UNIVERSIDADE
BEIRA INTERIOR

Architectural Design of Compute and Storage Clouds

Tecnologias de Virtualização e Centros de Dados
Mestrado em Engenharia Informática

Alexandre Fonte (Prof. Convidado)
Departamento de Informática

Ano Letivo 2023/2024

Última revisão:
6 maio de 2024

Credits

- These slides are partly based on the book:

  Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Kai Hwang, Jack Dongarra, Geoffrey C. Fox (Authors), Morgan Kaufmann, 1st edition, 2011, ISBN-13: 978-0123858801, 672 pages.

Agenda

- A Generic Cloud Architecture Design
- Layered Cloud Architectural Development
- Virtualization Support and Disaster Recovery
- Architectural Design Challenges
- Public Cloud Platforms: GCP, AWS and Azure

A Generic Cloud Architecture Design

- An Internet cloud is envisioned as a **public cluster of servers** provisioned on demand to perform collective web services or distributed applications using data-center resources.
- Cloud management receives the user request, finds the correct resources, and then calls the provisioning services which invoke the resources in the cloud. The cloud management software needs to support both physical and virtual machines.
- The platform needs to establish a very large-scale HPC (High performance computing) infrastructure. The hardware and software systems are combined to make it easy and efficient to operate.
- **Scalability, virtualization, efficiency, and reliability and in addition Security** are major design goals of a cloud computing platform.

A Generic Cloud Architecture Design - ***Cloud Platform Design Goals***

- Cloud Platform Design Goals
  - **Virtualization (slide anterior)** *(A virtualização é fundamental para a flexibilidade e eficiência das plataformas de computação em nuvem).*
  - **Efficiency (slide anterior)** *(Capacidade da Plataforma fornecer recursos de forma eficiente + a custos razoáveis para os utilizadores)*
  - **System scalability** can benefit from cluster architecture. If one service takes a lot of processing power, storage capacity, or network traffic, it is simple to add more servers and bandwidth. The scale of the cloud architecture can be easily expanded by adding more servers and enlarging the network connectivity accordingly.

(*Capacidade da Plataforma dimensionar horizontalmente (adicionar mais servidores e verticalmente (adicionar mais recursos a um servidor) para lidar com as cargas de trabalho ou necessidades dos utilizadores)*

  - **System reliability** can also benefit from this architecture. Data can be put into multiple locations. If one of the data centers crashes, the user data is still accessible.

A Generic Cloud Architecture Design - ***Cloud Platform Design Goals***

- Cloud Platform Design Goals (adicionais)
  - **+ Disponibilidade:** A capacidade da plataforma de estar sempre disponível e funcionando, garantindo que os utilizadores possam aceder aos seus recursos e aplicações a qualquer momento.
  - **+ Security Compliance**: Security in shared resources and shared access of data centers also pose another design challenge.

  (Capacidade da plataforma de proteger os dados e aplicações dos utilizadores contra ameaças internas e externas, garantindo que apenas utilizadores autorizados possam aceder aos dados/informações confidenciais).

A Generic Cloud Architecture Design - *Enabling Technologies for Clouds*

- The key driving forces behind cloud computing are the **ubiquity of broadband and wireless networking, falling storage costs, and progressive improvements in Internet computing software**.
- Cloud users are able to demand more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, **whereas service providers can increase system utilization** via **multiplexing**, **virtualization**, and **dynamic resource provisioning**.
- Many technologies play instrumental roles in making cloud computing a reality.

A Generic Cloud Architecture Design - *Enabling Technologies for Clouds*

- **In the hardware area, the rapid progress** in multicore CPUs, memory chips, and disk arrays has made it possible to build faster data centers with huge amounts of storage space.
- **Resource virtualization** enables rapid cloud deployment and disaster recovery. Service-oriented architecture (SOA) also plays a vital role.
- **Progress in providing SaaS, Web 2.0 standards, and Internet performance have all contributed to the emergence of cloud services**.
- Today's clouds are designed to serve a **large number of tenants** over massive volumes of data. The availability of large-scale, distributed storage systems is the foundation of today's data centers.
- Cloud computing is greatly benefitted by **the progress made in license management and automatic billing** techniques in recent years.

A Generic Cloud Architecture Design - *Enabling Technologies for Clouds*

- Clouds are enabled by the progress in hardware, software, and networking technologies summarized in Table 4.3.

Table 4.3 Cloud-Enabling Technologies in Hardware, Software, and Networking

Technology	Requirements and Benefits
Fast platform deployment	Fast, efficient, and flexible deployment of cloud resources to provide dynamic computing environment to users
Virtual clusters on demand	Virtualized cluster of VMs provisioned to satisfy user demand and virtual cluster reconfigured as workload changes
Multitenant techniques	SaaS for distributing software to a large number of users for their simultaneous use and resource sharing if so desired
Massive data processing	Internet search and web services which often require massive data processing, especially to support personalized services
Web-scale communication	Support for e-commerce, distance education, telemedicine, social networking, digital government, and digital entertainment applications
Distributed storage	Large-scale storage of personal records and public archive information which demands distributed storage over the clouds
Licensing and billing services	License management and billing services which greatly benefit all types of cloud services in utility computing

A Generic Cloud Architecture Design - *A Generic Cloud Architecture*

- Figure 4.14 shows a security-aware cloud architecture.

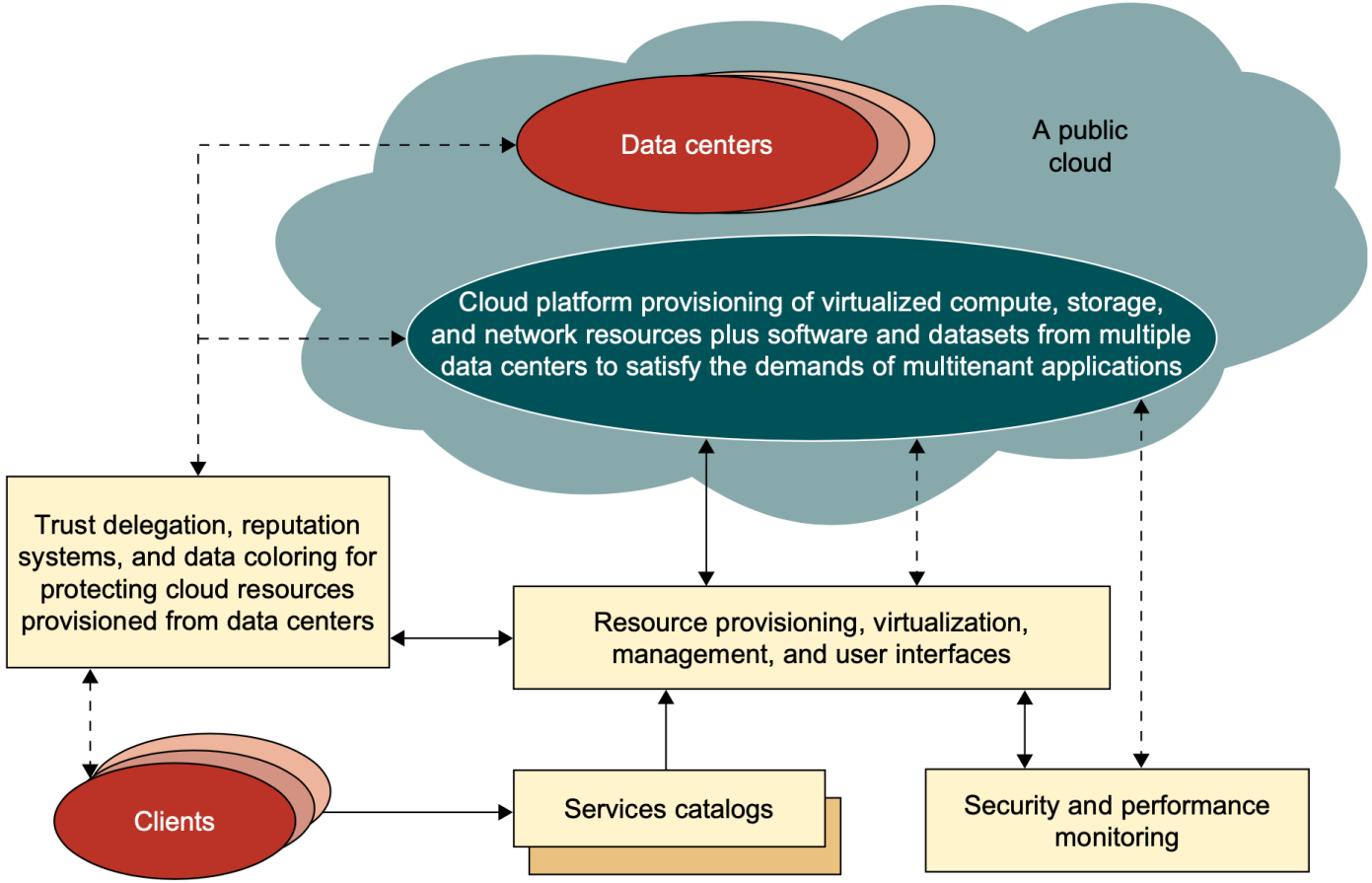


FIGURE 4.14

A security-aware cloud platform built with a virtual cluster of VMs, storage, and networking resources over the data-center servers operated by providers.

(*Courtesy of K. Hwang and D. Li, 2010 [36]*)

A Generic Cloud Architecture Design
- *A Generic Cloud Architecture*

- **The Internet cloud is envisioned as a massive cluster of servers.** These servers are provisioned on demand to perform collective web services or distributed applications using data-center resources.
- The cloud platform is formed dynamically by provisioning or deprovisioning servers, software, and database resources. **Servers in the cloud can be physical machines or VMs**.
- **User interfaces** are applied to request services. **The provisioning tool carves out the cloud system to deliver the requested service.**
- In addition to building the server cluster, the cloud platform demands distributed storage and accompanying services. **The cloud computing resources are built into the data centers,** which are typically owned and operated by a third-party provider.
- Consumers do not need to know the underlying technologies. In a cloud, software becomes a service.

A Generic Cloud Architecture Design - *A Generic Cloud Architecture*

- **The cloud demands a high degree of trust** of massive amounts of data retrieved from large data centers.
  - We need to build a framework to process large-scale data stored in the storage system.
  - This demands a distributed file system over the database system. Other cloud resources are added into a cloud platform, including storage area networks (SANs), database systems, firewalls, and security devices. Web service providers offer special APIs that enable developers to exploit Internet clouds.
- **Monitoring and metering units** are used to track the usage and performance of provisioned resources.
- **The software infrastructure of a cloud platform must handle all resource management and do most of the maintenance automatically.**
- Software must detect the status of each node server joining and leaving, and perform relevant tasks accordingly.

A Generic Cloud Architecture Design - *A Generic Cloud Architecture*

- Cloud computing providers, such as Google and Microsoft, have built a large number of data centers all over the world.
- Each data center may have thousands of servers. **The location of the data center is chosen to reduce power and cooling costs. Thus, the data centers are often built around hydroelectric power.**
- **The cloud physical platform builder is more concerned about the performance/price ratio and reliability issues than shear speed performance.**
- In general, private clouds are easier to manage, and public clouds are easier to access. The trends in cloud development are that more and more clouds will be hybrid. This is because many cloud applications must go beyond the boundary of an intranet.
- **Security becomes a critical issue** in safeguarding the operation of all cloud types.

Layered Cloud Architectural Development

- The architecture of a cloud is developed at **three layers:**

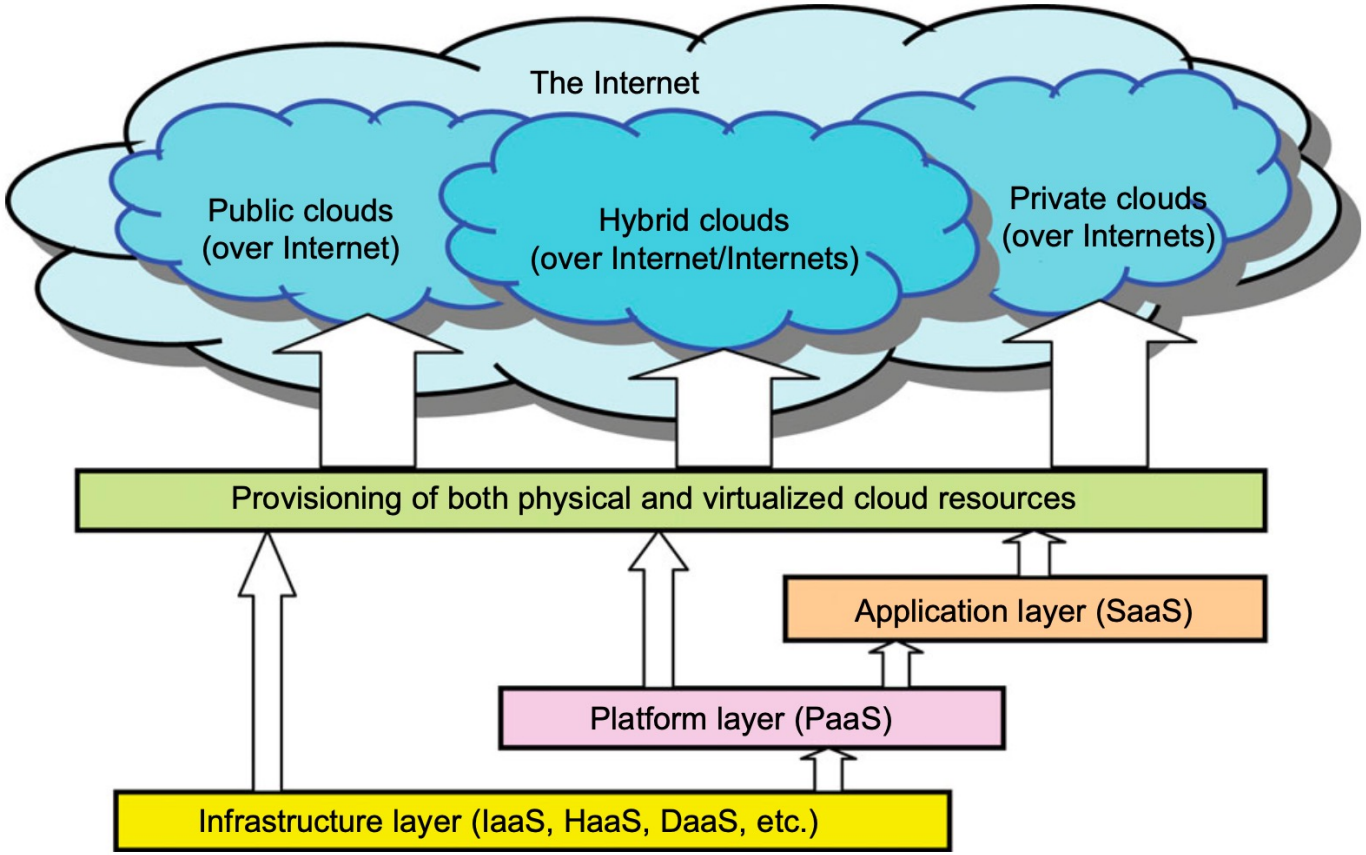


FIGURE 4.15

Layered architectural development of the cloud platform for IaaS, PaaS, and SaaS applications over the Internet.

Layered Cloud Architectural Development

- The architecture of a cloud is developed at **three layers: infrastructure, platform, and application.**
- These three development layers are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud.
- The services to public, private, and hybrid clouds are conveyed to users through networking support over the Internet and intranets involved.
- The **infrastructure layer** is deployed first to support IaaS services. This infrastructure layer serves as the foundation for building the **platform layer** of the cloud for supporting PaaS services.
- In turn, the **platform layer** is a foundation for implementing the **application layer** for SaaS applications.

Layered Cloud Architectural Development

- The **infrastructure layer** is built with virtualized compute, storage, and network resources. The abstraction of these hardware resources is meant to provide the flexibility demanded by users. Internally, virtualization realizes automated provisioning of resources and optimizes the infrastructure management process.
- The **platform layer** is for general-purpose and repeated usage of the collection of software resources. This layer **provides** users with **an environment to develop their applications**, to test operation flows, and to monitor execution results and performance.
- The platform should be able to assure users that they have scalability, dependability, and security protection. **In a way, the virtualized cloud platform serves as a "system middleware" between the infrastructure and application layers of the cloud.**

Layered Cloud Architectural Development

- The **application layer** is formed with a collection of all needed software modules for SaaS applications.
  - **Service applications** in this layer include daily office management work, such as information retrieval, document processing, and calendar and authentication services.
  - **Application layer** is also heavily used by enterprises in business marketing and sales, consumer relationship management (CRM), financial transactions, and supply chain management.
- It should be noted that not all cloud services are restricted to a single layer. Many applications may apply resources at mixed layers. After all, the three layers are built from the bottom up with a dependence relationship.
- **From the provider's perspective**, the services at various layers demand different amounts of functionality support and resource management by providers. **In general, SaaS demands the most work from the provider, PaaS is in the middle, and IaaS demands the least.**

Market-Oriented Cloud Architecture

- As consumers rely on cloud providers to meet more of their computing needs, they will require a specific level of QoS to be maintained by their providers. Cloud providers consider and meet the different QoS parameters of each individual consumer as negotiated in specific SLAs.
- To achieve this, (…) market-oriented resource management is necessary to regulate the supply and demand of cloud resources to achieve market equilibrium between supply and demand.
- The **SLA resource allocator** acts as the interface between the data center/cloud service provider and external users/brokers. It requires the interaction of the following mechanisms to support SLA-oriented resource management.
  - When a **service request** is first submitted the service request examiner interprets the submitted request for QoS requirements before determining whether to accept or reject the request.
  - The **request examiner** ensures that there is no overloading of resources whereby many service requests cannot be fulfilled successfully due to limited resources.

Market-Oriented Cloud Architecture

- **Pricing** serves as a basis for managing the supply and demand of computing resources within the data center and facilitates in prioritizing resource allocations effectively.
- **The Accounting mechanism** maintains the actual usage of resources by requests so that the final cost can be computed and charged to users. In addition, the maintained historical usage information can be utilized by the Service Request Examiner and Admission Control mechanism to improve resource allocation decisions.
- **The VM Monitor mechanism** keeps track of the availability of VMs and their resource entitlements.
- The **Dispatcher mechanism** starts the execution of accepted service requests on allocated VMs. The Service Request Monitor mechanism keeps track of the execution progress of service requests.

Market-Oriented Cloud Architecture

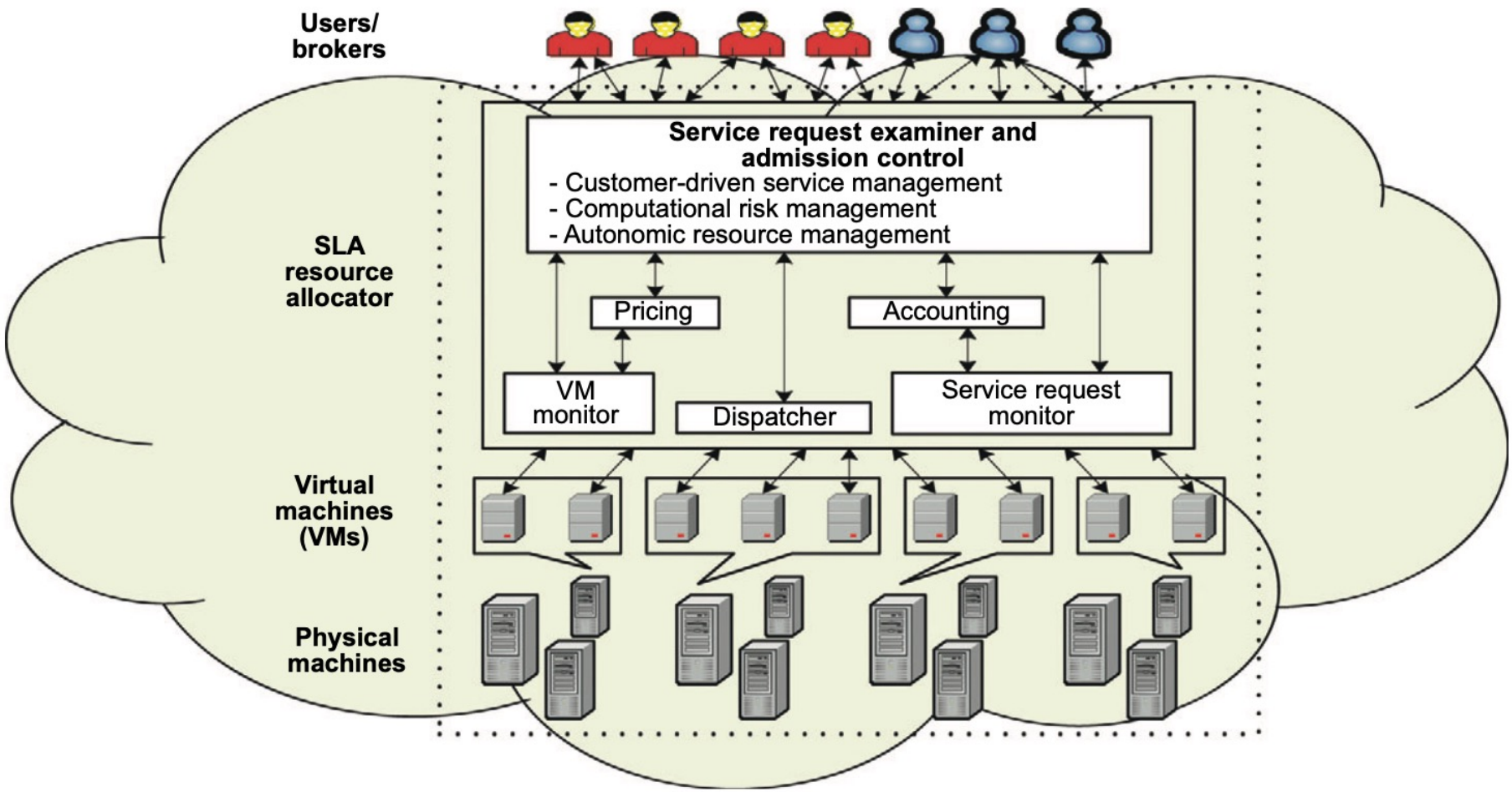


FIGURE 4.16

Market-oriented cloud architecture to expand/shrink leasing of resources with variation in QoS/demand from users.

(*Courtesy of Raj Buyya, et al. [11]*)

Virtualization Support and Disaster Recovery

- One very distinguishing feature of cloud computing infrastructure is the use of system virtualization and the modification to provisioning tools.
  - **Virtualization of servers on a shared cluster can consolidate web services.** As the VMs are the containers of cloud services, the provisioning tools will first find the corresponding physical machines and deploy the VMs to those nodes before scheduling the service to run on the virtual nodes.
  - **In addition, in cloud computing, virtualization also means the resources and fundamental infrastructure are virtualized.** The user will not care about the computing resources that are used for providing the services. Cloud users do not need to know and have no way to discover physical resources that are involved while processing a service request.
  - Also, **application developers** do not care about some infrastructure issues such as scalability and fault tolerance (i.e., they are virtualized). Application developers focus on service logic.

Virtualization Support and DR

- Figure 4.17 shows the infrastructure needed to virtualize the servers in a data center for implementing specific cloud applications.

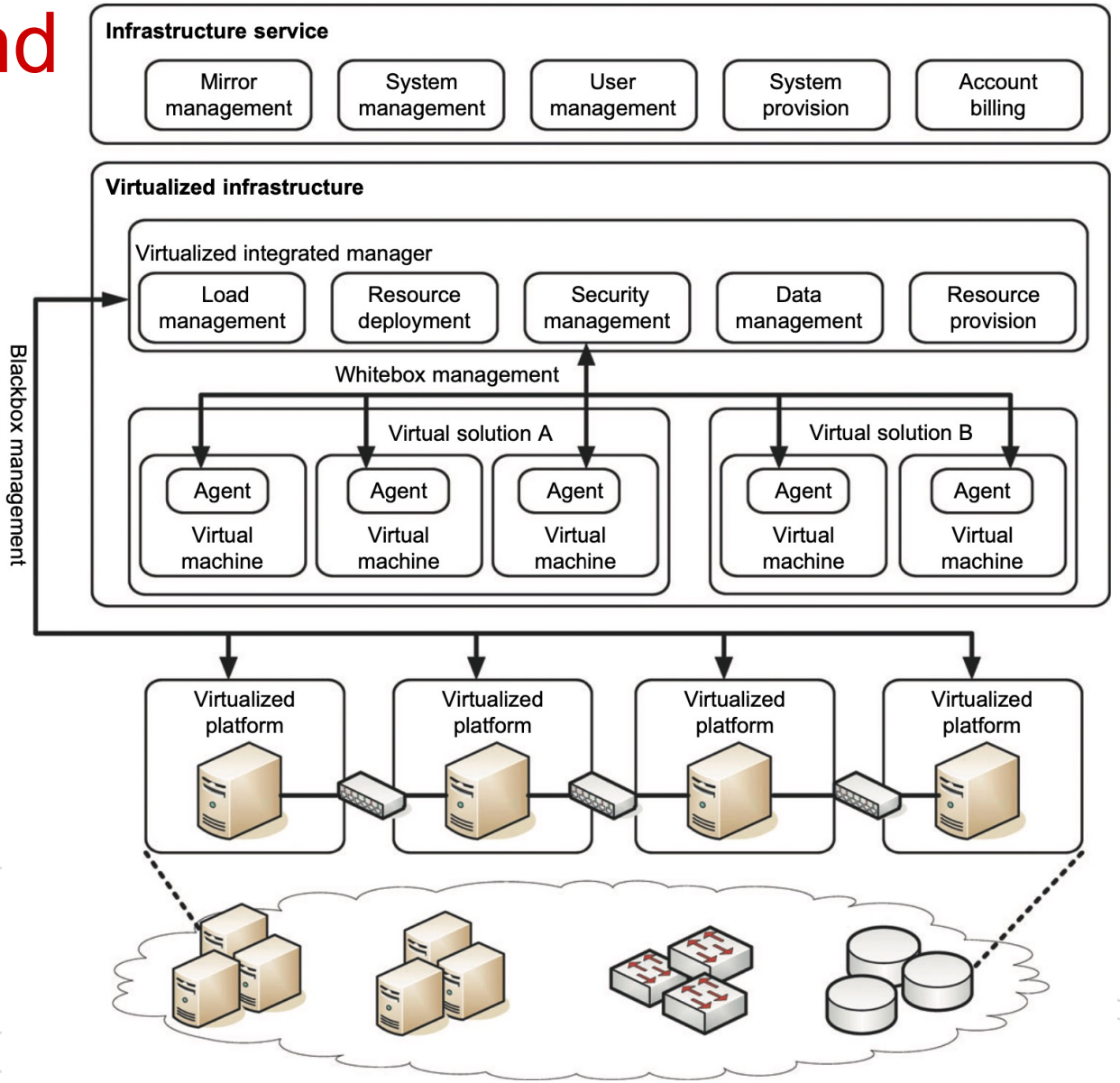


FIGURE 4.17

Virtualized servers, storage, and network for cloud platform construction.

(*Courtesy of Zhong-Yuan Qin, SouthEast University, China*)

Virtualization Support and Disaster Recovery – HW Virtualization

- In cloud computing systems, virtualization software is used to virtualize the hardware.
- **Cloud computing systems use virtualization software as the running environment for legacy software such as old OS and unusual applications.**
- **Virtualization software is also used as the platform for developing new cloud applications** that enable developers to use any operating systems and programming environments they like. **The development environment and deployment environment can now be the same**, which eliminates some runtime problems.
- Using VMs in a cloud computing platform ensures extreme flexibility for users. As the computing resources are shared by many users, a method is required to maximize the users' privileges and still keep them separated safely.

Virtualization Support and Disaster Recovery – **Virtualization for IaaS**

- VM technology has increased in ubiquity. This has enabled users to create customized environments atop physical infrastructure for cloud computing.
- Use of VMs in clouds has the following distinct benefits:
  - (1) System administrators **consolidate workloads** of underutilized servers in fewer servers;
  - (2) VMs have the ability to **run legacy code** without interfering with other APIs;
  - (3) VMs can be used to **improve security** through creation of sandboxes for running applications with questionable reliability;
  - And (4) virtualized cloud platforms **can apply performance isolation, letting providers offer some guarantees and better QoS to customer applications**.

Virtualization Support and DR – **VM Cloning for Disaster Recovery**

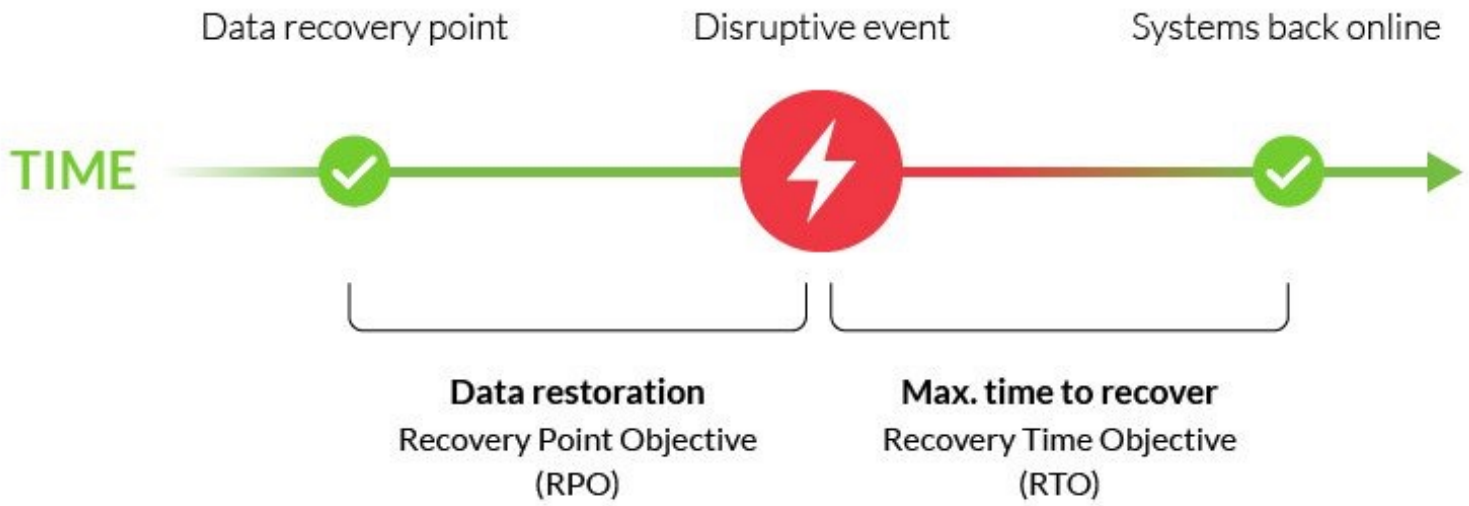
- **VM technology requires an advanced disaster recovery scheme.**
- One scheme is to recover one physical machine by another physical machine.
- **The second scheme is to recover one VM by another VM**.
- Traditional disaster recovery from one physical machine to another is rather slow, complex, and expensive. Total recovery time is attributed to the hardware configuration, installing and configuring the OS, installing the backup agents, and the long time to restart the physical machine.
- **To recover a VM platform, the installation and configuration times for the OS and backup agents are eliminated.** Therefore, we end up with a much shorter disaster recovery time, about 40 percent of that to recover the physical machines. Virtualization aids in fast disaster recovery by VM encapsulation.

Virtualization Support and DR – **VM Cloning for Disaster Recovery**

- **The cloning of VMs offers an effective solution.** The idea is to make a clone VM on a remote server for every running VM on a local server.
- Among all the clone VMs, only one needs to be active. **The remote VM should be in a suspended mode**. A cloud control center should be able to activate this clone VM in case of failure of the original VM, taking a snapshot of the VM to enable live migration in a minimal amount of time.
- The migrated VM can run on a shared Internet connection. Only updated data and modified states are sent to the suspended VM to update its state.
- **Recovery Point Objective (RPO)** (how fresh recovered data will be) and **Recovery Time Objective (RTO)** (the amount of downtime a business can tolerate) are two of the most important parameters of a disaster recovery or data protection plan. VM migration allow higher RPOs.
- Security of the VMs should be enforced during live migration of VMs.

Virtualization Support and DR – **VM Cloning for Disaster Recovery**

- **The cloning of VMs offers an effective solution.** (…)



"A very short RPO, for example, 10 to 30 seconds, means that data must be backed up very frequently, necessitating the use of high-speed backup technologies such as data mirroring or continuous replication, especially if backups are stored off site in a cloud or other arrangement".

Virtualization Support and DR – **VM Cloning for Disaster Recovery**

- **The cloning of VMs offers an effective solution. (…) Resumo dos Principais benefícios:**
  - **Rapid Recovery (reduced downtime):** VM cloning enables rapid recovery, reducing downtime and ensuring business continuity. It allows the organization to restore critical applications and services quickly.
  - **Cost-Effective:** VM cloning is a cost-effective disaster recovery solution, as it requires fewer hardware resources than traditional disaster recovery solutions. VM cloning eliminates the need for additional physical servers or storage devices, saving money and space.
  - **Efficient Testing:** VM cloning enables efficient testing of disaster recovery procedures. The cloned VM can be tested in a safe environment without affecting the production environment. This enables organizations to identify and address any issues before an actual disaster occurs.
  - **Flexibility:** VM cloning provides flexibility, as it allows organizations to clone VMs across different data centers, cloud environments, or geographical locations. This flexibility enables organizations to design a disaster recovery plan that meets their specific needs.

Architectural Design Challenges

- *Six open challenges in cloud architecture development*

- **Challenge 1—Service Availability and Data Lock-in Problem**
  - The management of a cloud service by a single company is often the source of single points of failure. **To achieve HA, one can consider using multiple cloud providers**. Even if a company has multiple data centers located in different geographic regions, it may have common software infrastructure and accounting systems. Therefore, **using multiple cloud providers may provide more protection from failures.**
  - Another availability obstacle is distributed denial of service (DDoS) attacks. Criminals threaten to cut off the incomes of SaaS providers by making their services unavailable. (…)
  - **Software stacks have improved interoperability among different cloud platforms, but the APIs itself are still proprietary.** Thus, customers cannot easily extract their data and programs from one site to run on another.
  - The obvious solution is to standardize the APIs so that a SaaS developer can deploy services and data across multiple cloud providers. This will rescue the loss of all data due to the failure of a single company. In addition to mitigating data lock-in concerns, standardization of APIs enables a new usage model in which the same software infrastructure can be used in both public and private clouds.

Architectural Design Challenges

- *Six open challenges in cloud architecture development*

- **Challenge 2—Data Privacy and Security Concerns**
  - **Current cloud offerings are essentially public networks, exposing the system to more attacks**.
  - Many obstacles can be overcome immediately with well-understood technologies such as encrypted storage, virtual LANs, and network middleboxes (e.g., firewalls, packet filters). For example, you could encrypt your data before placing it in a cloud. Many nations have laws requiring SaaS providers to keep customer data and copyrighted material within national boundaries.
  - Traditional network attacks include buffer overflows, DoS attacks, spyware, malware, rootkits, Trojan horses, and worms.
  - In a cloud environment, newer attacks may result from hypervisor malware, guest hopping and hijacking, or VM rootkits. Another type of attack is the man-in-the-middle attack for VM migrations. In general, passive attacks steal sensitive data or passwords. Active attacks may manipulate kernel data structures which will cause major damage to cloud servers.

Architectural Design Challenges

- *Six open challenges in cloud architecture development*

- **Challenge 3—Unpredictable Performance and Bottlenecks**
  - **Multiple VMs can share CPUs and main memory in cloud computing, but I/O sharing is problematic**. For example, to run 75 EC2 instances with the STREAM benchmark requires a mean bandwidth of 1,355 MB/second. However, for each of the 75 EC2 instances to write 1 GB files to the local disk requires a mean disk write bandwidth of only 55 MB/second. This demonstrates the problem of I/O interference between VMs. One solution is to improve I/O architectures and operating systems to efficiently virtualize interrupts and I/O channels.
  - **Internet applications continue to become more data-intensive.** If we assume applications to be "pulled apart" across the boundaries of clouds, this may complicate data placement and transport. Cloud users and providers have to think about the implications of placement and traffic at every level of the system, if they want to minimize costs. **Therefore, data transfer bottlenecks must be removed, bottleneck links must be widened, and weak servers should be removed.**

Architectural Design Challenges

- *Six open challenges in cloud architecture development*

- **Challenge 4—Distributed Storage and Widespread Software Bugs**
  - The database is always growing in cloud applications. The opportunity is to create a storage system that will not only meet this growth, but also combine it with the cloud advantage of scaling arbitrarily up and down on demand.
  - This demands the design of efficient distributed SANs. Data centers must meet programmers' expectations in terms of scalability, data durability, and HA. **Data consistence checking in SAN-connected data centers is a major challenge in cloud computing.**
  - Large-scale distributed bugs cannot be reproduced, so the debugging must occur at a scale in the production data centers. One solution may be a reliance on using VMs in cloud computing. The level of virtualization may make it possible to capture valuable information in ways that are impossible without using VMs. Debugging over simulators is another approach to attacking the problem, if the simulator is well designed.

Architectural Design Challenges

- *Six open challenges in cloud architecture development*

- **Challenge 5—Cloud Scalability, Interoperability, and Standardization**
  - The pay-as-you-go model applies to storage and network bandwidth; both are counted in terms of the number of bytes used. Computation is different depending on virtualization level. GCP automatically scales in response to load increases and decreases; users are charged by the cycles used. AWS charges by the hour for the number of VM instances used, even if the machine is idle. The opportunity here is to scale quickly up and down in response to load variation, in order to save money, but without violating SLAs.
  - Open Virtualization Format (OVF) describes an open, secure, portable, efficient, and extensible format for the packaging and distribution of VMs. It also defines a format for distributing software to be deployed in VMs. The approach is to address virtual platform-agnostic packaging with certification and integrity of packaged software. The package supports virtual appliances to span more than one VM.

Architectural Design Challenges

- *Six open challenges in cloud architecture development*

- **Challenge 6—Software Licensing and Reputation Sharing**
  - Many cloud computing providers originally relied on open source software because the licensing model for commercial software is not ideal for utility computing. The primary opportunity is either for open source to remain popular or simply for commercial software companies to change their licensing structure to better fit cloud computing. One can consider using both pay-for-use and bulk-use licensing schemes to widen the business coverage.
  - One customer's bad behavior can affect the reputation of the entire cloud. For instance, blacklisting of EC2 IP addresses by spam-prevention services may limit smooth VM installation. An opportunity would be to create reputation-guarding services similar to the "trusted e-mail" services currently offered (for a fee) to services hosted on smaller ISPs. Another legal issue concerns the transfer of legal liability. Cloud providers want legal liability to remain with the customer, and vice versa. This problem must be solved at the SLA level.

PUBLIC CLOUD PLATFORMS: GAE, AWS, AND AZURE

- Cloud services are demanded by computing and IT administrators, software vendors, and end users.
- Figure 4.19 introduces five levels of cloud players.
  - At the top level, individual users and organizational users demand very different services.
  - The application providers at the SaaS level serve mainly individual users. Most business organizations are serviced by IaaS and PaaS providers.
  - The infrastructure services (IaaS) provide compute, storage, and communication resources to both applications and organizational users.
  - The cloud environment is defined by the PaaS or platform providers. Note that the platform providers support both infrastructure services and organizational users directly.

PUBLIC CLOUD PLATFORMS: GAE, AWS, AND AZURE

- Figure 4.19 introduces five levels of cloud players.

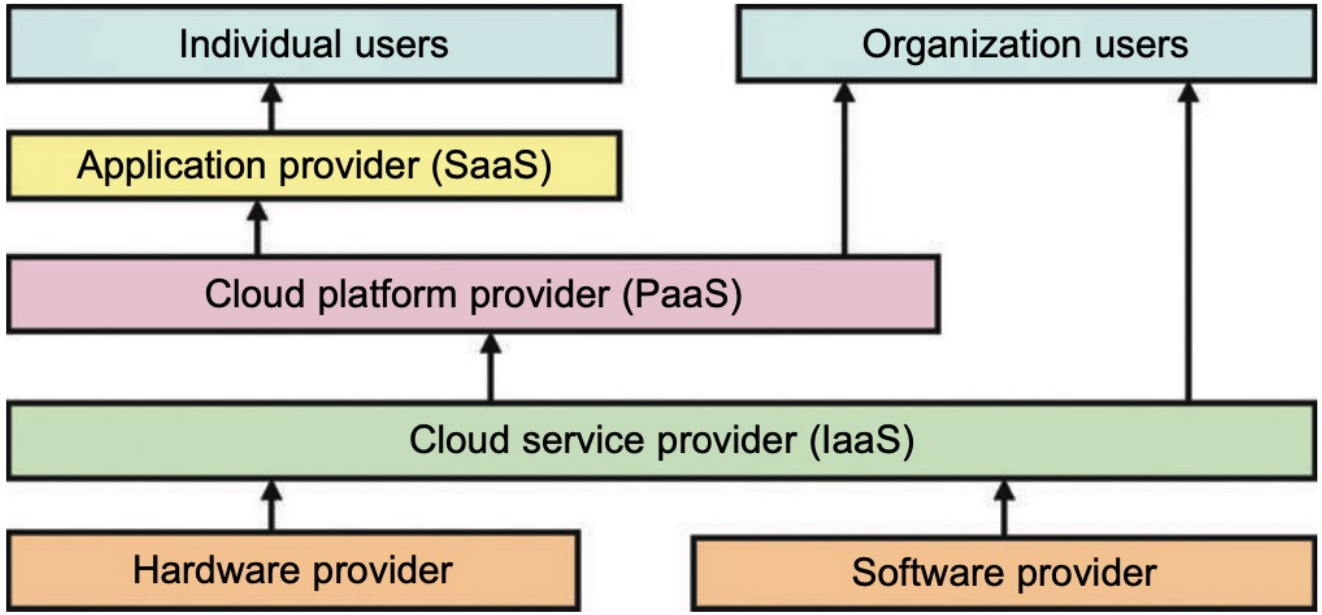


FIGURE 4.19

Roles of individual and organizational users and their interaction with cloud providers under various cloud service models.

PUBLIC CLOUD PLATFORMS: GCP, AWS, AND AZURE

- Cloud services rely on new advances in machine virtualization, SOA, grid infrastructure management, and power efficiency.
- Consumers purchase such services in the form of IaaS, PaaS, or SaaS as described earlier. Also, many cloud entrepreneurs are selling value-added utility services to massive numbers of users.
- The cloud industry leverages the growing demand by many enterprises and business users to outsource their computing and storage jobs to professional providers. The provider service charges are often much lower than the cost for users to replace their obsolete servers frequently.
- Table 4.5 summarizes the profiles of five major cloud providers by 2010 standards.

PUBLIC CLOUD PLATFORMS: GCP, AWS, AND AZURE

- Table 4.5 summarizes the profiles of five major cloud providers by 2010 standards.

Table 4.5 Five Major Cloud Platforms and Their Service Offerings [36]

Model	IBM	Amazon	Google	Microsoft	Salesforce
PaaS	BlueCloud, WCA, RC2		App Engine (GAE)	Windows Azure	Force.com
IaaS	Ensembles	AWS		Windows Azure	
SaaS	Lotus Live		Gmail, Docs	.NET service, Dynamic CRM	Online CRM, Gifttag
Virtualization		OS and Xen	Application Container	OS level/ Hypel-V	
Service Offerings	SOA, B2, TSAM, RAD, Web 2.0	EC2, S3, SQS, SimpleDB	GFS, Chubby, BigTable, MapReduce	Live, SQL Hotmail	Apex, visual force, record security
Security Features	WebSphere2 and PowerVM tuned for protection	PKI, VPN, EBS to recover from failure	Chubby locks for security enforcement	Replicated data, rule-based access control	Admin./record security, uses metadata API
User Interfaces		EC2 command-line tools	Web-based admin. console	Windows Azure portal	
Web API	Yes	Yes	Yes	Yes	Yes
Programming Support	AMI		Python	.NET Framework	

Note: *WCA: WebSphere CloudBurst Appliance; RC2: Research Compute Cloud; RAD: Rational Application Developer; SOA: Service-Oriented Architecture; TSAM: Tivoli Service Automation Manager; EC2: Elastic Compute Cloud; S3: Simple Storage Service; SQS: Simple Queue Service; GAE: Google App Engine; AWS: Amazon Web Services; SQL: Structured Query Language; EBS: Elastic Block Store; CRM: Consumer Relationship Management.*

Nota:
Esta lista de ofertas está datada mas é ilustrativa do potencial deste modelo computacional

Links Úteis

- Sobre Multitenancy
  - https://docs.microsoft.com/en-us/azure/architecture/multitenant-identity/.
- Sobre Cloud Trust
  - https://www.bearingpoint.com/en/insights-events/insights/in-cloud-we-trust/
- Sobre RPO vs RTO
  - https://www.druva.com/blog/understanding-rpo-and-rto/