

Program Fresh Graduate Academy Digital Talent Scholarship 2019 | Machine Learning

Pendahuluan Cloud Computing

Isman Kurniawan, Ph.D.





Bagian 1

Sejarah Cloud Computing

Introduction

- No longer the next big thing – the current big thing:
 - Began in 2007 – IBM and Google “Blue Cloud”.
 - Name cloud inspired by cloud symbol representing internet in diagrams.
 - Amazon popularized idea of the cloud.

Sejarah Cloud Computing

- ***Cloud computing*** adalah hasil dari evolusi bertahap di mana sebelumnya terjadi fenomena *grid computing*, virtualisasi, *application service provision (ASP)* dan *Software as a Service (SaaS)*.
- Konsep penyatuan *computing resources* melalui jaringan global sendiri dimulai pada tahun 1960-an.
- Saat itu muncul “*Intergalactic computer network*” oleh J.C.R. Licklider.

Sejarah Cloud Computing

- Licklider bertanggung jawab atas pembangunan ARPANET (Advanced Research Projects Agency Network) di tahun 1969.
- J.C.R. Licklider memiliki sebuah cita-cita di mana setiap manusia di dunia ini dapat terhubung dan bisa mengakses program dan data dari situs manapun, di manapun.

Sejarah Cloud Computing

- Perkembangan berikutnya adalah adanya **Amazon Web Services** di tahun 2006, dengan teknologi **Elastic Compute Cloud (EC2)**.
- Tumbuhnya situs layanan web yang di komersialkan sehingga memungkinkan perusahaan kecil dan individu untuk menyewa komputer atau server untuk menjalankan aplikasi komputer mereka.

Keuntungan Cloud Computing

- ***Cloud computing*** menghapus silo-silo dalam ‘*data center*’ tradisional.
- Arsitektur awan memiliki **skalabilitas, fleksibilitas, dan transparansi** yang memungkinkan layanan TI dapat disediakan dengan cepat dan biaya efektif.
- Arsitektur awan menggunakan *service level agreements (SLA)* yang mencakup *IT requirement* dan *policy* untuk menyepakati terkait *high utilization*, dinamika, respon perubahan, dan tingkat keamanan dan kinerja.

Keuntungan Cloud Computing

- *Reduced cost*
- *Flexibility*
- *Improved Automation*
- *Sustainability*
- *Focus on Core Competency*

Dampak Cloud Computing

- *Cloud computing* memiliki potensi membalikkan paradigma industri *software*, yang tadinya suatu aplikasi harus dibeli dan dijalankan di desktop, sementara pada cloud aplikasi dan lisensi berjalan melalui jaringan.
- Dengan adanya perubahan ini akan memungkinkan *data center* dan administrator-administrator yang berada di pusat jaringan distribusi, *processing power*, *electrical*, *bandwidth* dan *storage* dapat dikelola secara *remote*.
- Hal ini tidak hanya berdampak kepada model bisnis, tetapi juga berkaitan dengan arsitektur utama bagaimana suatu aplikasi dikembangkan, dibangun dan dijalankan.

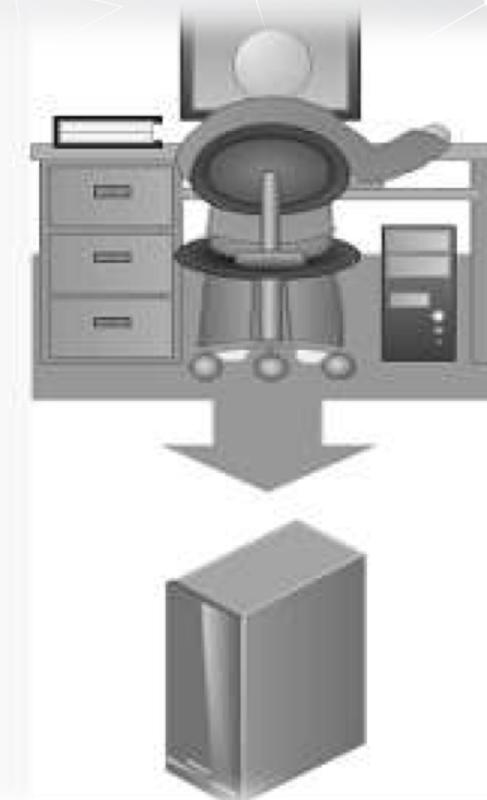
Perbandingan Cloud Computing dan Model Computing Lainnya

- Perbedaan dengan desktop biasa dengan *cloud computing*
- Perbedaan jaringan komputer dengan *cloud computing*
- Evolusi model *computing*
- Hubungan *cloud computing* dengan model *computing* lainnya
 - *Grid Computing*
 - *Utility Computing*
 - *Distributed Computing*
 - *Computer Cluster*
 - *Virtualization*

Perbedaan Desktop biasa dengan Cloud Computing

Desktop biasa

- PC sentris.
- Perangkat lunak dijalankan pada komputer tersebut.
- Semua dokumen yang dibuat, diolah dan simpan Pada komputer tersebut.



Perbedaan Desktop biasa dengan Cloud Computing

Cloud computing

- Dokumen sentris.
- Program perangkat lunak yang digunakan tidak berada pada komputer kita, melainkan tersimpan pada server-server yang diakses melalui internet.



Perbedaan Jaringan Komputer dengan Cloud Computing

- ***Cloud computing*** bukanlah jaringan computer dikarenakan dalam jaringan komputer aplikasi/ dokumen tersimpan pada server perusahaan dan hanya dapat diakses melalui jaringan komputer perusahaan.
- ***Cloud computing*** lebih besar dari jaringan computer karena melibatkan lebih banyak perusahaan, server, dan jaringan, sementara jaringan komputer hanya dapat diakses pada jaringan komputer perusahaan itu sendiri.
- Pada ***cloud computing***, ***cloud services*** dan ***storage*** dapat diakses dari mana saja selama terdapat koneksi internet.

Evolusi Model Computing

- ***Cloud computing*** adalah *next generation internet computing* dan *next generation data centers* hasil inovasi pengembangan dari teknologi *computing* sebelumnya seperti *grid computing*, *utility computing* dan *software as a services* dan lain-lain.
- ***Cloud computing*** menggunakan kombinasi teknologi processor baru berkecepatan tinggi, teknologi virtualiasi, *distributed storage*, *broadband internet access*, *automated management* serta server yang tidak terlalu mahal.

Evolusi Model Computing

Grid Computing

- Solving large problems with parallel computing
- Made mainstream by Globus Alliance



Utility Computing

- Offering computing resources as a metered service
- Introduced in late 1990s



Software as a Service

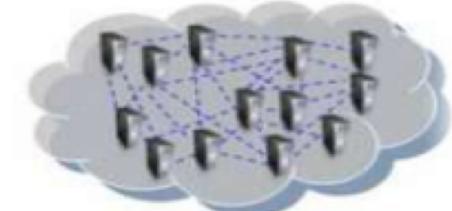
Network-based subscriptions to applications
Gained momentum in 2001



Cloud Computing

Next-Generation Internet computing

Next-Generation Data Centers



Hubungan Cloud Computing dan Model Computing lainnya

Cloud computing telah menjadi perbicangan hangat di dunia industri beberapa tahun terakhir ini, dan sedikit banyak selalu bersinggungan dengan istilah-istilah teknologi sebelumnya:

- *grid computing*
- *utility computing*
- *virtualization*
- *server cluster*
- *dedicated server*
- *collocation*

Hubungan Cloud Computing dan Model Computing lainnya

Infrastruktur *cloud computing* menggunakan teknologi virtualisasi yang dibangun berbasis *server cluster* dan memiliki relasi dengan *grid computing* dan *utility computing* yang digunakan untuk berkompetisi dengan *dedicated server* dan *collocation*.

Grid Computing

- **Definition**

- Combination of computer resources from multiple administrative domains applied to a common task.

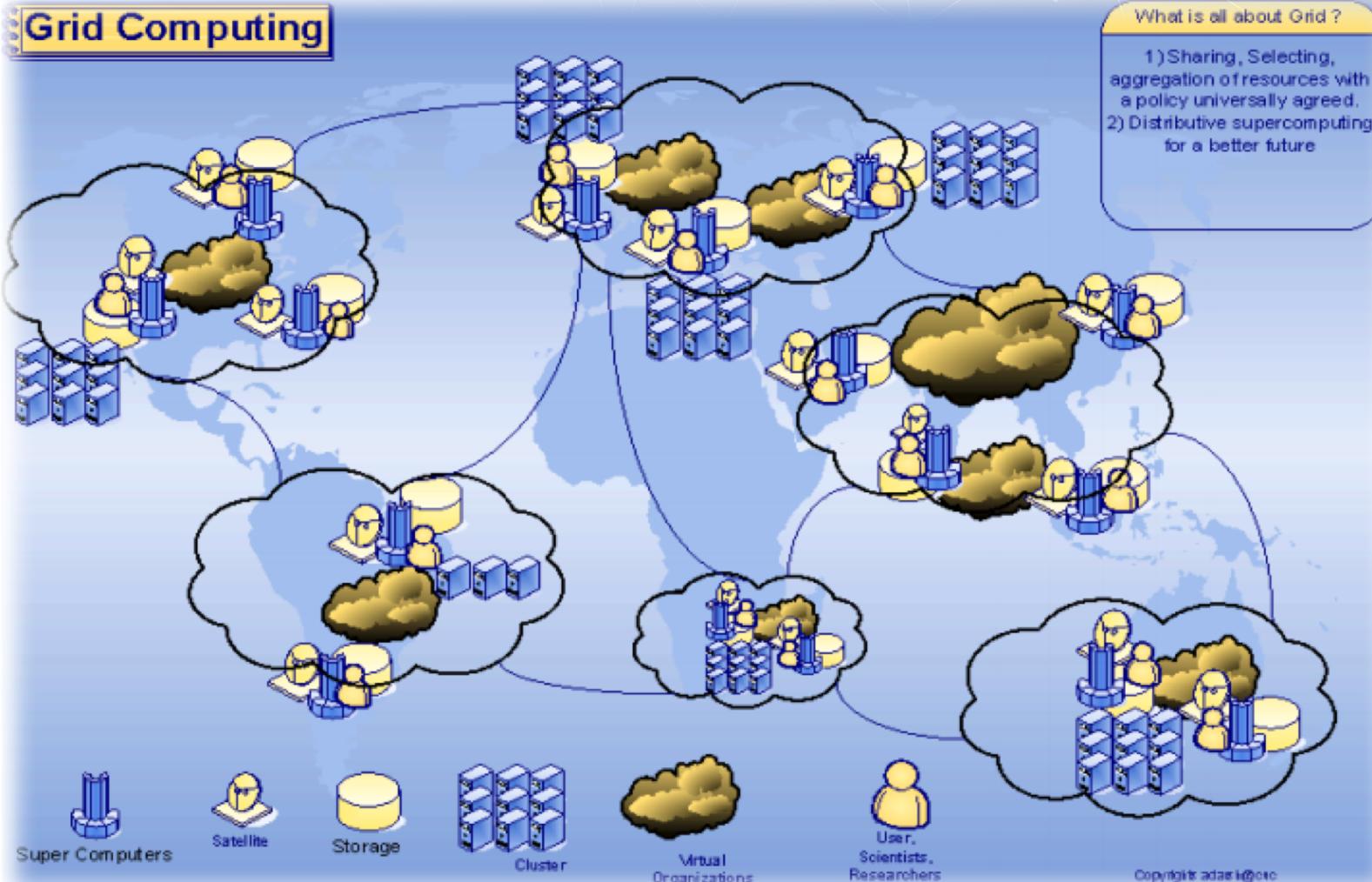
- **Core idea**

- Distributed parallel computation
- Super virtual computer

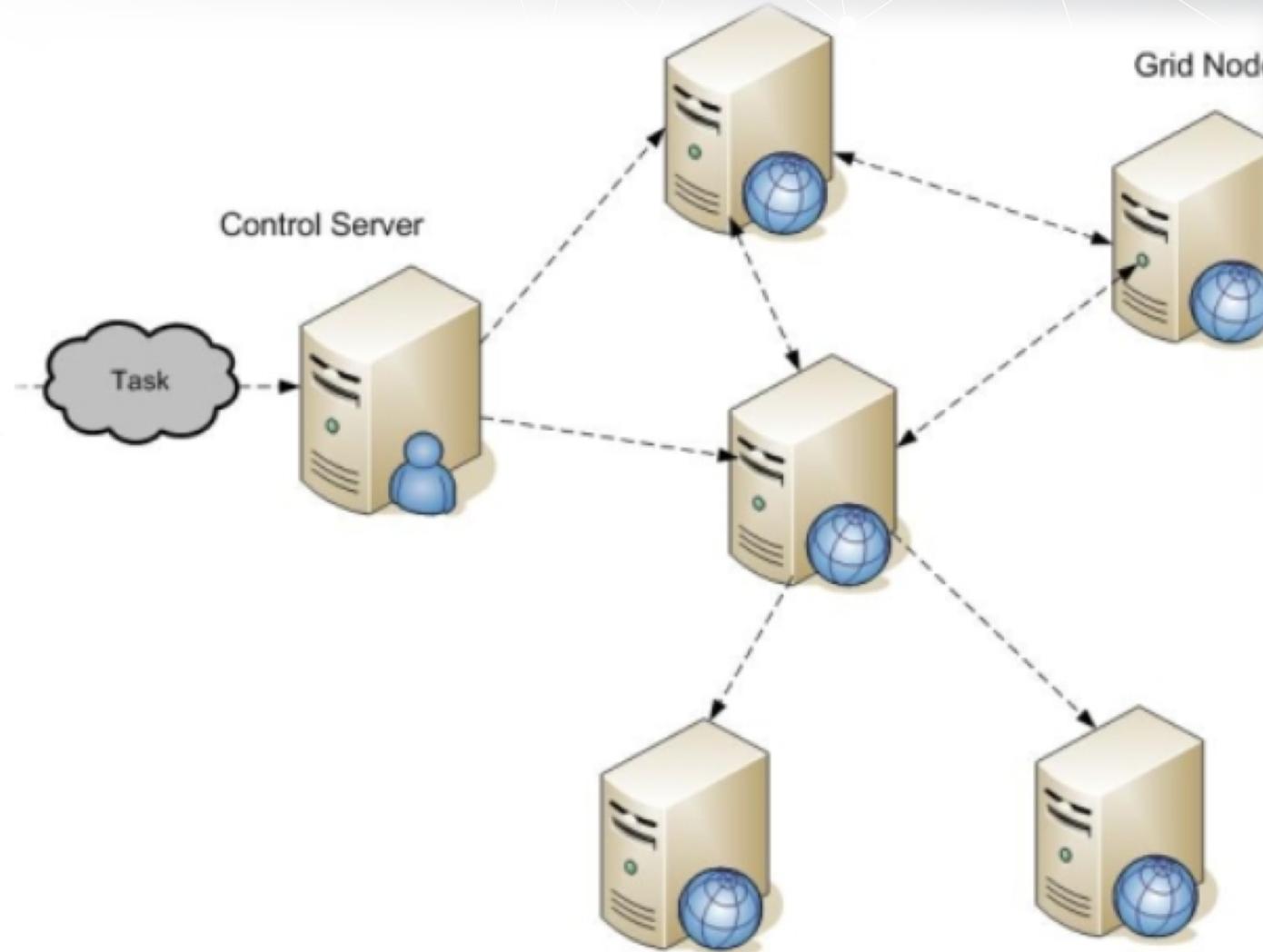
Grid Computing

- ***Grid computing*** adalah penggunaan sumber daya yang melibatkan banyak komputer yang letaknya terpisah secara geografis dan saling terhubung melalui jalur komunikasi untuk memecahkan persoalan komputasi skala besar.
- Dengan kata lain konsep ***grid computing*** adalah komputasi parallel dengan infrastruktur perangkat keras dan perangkat lunak yang dapat menyediakan akses yang bisa diandalkan, konsisten, tahan lama dan tidak mahal terhadap kemampuan komputasi mutakhir yang tersedia.

Grid Computing



Grid Computing



Perbedaan Cloud Computing dan Grid Computing

- ***Grid computing*** lebih menekankan pada 'resources sharing', setiap *node grid* dapat mengajukan permintaan sumber daya dari node lain, dan setiap node harus memberikan kontribusi sumber daya ke grid.
- Fokus ***grid computing*** adalah kepada kemampuan memindahkan beban kerja ke lokasi sumber daya yang memerlukan, yang sebagian besar terpencil dan sudah tersedia untuk digunakan.

Perbedaan Cloud Computing dan Grid Computing

- ***Cloud computing*** lebih menekankan pada kepemilikan (*proprietary*), setiap pengguna *cloud* bisa mendapatkan sumber daya pribadi dari *cloud*, yang disediakan oleh *service provider* tertentu dan pengguna tidak perlu berkontribusi dalam penyediaan sumber daya.
- *Computing resource* di dalam lingkungan *cloud* seperti server, dapat dibentuk secara dinamis dari *hardware* infrastruktur utamanya dan dibuat tersedia untuk suatu beban kerja.
- Selain '***cloud computing***' dapat mendukung grid, juga dapat mendukung lingkungan non-grid, seperti arsitektur *web three-tier* untuk menjalankan aplikasi tradisional atau aplikasi Web 2.0.

Perbedaan Cloud Computing dan Grid Computing

- ***Grid computing*** menekankan pada tugas-tugas komputasi yang sensitif dan sulit untuk yang berskala otomatis.
- ***Cloud computing*** menekankan pada aplikasi transaksi, suatu permintaan dengan jumlah besar dan terpisah-pisah, dan dapat digunakan untuk skala otomatis atau semi otomatis.

Perbedaan Cloud Computing dan Grid Computing

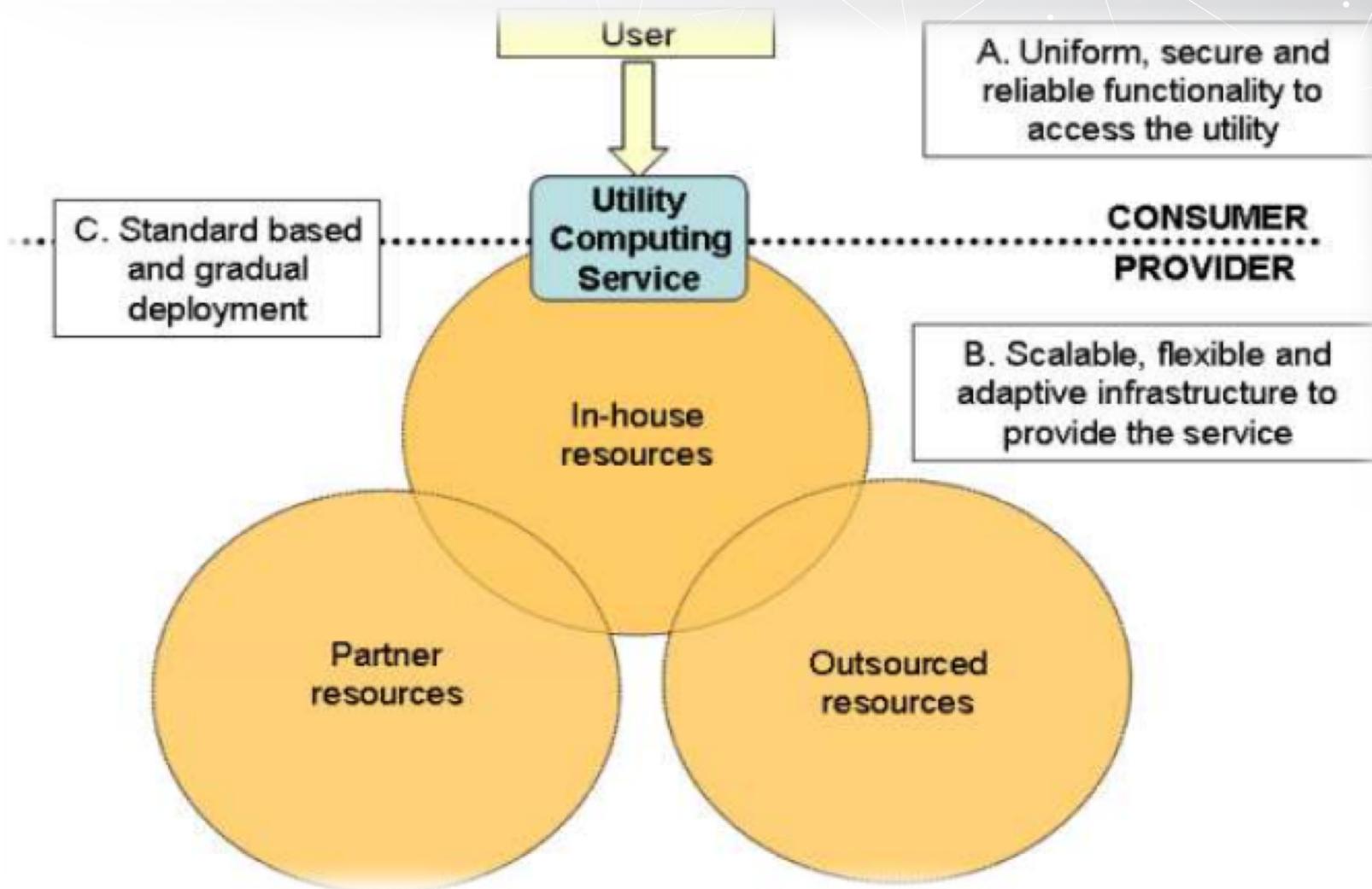
	EGEE Grid	Amazon Cloud
Target Group	Scientific community	Business
Service	short-lived batch-style processing (job execution)	long-lived services based on hardware virtualization
SLA	Local (between the EGEE project and the resource providers)	Global (between Amazon and users)
User Interface	High-level interfaces	HTTP(S), REST, SOAP, Java API, BitTorrent
Resource-side middleware	Open Source (Apache 2.0)	Proprietary
Ease of Use	Heavy	Light
Ease of Deployment	Heavy	Unknown
Resource Management	probably similar	
Funding Model	Publicly funded	Commercial

Summary of „An EGEE Comparative Study: Grids and Clouds Evolution or Revolution“ by Markus Klems

Utility Computing

- **Definition**
 - The packaging of computing resources (computation, storage etc.) as a metered service similar to a traditional public utility.
- **Observation**
 - Not a new concept
 - If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility.

Utility Computing



Perbandingan Utility Computing dan Cloud Computing

Cloud computing



Utility computing

Monitor
Meter
Billing
Pay



Perbandingan Utility Computing dan Cloud Computing

- **Utility computing** adalah suatu model bisnis penyediaan aplikasi sumber daya infrastruktur IT khususnya berkaitan dengan “*price model*”.
- **Cloud computing** adalah “*computing model*” berkaitan dengan cara kita mendisain, mengembangkan, menyebarkan/mendistribusikan dan menjalankan aplikasi yang dioperasikan dengan di suatu “*sharing resources*”.
- **Utility computing** sering menggunakan infrastuktur *cloud computing* meskipun tidak selalu.
- Berdasarkan kesamaan tersebut, kita dapat mengadopsi *utility computing* dan dapat mengadopsi *price model* dari model yang lain.

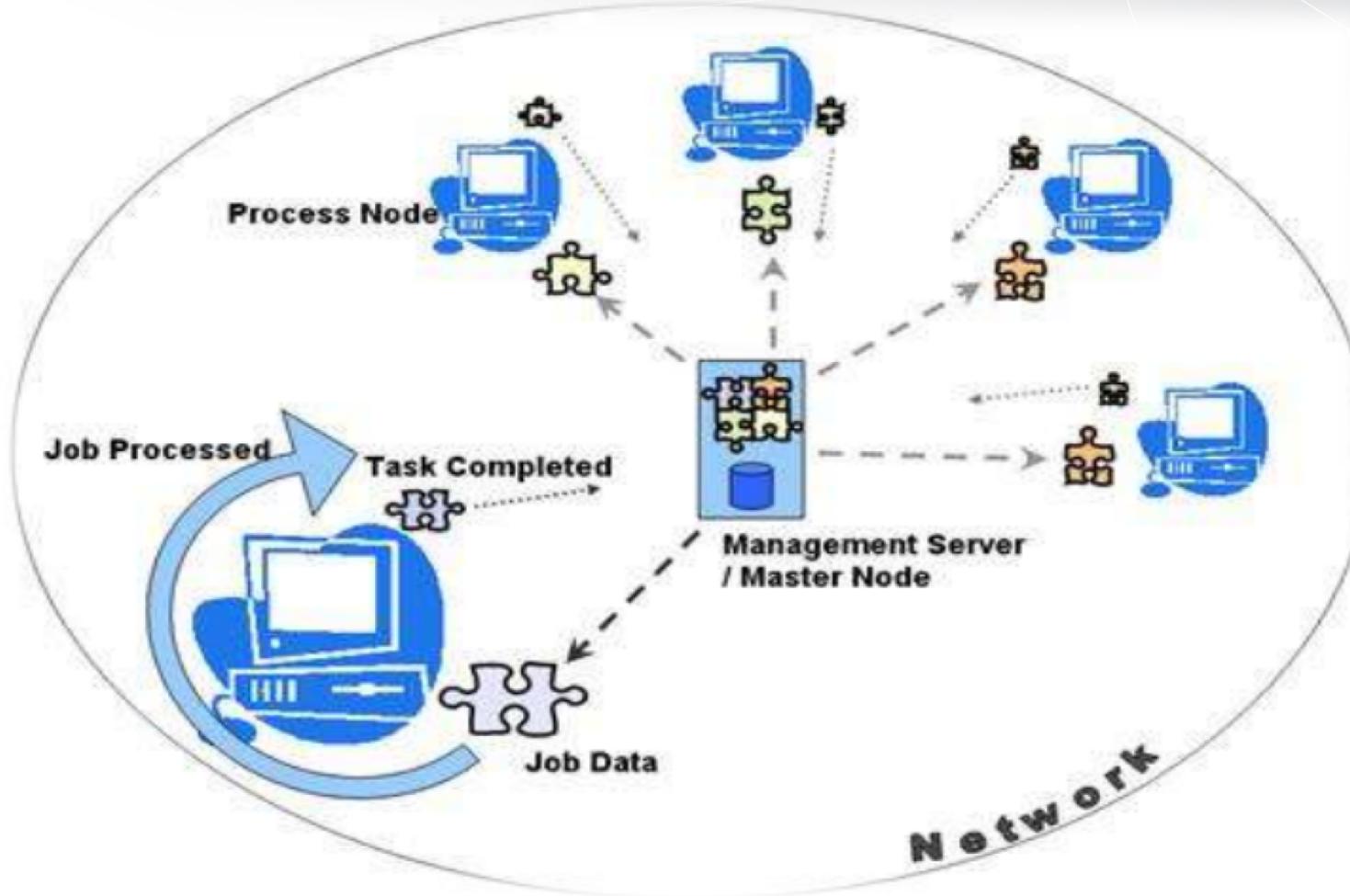
Distributed Computing

- ***Distributed computing*** berkaitan dengan sistem *hardware* dan *software* yang memiliki lebih dari satu elemen pemrosesan atau *storage element, concurrent process*, atau multiple program berjalan dibawah pengendalian yang ketat.
- Pada ***distributed computing***, suatu program dipecah ke beberapa bagian yang dijalankan secara bersamaan pada banyak komputer yang berkomunikasi melalui jaringan.

Distributed Computing

- ***Distributed computing*** adalah suatu bentuk dari *parallel computing*, dimana parallel computing umum digunakan untuk menggambarkan bagian-bagian program yang berjalan secara bersamaan di atas *multiple processor* di komputer yang sama.
- Kedua tipe *processing* (*distributed* dan *parallel computing*) membutuhkan pembagian program kepada bagian-bagian yang dapat berjalan serempak, tetapi *distributed computing* dapat berjalan secara simultan.

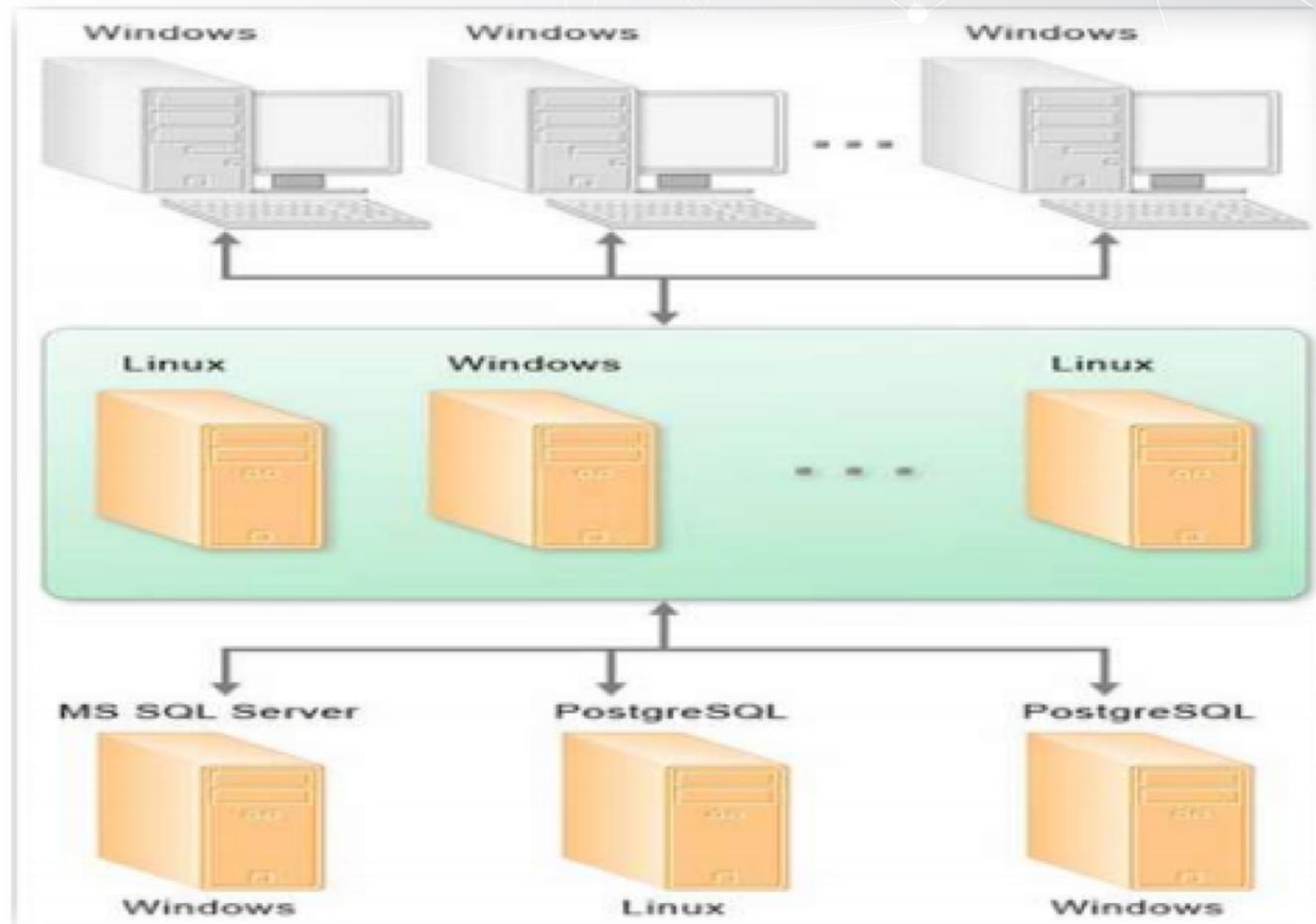
Prinsip Kerja Distributed Computing



Computer Cluster

- Komponen *cluster* biasanya LAN kecepatan tinggi (*fast ethernet*).
- *Cluster* bertujuan untuk meningkatkan performance dan/atau *availability* dibanding sebuah computer.

Computer cluster



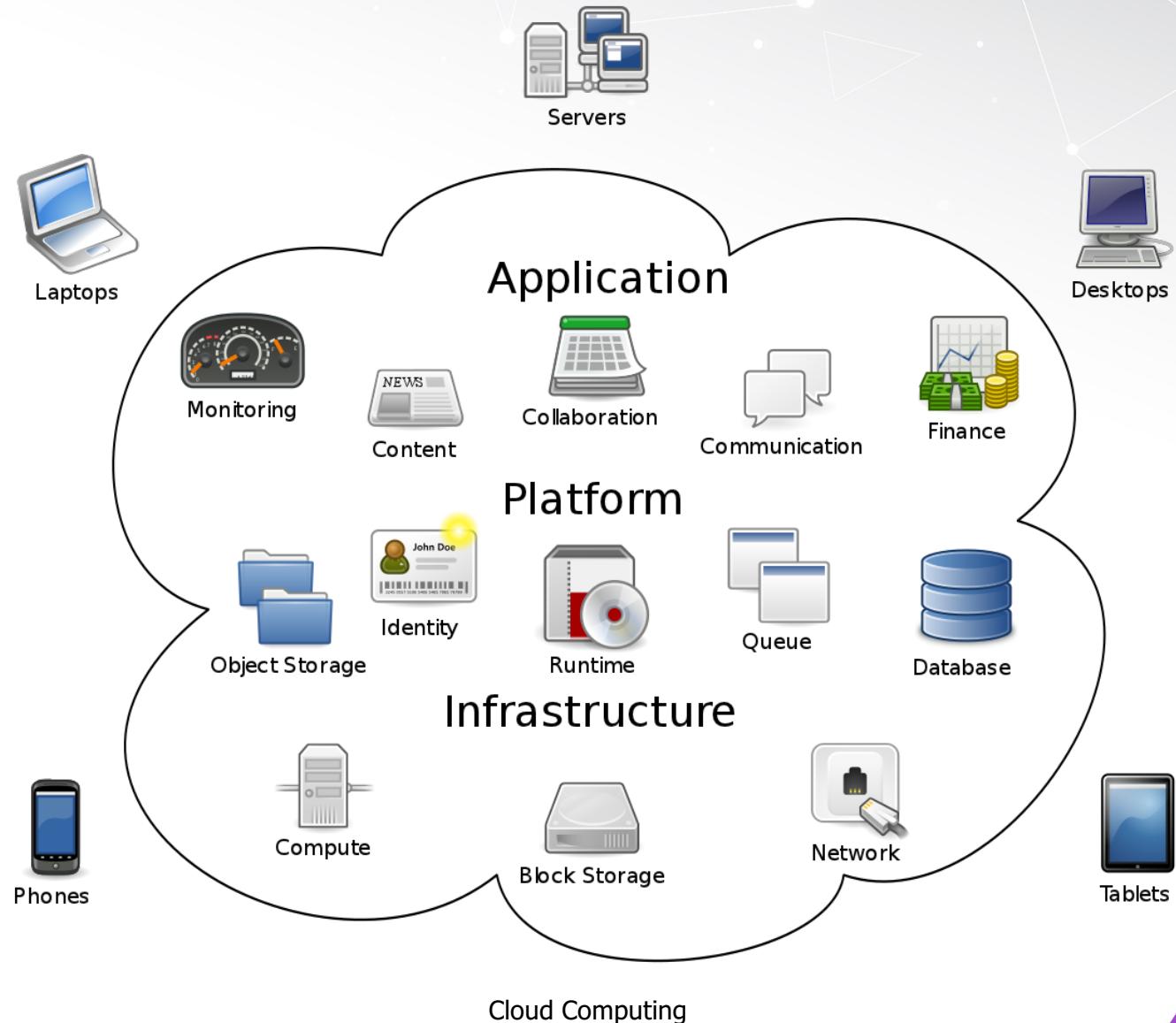
Cloud Computing

- **Is cloud computing?**
 - grid computing + utility computing ??
 - difficult to define - means different things to different parties
- **Various definitions**
 - NIST – National Institute of Standards and Technology - “universally” accepted definition
 - “Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services)”.

Cloud Definition

Cloud computing is a set of service-oriented architectures, which allow users to access a number of resources in a way that is elastic, cost-efficient, and on-demand.

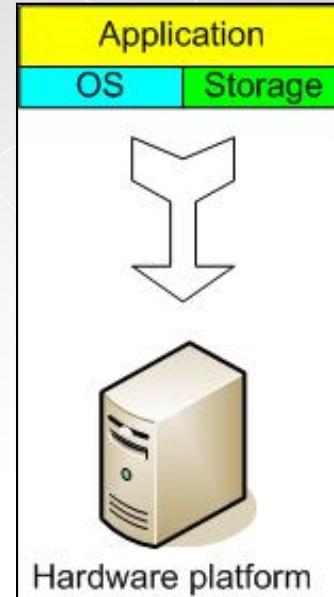
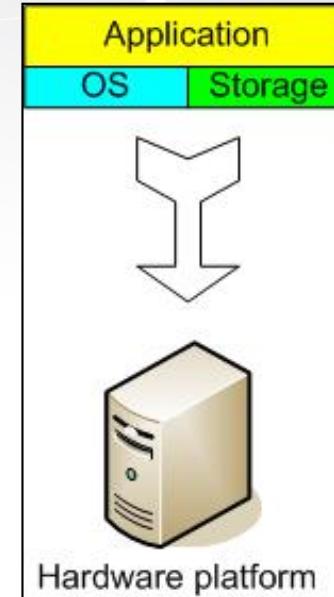
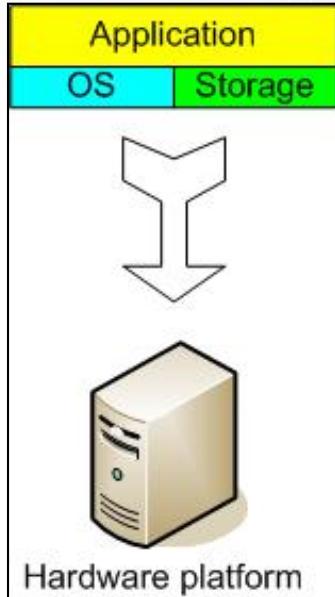
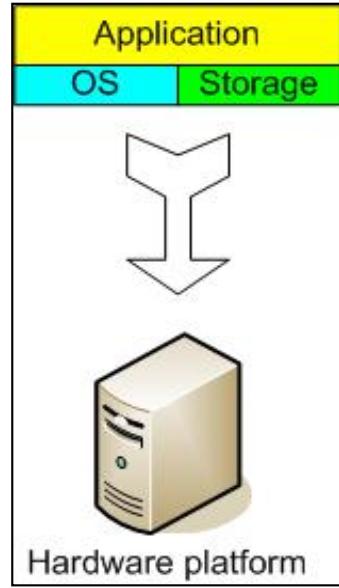
Cloud Computing Overview



The Traditional Server Concept

- Shared hosting
- Dedicated server
- VPS
- Cloud

The Traditional Server Concept

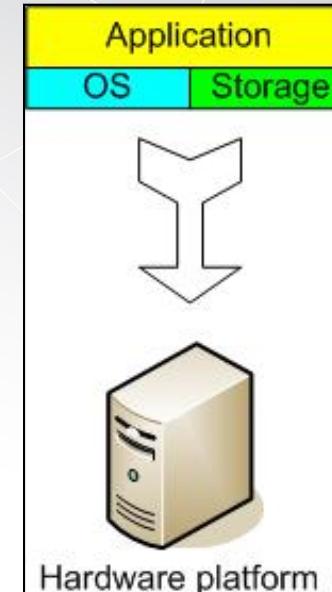
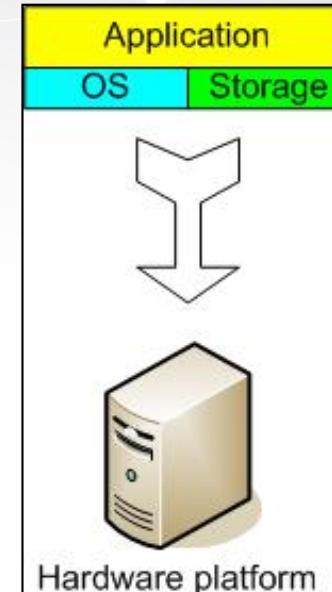
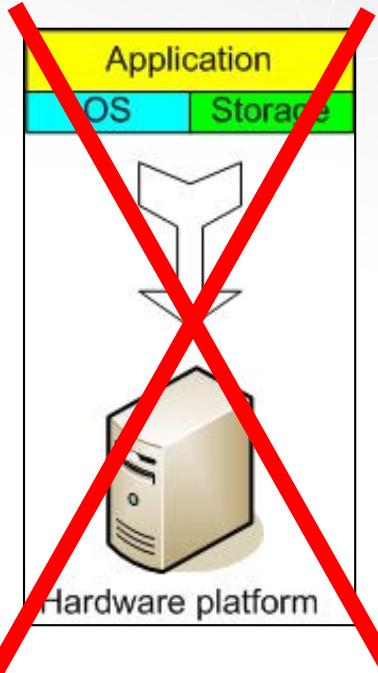
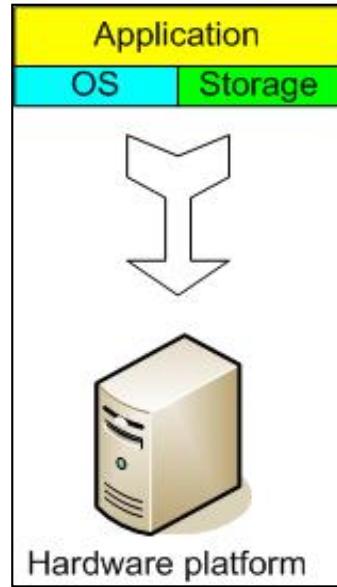


- Web Server
- App Server
- DB Server
- EMail
- Windows
- Linux
- MySQL
- Glassfish
- IIS

The Traditional Server Concept

- System administrators often talk about servers as a whole unit that includes the hardware, the OS, the storage, and the applications.
- Servers are often referred to by their function i.e. the exchange server, the SQL server, the file server, etc.
- If the file server fills up, or the exchange server becomes overtaxed, then the system administrators must add in a new server.

The Traditional Server Concept



- Web Server
- App Server
- Windows
- DOWN!
- IIS
- DB Server
- Linux
- MySQL
- EMail
- Windows
- Exchange

The Traditional Server Concept

- Unless there are multiple servers, if a service experiences a hardware failure, then the service is down.
- System admins can implement clusters of servers to make them more fault tolerant.
- However, even clusters have limits on their scalability, and not all applications work in a clustered environment.

The Traditional Server Concept

Pros

- Easy to conceptualize
- Fairly easy to deploy
- Easy to backup
- Virtually any application/service can be run from this type of setup

The Traditional Server Concept

Cons

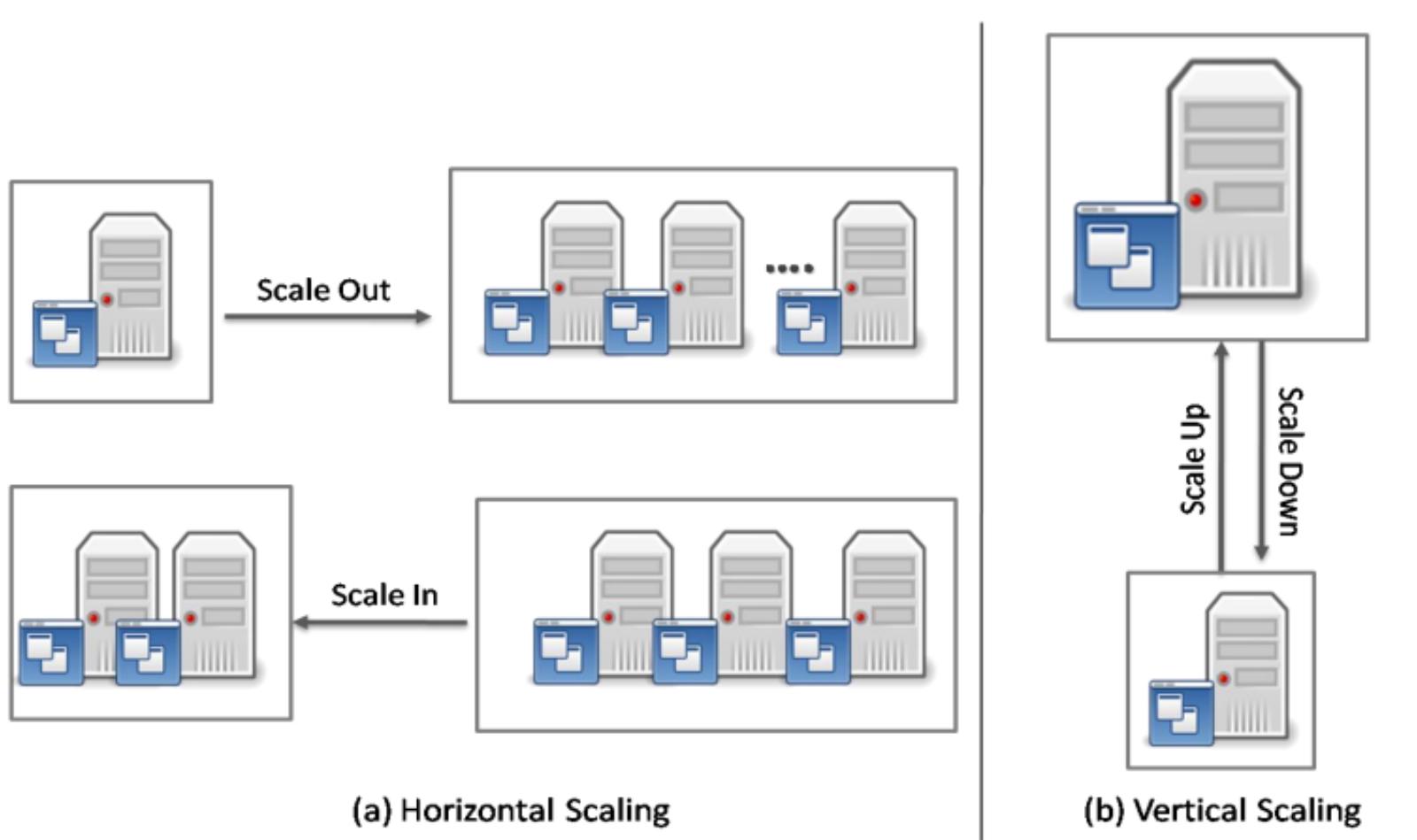
- Expensive to acquire and maintain hardware
- Not very scalable
- Difficult to replicate
- Redundancy is difficult to implement
- Vulnerable to hardware outages
- In many cases, processor is under-utilized

Cloud computing

Computing at scale:

- The need for scalability; scale of current services
- Scaling up: From PCs to data centers
- Problems with 'classical' scaling techniques

Types of Scaling – Scaling Architecture



Horizontal vs. Vertical Scaling

Horizontal (Scale-out and Scale-in)

- More computing resources (e.g., servers)
- Reliable – fail-over scenario
- Fully automated
- Growing management complexity

Horizontal vs. Vertical Scaling

Vertical (Scale-up and Scale-down)

- More powerful computing resources - bigger servers
- Single point of failure
- Human intervention
- Reasonable management overhead

Horizontal vs. Vertical Scaling

Table 6 Potential elasticity strategies for MyShop application's tier (weekend workload)

Scaling strategy	Server scaling costs	Monitoring costs	Application's availability and reliability
Horizontal scaling	$24 \text{ hrs/w} \times \$0.085/\text{hr} \times 6 \text{ servers} \times 52 \text{ weeks} = \$636.48/\text{yr}$	Defining and configuring 7 metrics for 6 servers Costs: $\$3.5 \text{ per server/mo} \times 6 \text{ servers} \times 12 \text{ months} = \252	Highly available—no single point of failure Highly likely reliable—quick recovery time
Vertical scaling	$24 \text{ hrs/w} \times \$0.68 \times 1 \text{ server} \times 52 \text{ weeks} = \848.64	Defining and configuring 7 metrics for 1 server Costs: $\$3.5 \text{ per server/mo} \times 1 \text{ server} \times 12 \text{ months} = \42	Low availability—single point of failure Highly likely unreliable—long recovery time
Hybrid scaling	$(24 \text{ hrs/w} \times \$0.085/\text{hr} \times 3 \text{ servers} \times 52 \text{ weeks}) + (24 \text{ hrs/w} \times \$0.34 \times 1 \text{ server} \times 52 \text{ weeks}) = \742.48	Defining and configuring 7 metrics for 4 servers Costs: $\$3.5 \text{ per server/mo} \times 4 \text{ servers} \times 12 \text{ months} = \168	Improved availability—no single point of failure Improved reliability—medium recovery time

IaaS Elasticity (Auto-Scaling) Services

- AWS Elastic Load Balancing
- GoGrid's Infrastructure and RAM Scaling
- Rackspace Cloud Monitoring and AWS CloudWatch
- AWS and Rackspace Auto Scaling
- RightScale cloud management platform

Elasticity Rules - Examples

IaaS Service	Auto Scaling Rules
Amazon Web Services (AWS)	If the application latency seen by the load balancer is greater than 800 millisecond for 9 minutes continuously, scale out the application tier
Rackspace Auto Scale	If average memory utilization across all web servers is greater than 85%, then scale out the web servers

Elasticity (Auto-Scaling) Rules

Rule - based mechanism

- Monitor certain resources/application metrics
- Determine when to trigger adding releasing computing resources
- Determine how much computing resources to add/release
- Choose appropriate values for the core thresholds and parameters

Structure of Elasticity Rules

Monitor <*MetricName*> every <*T*>

} Metric Monitoring

// Evaluate Elasticity Condition (EC) every <*T_i*>

IF <*MetricName*> {Comparison Operator} <*M_θ*> FOR <*T_w*>

} Elasticity Condition

// Execute Elasticity Action (EA) when EC is satisfied

Change <*ResourceName*> by <*P*>

} Elasticity Action

Wait for <*T_c*>

Auto-scaling Rules – Example

Monitor CPU Utilization (CPUUtil) every 1 min. interval

IF CPUUtil > 80% FOR 7 minutes

Add 1 server of small capacity //Scale-out

Wait 5 consecutive 1 min. intervals

Auto-scaling Rules – Example

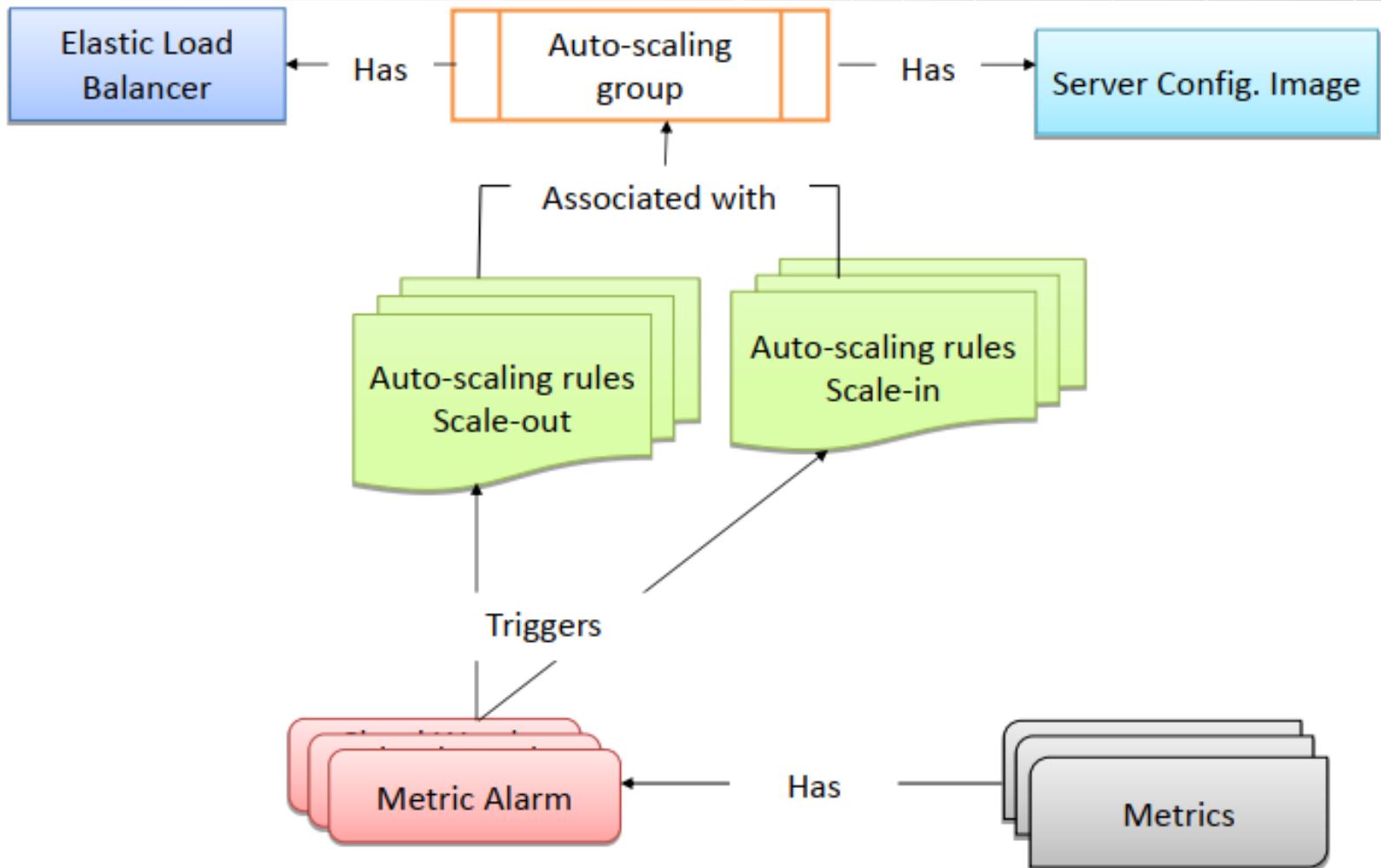
Monitor CPU Utilization (CPUUtil) every 1 min. interval

IF CPUUtil < 30% FOR 10 minutes

Remove 1 server of small capacity //Scale-in

Wait 7 consecutive 1 min. interval

Elasticity (Auto-Scaling)





IKUTI KAMI



- digitalent.kominfo
- digitalent.kominfo
- DTS_kominfo
- Digital Talent Scholarship 2019

Pusat Pengembangan Profesi dan Sertifikasi
Badan Penelitian dan Pengembangan SDM
Kementerian Komunikasi dan Informatika
Jl. Medan Merdeka Barat No. 9
(Gd. Belakang Lt. 4 - 5)
Jakarta Pusat, 10110

