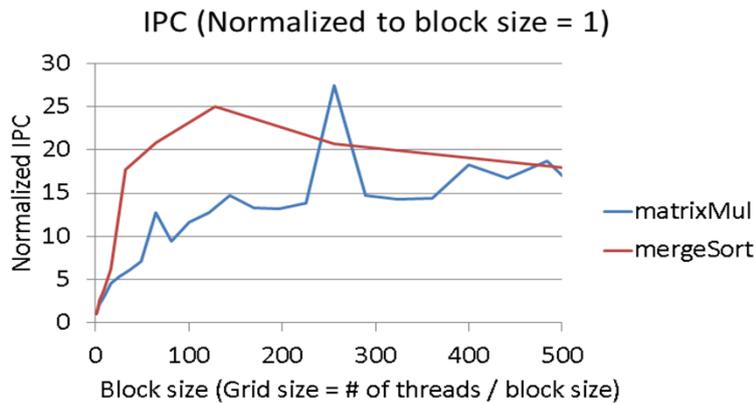
GPU programming is HARD

- Programmers must do **everything!**
- Programmers must do everything **right!**
- Programmers must do this process **over and over!**

**GPU is useless to the most of programmers,
if you care PERFORMANCE.**

Programming (for performance) is Hard

- Application-dependent performance variation



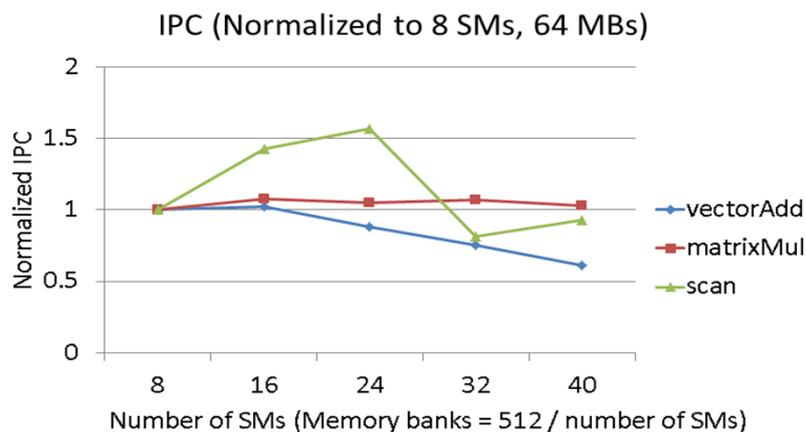
IPC varies with grid/block combinations

© 2012 Jangwoo Kim



Portability (for performance) is Hard

- Architecture-dependent performance variation



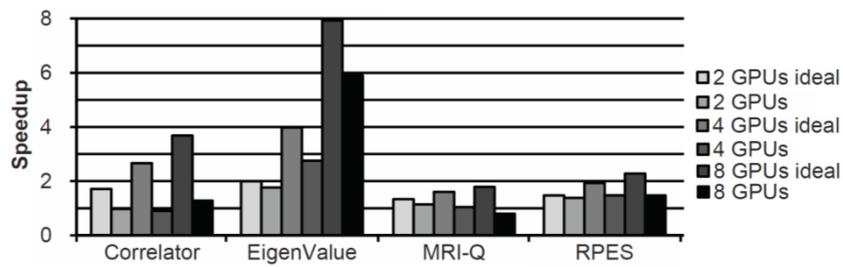
IPC varies with architecture combinations

© 2012 Jangwoo Kim



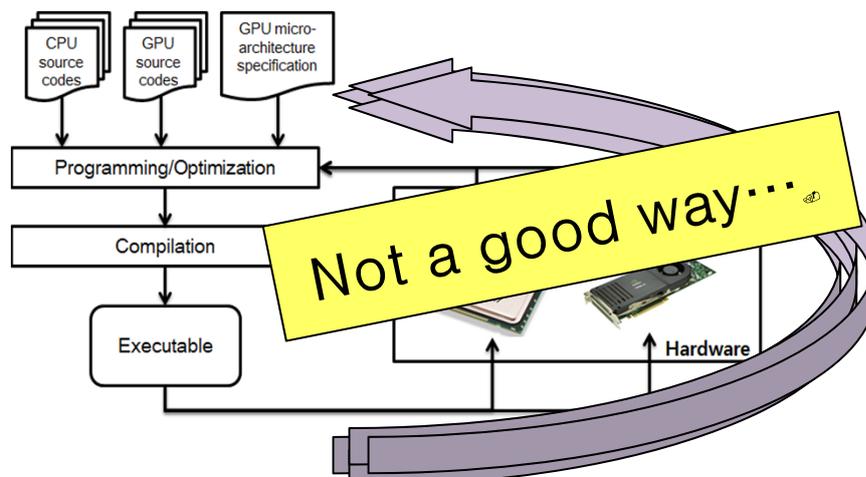
Scalability (for performance) is Hard

- Multi GPU–dependent performance variation



Performance won't increase even if you add more GPUs

Current Programming & Optimization

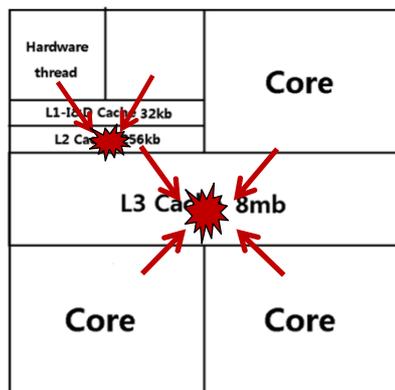


Unbounded # of painful re-optimizations (if possible)

목차

- 성공적인 시스템
- 성공적인 시스템의 개발 방법
- 성공적인 시스템 개발 방법론 @ POSTECH
 - CPU 분야 연구
 - GPU 분야 연구
 - Cloud/Big Data 분야 연구
- 요약

Performance cannot be guaranteed



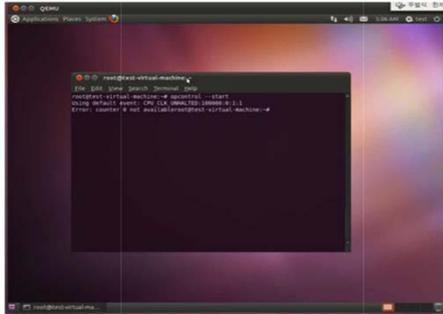
[Example 4-core CPU]

No true **'multi'** thing!

Virtual machine is only 'virtual'

- > Resource contention exists
 - ALUs,
 - caches
 - I/O devices
 - Net bandwidth
- > Difficult performance analysis

Performance cannot be monitored



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 my system

There is no standard workload for cloud

• How to model workloads on cloud?

–Cloud workloads are unique

▪ Typical

○ SPEC CPU, PARSEC, TPC-C, TPC-H, ...

□ Known for straightforward instruction-, memory-, and thread-parallelism

▪ Cloud

○ NoSQL database, web server, mail server, map & reduce, ...

(likely on top of virtualization)

□ Known for large scale, network and I/O bound, complicated behaviors

How to model, build, reduce, and evaluate
cloud workloads?

Cloud software is 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)

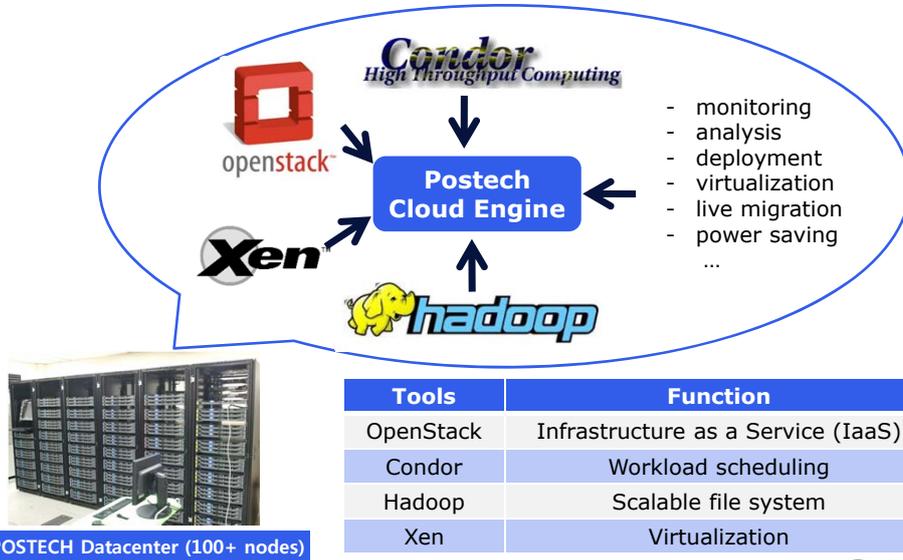
**Price for 1,000~10,000 nodes?
How to modify commercial engines?**

Our approach @ POSTECH

PosCloud:

Advanced open–source based cloud engine

PosCloud: open-source implementation

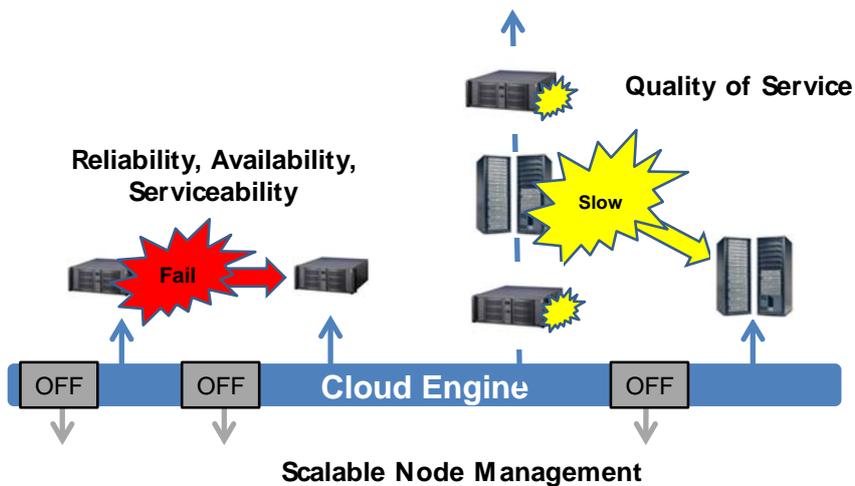


© 2012 Jangwoo Kim

26



PosCloud: advanced operating system

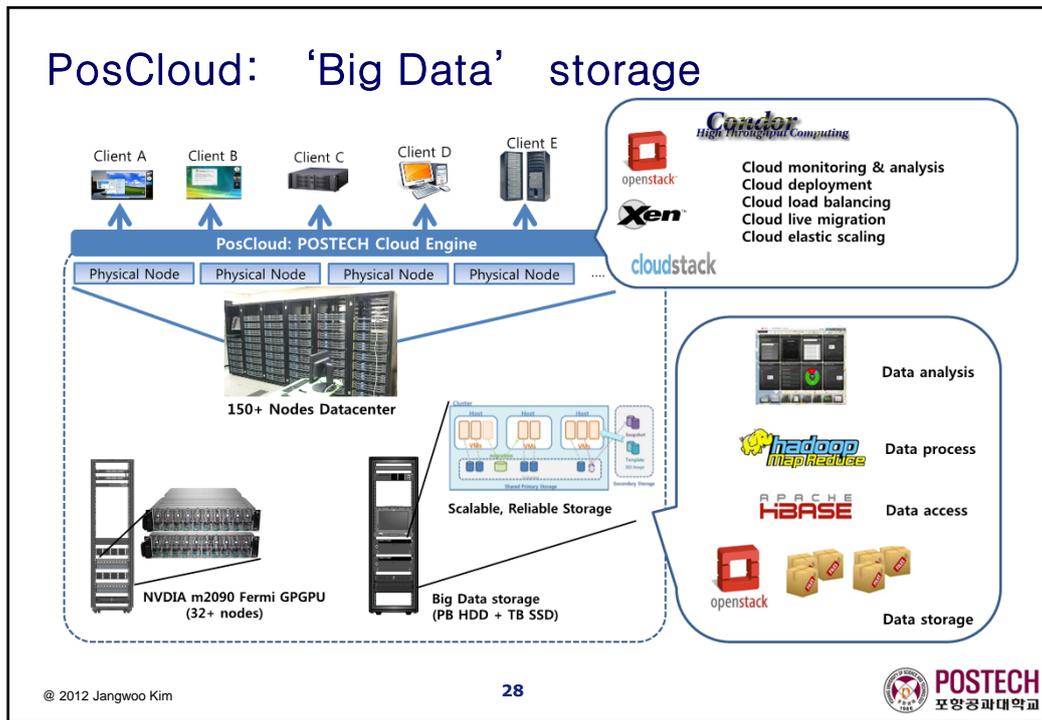


© 2012 Jangwoo Kim

27



PosCloud: 'Big Data' storage



PosCloud: Workload

- **CloudSuite**

- Benchmark suite consists of scale-out applications
 - Covers broad range of applications: 6 different categories

- **SPECvirt**

- Performance evaluation of a single datacenter server
 - Measures all system components: hardware, virtualization platform, virtualized guest OS and application software

PosCloud vs Others

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!

Research projects under PosCloud

- Performance monitoring
 - Performance counter virtualization
 - Resource contention identification
- Fast, live migration of virtual machines
 - Quality-of-Service guarantee
 - Load balancing for power re-cycling
- Realistic workloads
 - Large-scale, virtual machine throughput test
 - Potential workload reduction
- Big data management
 - Scalable object-oriented storage engine
 - SSD-HDD hybrid storage

Summary

- What we are doing @ POSTECH
 - CPU performance analyzer
 - Fast and accurate simulation and modeling
 - GPU programming
 - Compiler and architectural supports
 - PosCloud: open-source cloud engine
 - Quality-of-Service guarantee
 - Performance monitoring and prediction
 - VM, process, function migration
 - Cloud-level workloads
 - Scalable & reliable big data storage

Question?

Thank You!

Jangwoo Kim
e-mail: jangwoo@postech.ac.kr
<http://www.postech.ac.kr/~jangwoo>