Cloud Computing

Tran Giang Son, tran-giang.son@usth.edu.vn

ICT Department, USTH

What & Why?

What?

- Externalize computing infrastructure
  - Customer
  - Provider
- Concentration
  - Customer : business
  - Provider : infrastructure management
- Connects the customer's needs with the provider's services

What?

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"

- NIST

Chacteristics

- On-demand Self-service
  - Less / more resource at runtime
  - Automatically served
- Remote Access
  - Resource can be accessed remotely over the network

Chacteristics

- Resource Pooling
  - Provider's resource is shared by multiple customers
  - Dynamically assigned and reassigned to different customers at runtime
- Rapid Elasticity
  - Resource can be elastically expanded and collapsed rapidly at runtime

Related Technologies

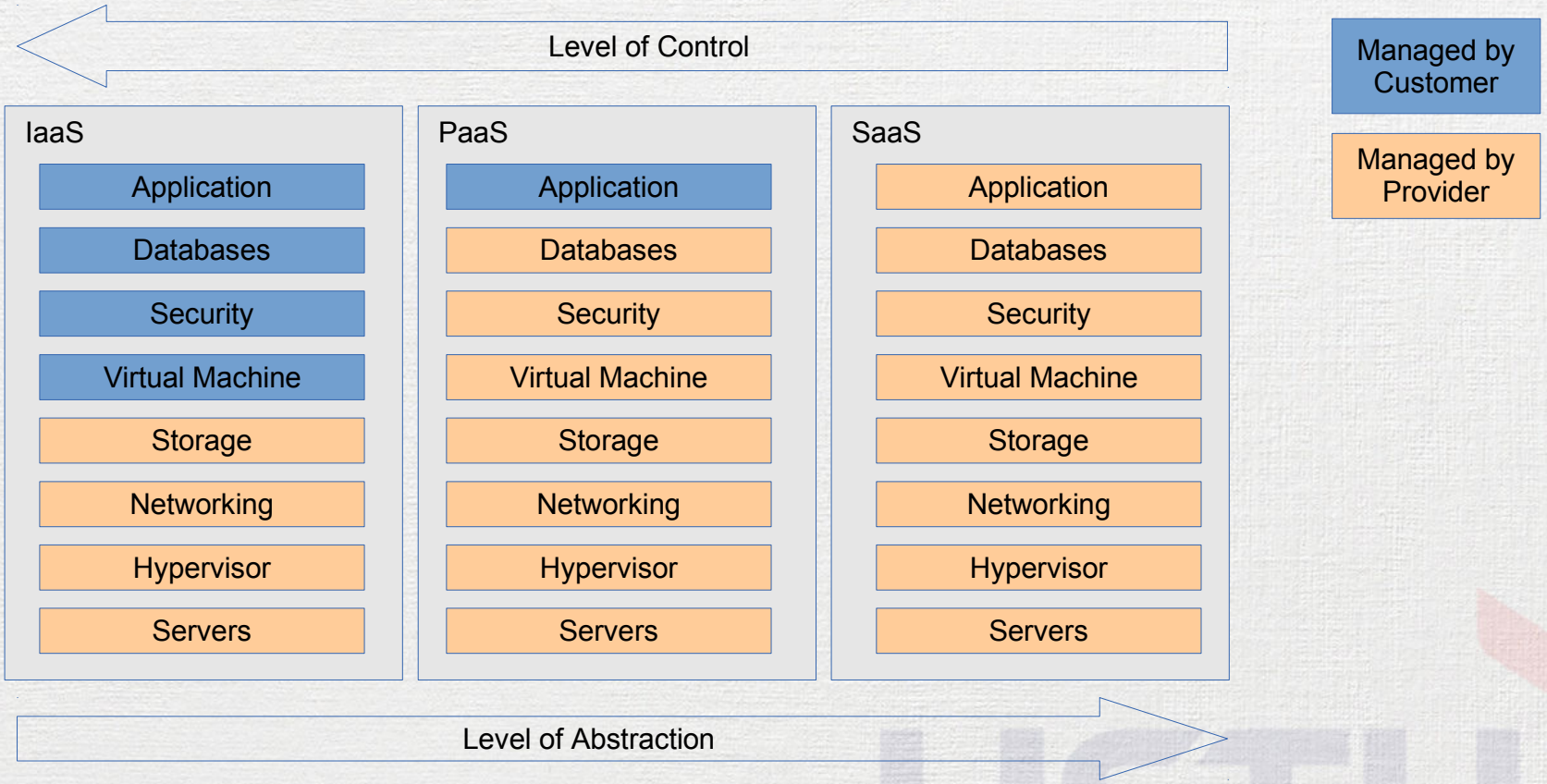
- Grid computing
  - Hardware deployment
  - Hardware management
- Virtualization
  - Resource sharing
- Utility computing
  - Metered and billed resources

Contracts

- Quality of Service
  - Number of allocated processors
  - FLOPS
  - Dedicated memory
  - Storage
  - Network bandwidth
- Service Level Agreements
  - Don't be greedy ☺

Model

- IaaS. PaaS. SaaS.



Model: IaaS

- Virtual machines
  - Primitive form of resources: computing, memory, network, storage
  - Load balancer
- Customer takes care of
  - OS
  - Applications
- EC2, GCE, Azure

Model: PaaS

- Virtualized execution platform
  - API
  - Libraries
  - Services
  - Tools
- Customer creates apps on top of platform
- GAE, Elastic Beanstalk

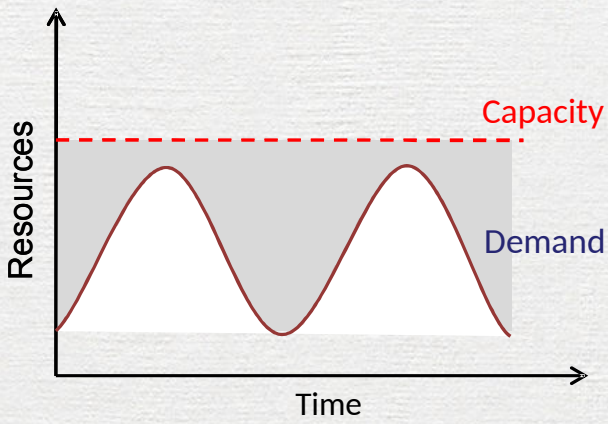
Model: SaaS

- Provides software
  - Accessible with thin client interface
  - Web browser
- Customer uses the app
- Google Apps, MS Office 365

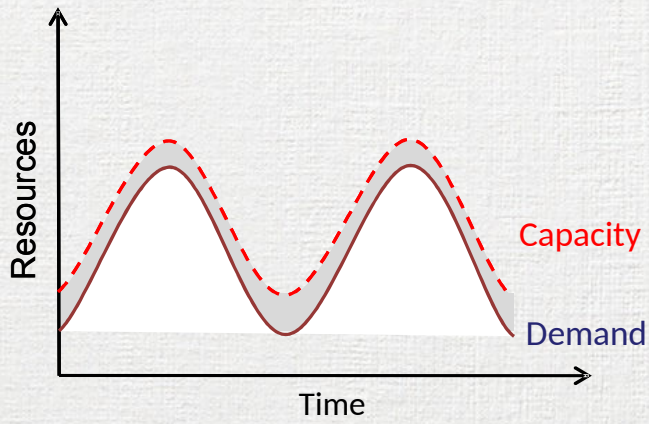
Why?

- Benefits: dedication
- Customer
  - Unlimited resources
  - Scalability
  - Pay-as-you-go...
- Provider
  - Resource sharing
  - Only pay for initial investment

Why?



Static data center



Data center in the cloud

Why NOT?

- Performance
- Data privacy
- System security
- Standardization
- Resource management

A New Model

What about Teacher-as-a-Service?

Virtualization

Virtualization

- Software- and/or hardware-based solution
- Building and running many operating systems simultaneously
- Separate physical hardware and the executing operating systems

Virtualization

- Software- and/or hardware-based solution
- Building and running many operating systems simultaneously
- Separate physical hardware and the executing operating systems
- E.g.

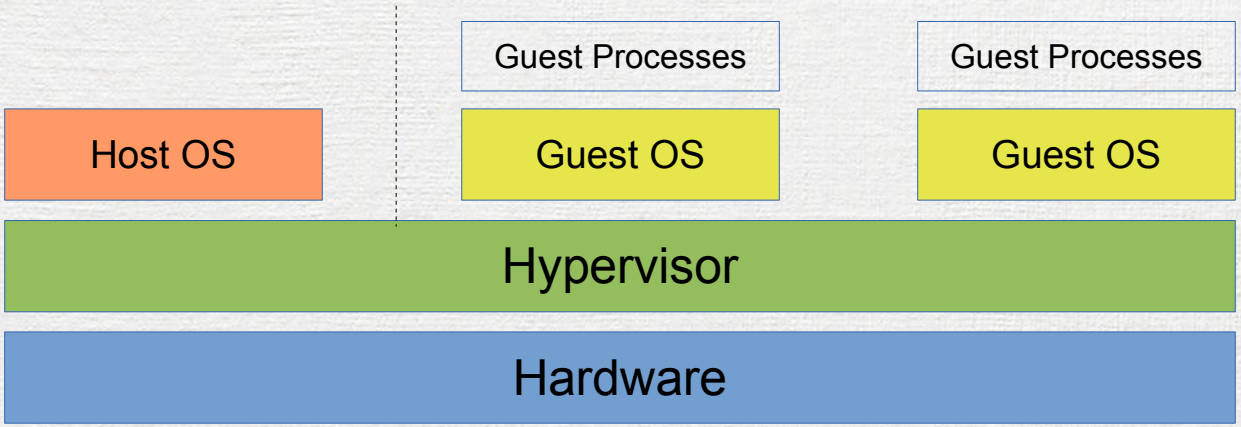
Virtualization

- Software- and/or hardware-based solution
- Building and running many operating systems simultaneously
- Separate physical hardware and the executing operating systems
- E.g. VMware Workstation, VMware ESXi, VirtualBox, kvm, Xen, OpenVZ, Parallels Desktop, Hyper-V, Docker, LXC

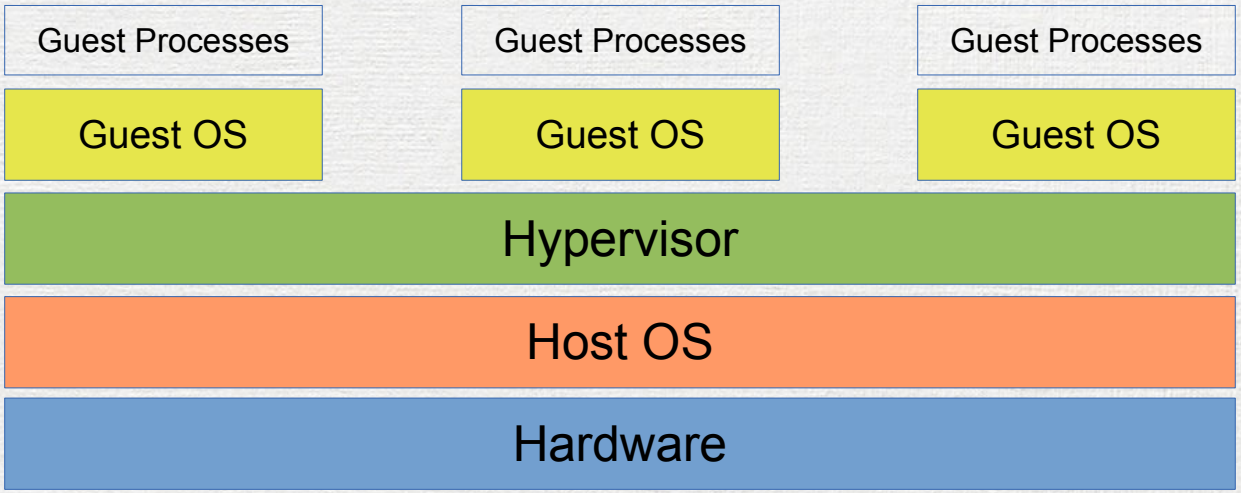
Hypervisor

- A.k.a. Virtual Machine Monitor
- Responsible for hardware emulation and communication
  - CPU
  - Memory
  - Graphic cards
  - Audio
  - Storage
  - Network interface
- Share its hardware resources to Guest OS

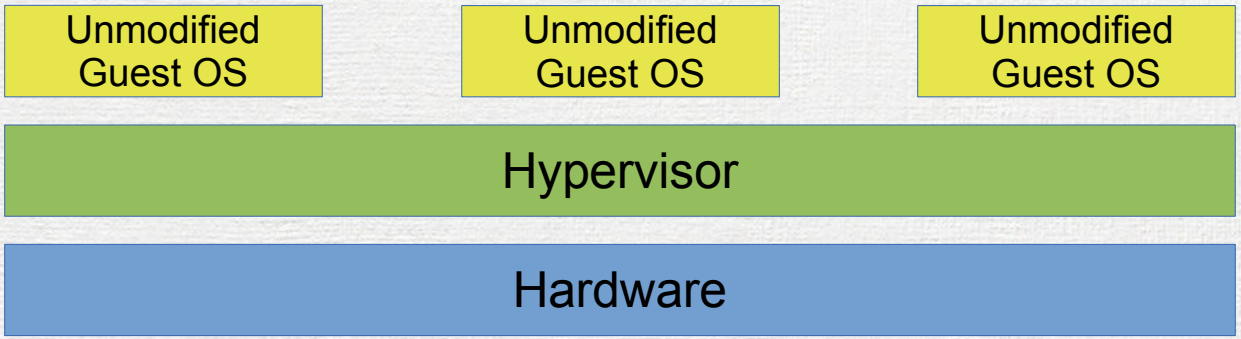
Hypervisor Type 1: "Bare Metal"



Hypervisor Type 2: "Hosted"

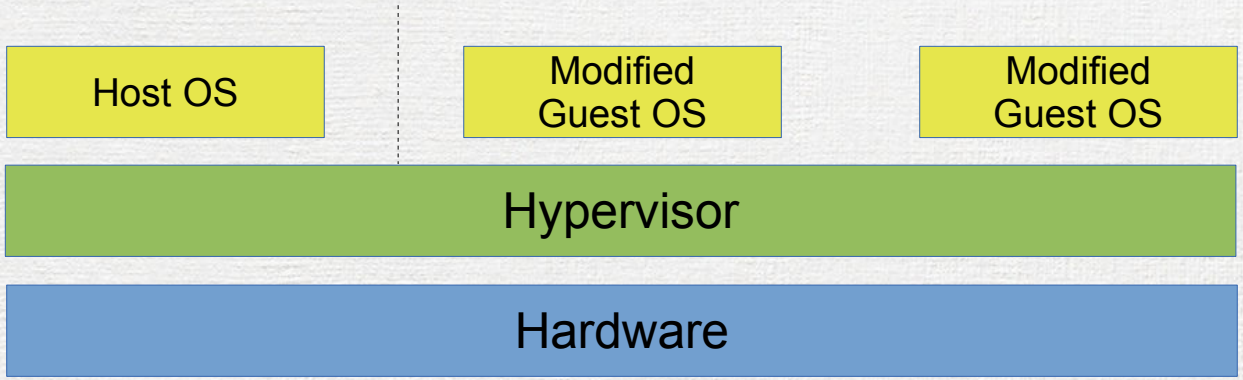


Hardware Level: Full Virtualization



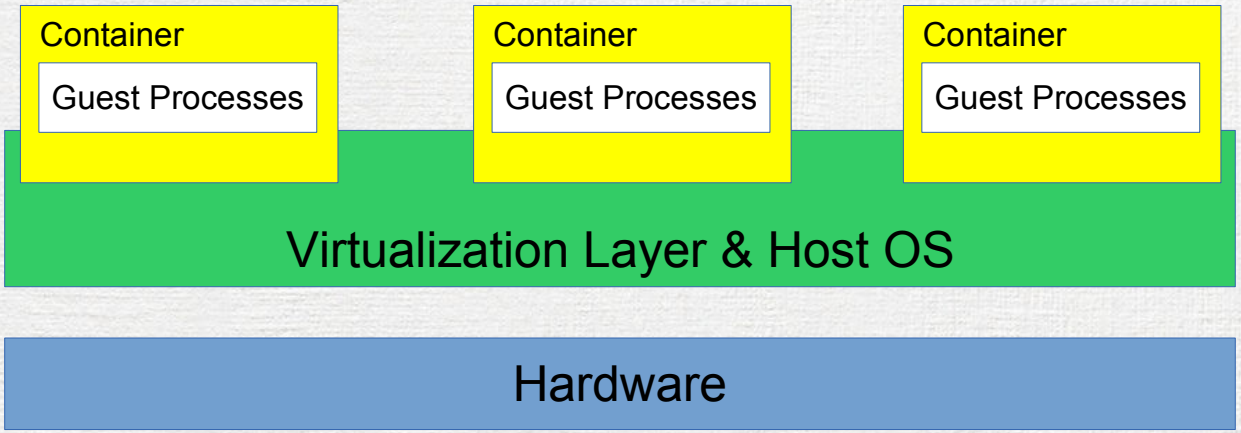
- Allows virtualization of unmodified guests
  - "Binary translation"
  - Parse, translate, execute
  - Possible different platform e.g. ARM on x86
  - Slow

Hardware Level: Paravirtualization



- Guest is modified
  - Better performance
  - Less overhead
  - No binary translation
  - Easier to implement

Operating System Level Virtualization



- Hypervisor integrated inside Host OS
- Containers
- Isolated
- Same kernel
- Fast!

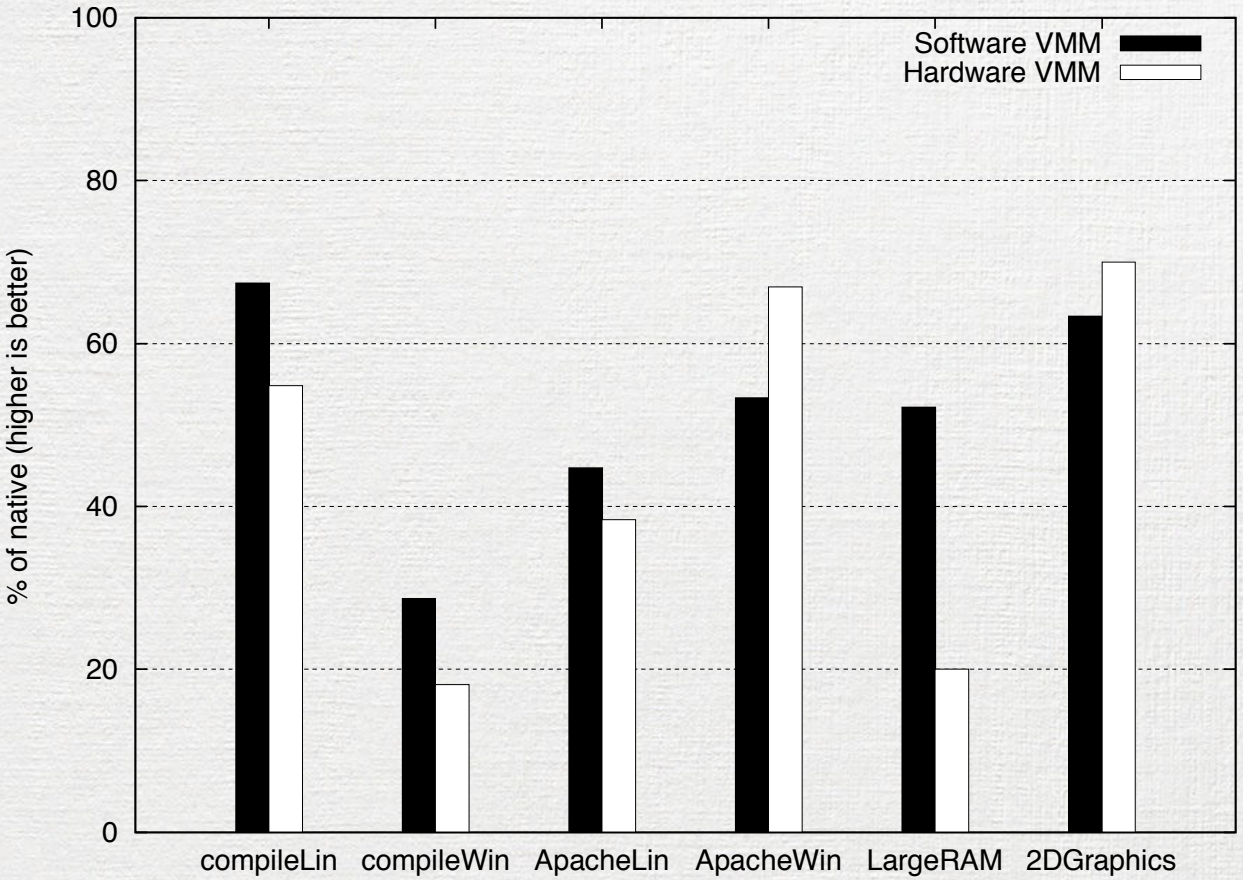
Why?

- Cost reduction
  - Hardware cost
  - Maintenance cost
- Hardware utilization
- Isolation
- Autonomic resource management
  - Both hardware entity and software entity
  - Reduce cost
  - Scalabilty

Why NOT?

- Lack of standards
- Overhead vs unvirtualized systems
  - Cost of virtualization

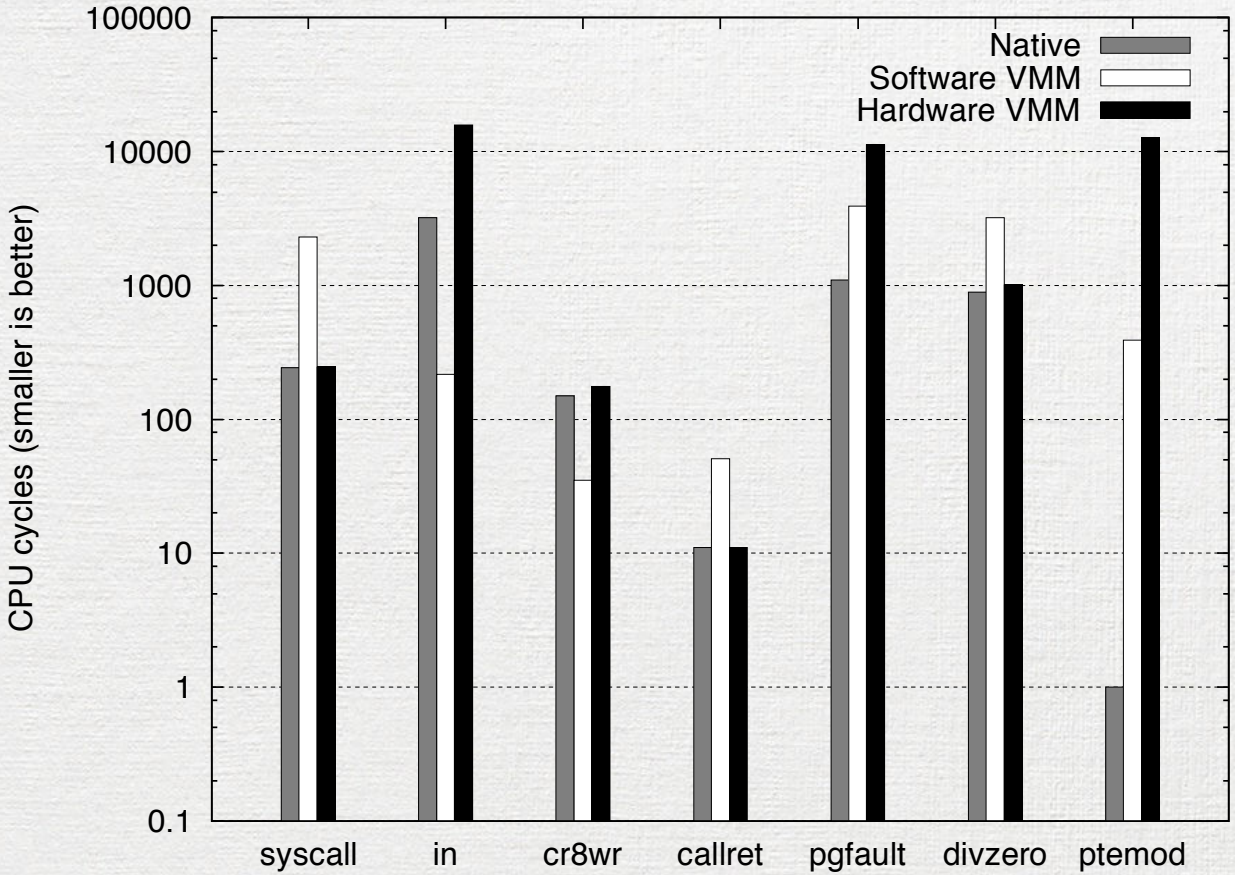
Why NOT?



Macro benchmark[1]

[1] Adams, K. and Agesen, O. (2006). A comparison of software and hardware techniques for x86 virtualization. In *ACM SIGOPS Operating Systems Review*, volume 40, pages 2–13. ACM.

Why NOT?



Nanobenchmarks[2]

[2] Adams, K. and Agesen, O. (2006). A comparison of software and hardware techniques for x86 virtualization. In *ACM SIGOPS Operating Systems Review*, volume 40, pages 2–13. ACM.

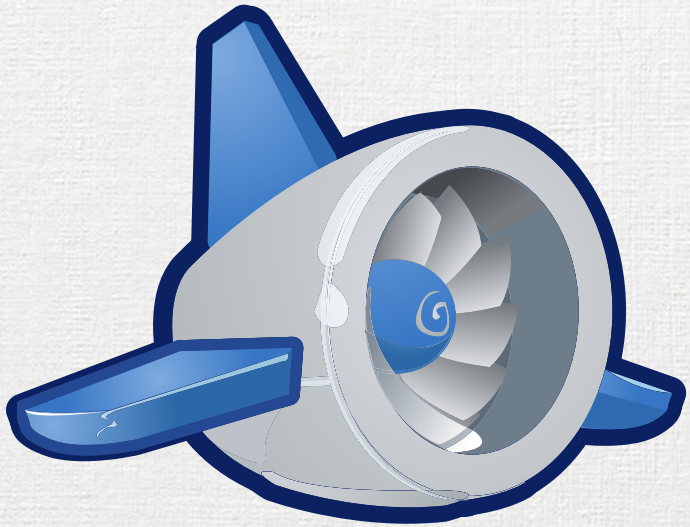
How

How?

- Cloud == very large term
- PaaS on GAE

Google App Engine

- Web framework
- Cloud computing platform
- Google infrastructure
- Sandboxed applications
- Automatic scaling
- Free for small projects



Google App Engine

- Languages: Java, Python, Ruby, Go, PHP
  - Java: Servlet 2.5 (with Jetty), JSP, JSF, Spring, Struts
- Data: No direct file system writes
- Database: Google Cloud Datastore, JPA, JDO
- SLA 99.95% uptime

Google App Engine: How?

- In this course, Java
- Good IDE: Eclipse, Intellij IDEA Ultimate,...
- Google Cloud SDK
- Plenty of tutorials

Practical work 7: GAE

- http://console.cloud.google.com
- Create a Sudoku game on GAE
- Write a short report in LaTeX:
  - Name it « 07.sudoku.tex »
  - How you structure your practical work
  - Important snippets.
  - Who does what.
- Work in your group, in parallel
- Push your report to corresponding forked Github repository