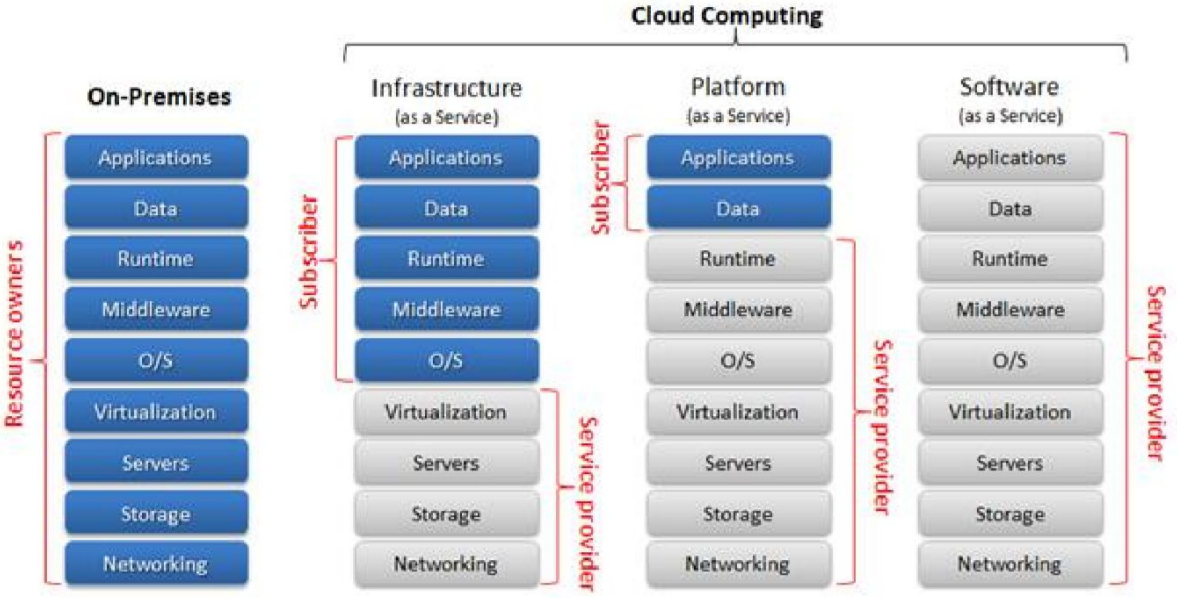
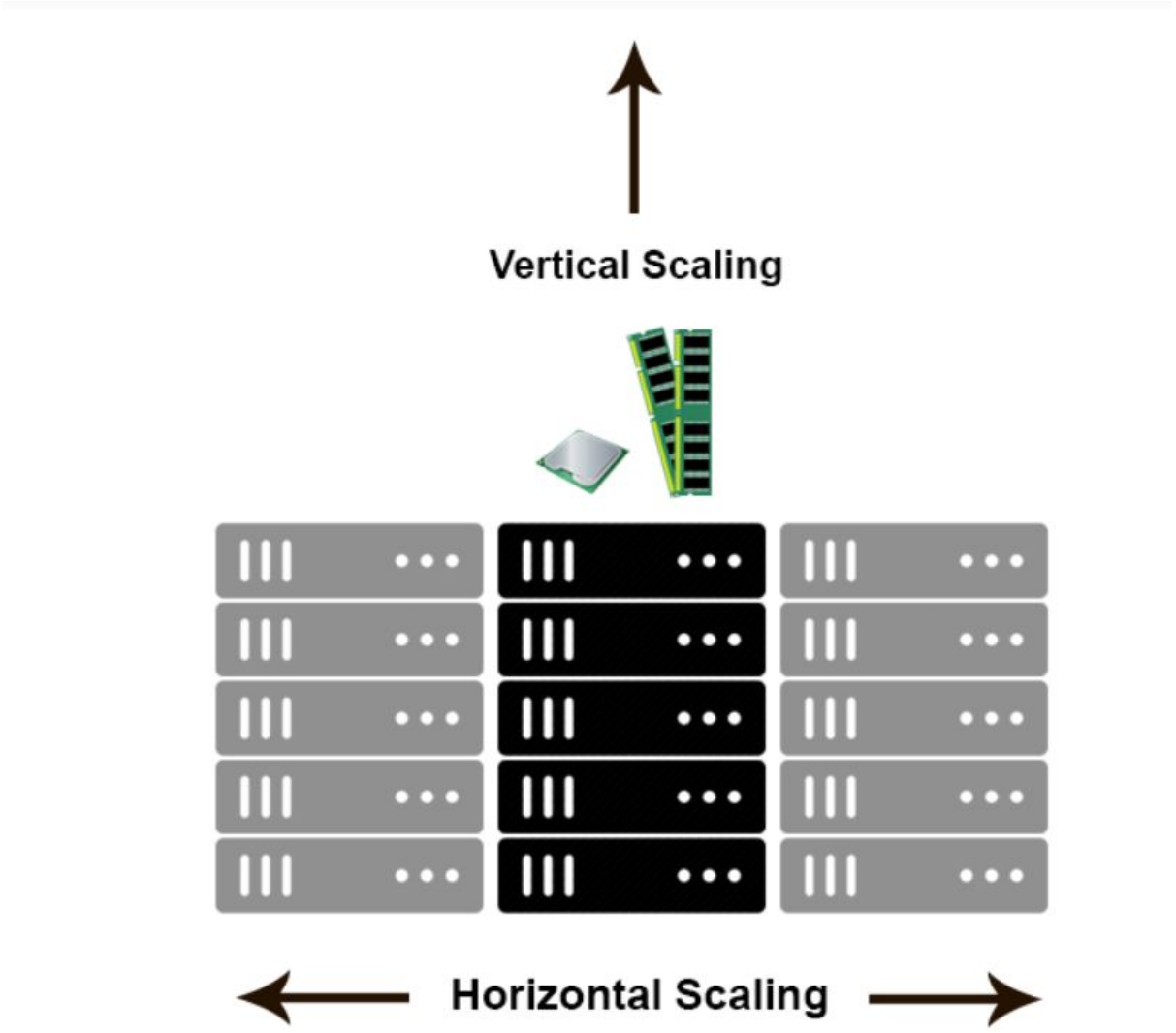
Week 7: Cloud Computing: Paradigm, Concepts, Models, Architectures & Technologies

1. Define cloud computing
2. List the pros and cons of cloud computing.
3. Distinguish between IaaS, PaaS and SaaS.
4. Define multi-tenancy. What is the difference between virtual and organic multi-tenancy?
5. What is the difference between horizontal scaling and vertical scaling? Describe scenarios in which you will use each type of scaling.
6. Define virtualization. What is the difference between full, para- and hardware-assisted virtualization?
7. Assume your company wants to launch an e-commerce website. Which cloud services and deployment models will you consider for the website?

1. Cloud Computing is a model for enabling ubiquitous, convenient and on-demand network access to a shared pool of dynamically configurable computing resources (networks, servers, storage, applications and services) that can be rapidly provisioned and made available to the business client with minimal management effort and service provider interaction.
2. Pros: improved disaster recovery; Increased Collaboration and Flexibility; Environmentally Friendly
   Cons: Internet Connectivity; Costs; Security
3. Software as a service/Platform as a service/Infrastructure as a service



4. Multi-tenancy is isolation between a shared system for multiple customers to provide all users with a dedicated share of the instance.
5. scaling up (vertical scaling) and/or scaling out (horizontal scaling)



Horizontal scaling means that you scale by adding more machines into your pool of resources whereas Vertical scaling means that you scale by adding more power (CPU, RAM) to an existing machine.

6. Define1: Partitioning of a physical computing system into multiple virtual resources - computing, storage, network and memory
   Define2: Creation of flexible substitutes for actual [physical] resource
   a) The substitutes have the same functions and external interfaces as their actual counterparts but that differ in attributes such as size, performance, and cost.
   b) The substitutes are called virtual resources; their users are typically unaware of the substitution.

   完全虚拟化（FV）完全将客户操作系统或虚拟机（VM）与底层硬件分离;
   Full Virtualisation (FV) completely decouples the guest OS or virtual machine (VM) from the underlying hardware;
   在半虚拟化（PV）中，修改VM以实现与管理程序的通信;

In Para-Virtualisation (PV), the VM is modified to enable communication with the hypervisor;
它是一种虚拟化技术，它提供了计算机硬件方面的抽象执行环境，客户操作系统可以在其上运行。
It is a virtualization technique that provides an abstract execution environment in terms of computer hardware on top of which a guest OS can be run.
硬件辅助虚拟化（HAV）减少了主机系统在管理权限和地址空间转换问题方面的参与。支持HAV允许VM更简单，更小的管理程序代码和接近本机的性能
Hardware-Assisted Virtualization (HAV) reduces the involvement of the host system in managing privilege and address space translation issues.
Supporting HAV allows simpler and smaller hypervisor code and near-native performance for VMs

7. none

Cloud Computing for the Internet of Things
Cloud Concepts & Technologies

1. What are the various layers in a virtualisation architecture?

Virtual Machine(applications/operating systems)
Virtualization Layer
Physical Layer

2. What is the difference between full and para-virtualisation?

见上文答案

3. What are the benefits of load balancing?

(Load balancing refers to efficiently distributing incoming network traffic across a group of backend servers, also known as a server farm or server pool.)

Enables Cloud-based apps to achieve high availability and reliability
To end-user, load balancing makes a pool of servers appear as single server with very high computing capacity.

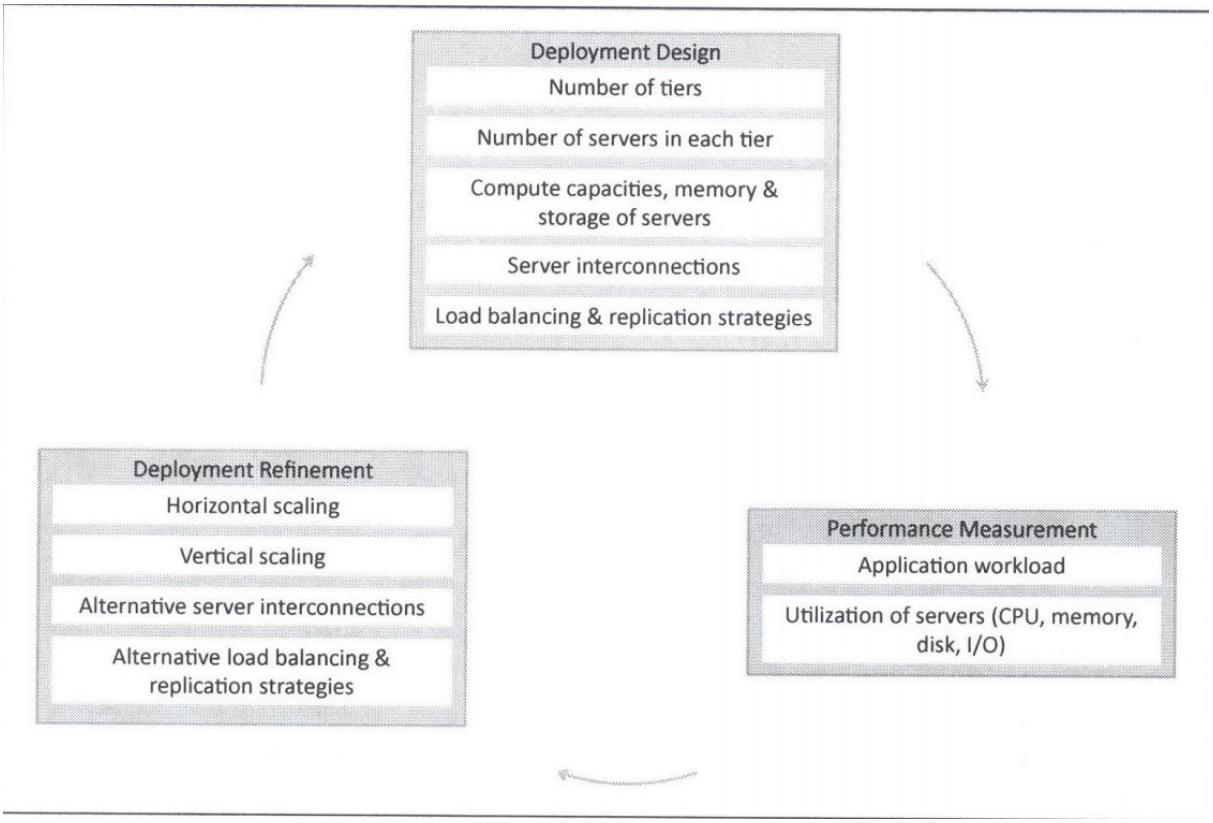
Elasticity VS Scalability:
Elasticity is to match the resources allocated with actual amount of resources needed at any given point in time.
Scalability includes the ability to increase workload size within existing infrastructure (hardware, software, etc.) without impacting performance.

4. What are sticky sessions?

none

5. What are the differences between traditional and on-demand scaling approaches?

6. What are the various stages in the cloud application deployment lifecycle?



7. What is the difference between array-based and host-based replication?
Host-based replication typically operates at a much more granular level than array-based replication products, and that means array-based replication is typically easier to manage and does not consume host processing power.
(Host-based replication using servers to copy data from one site to another. Array-based replication is an approach to data backup in which compatible storage arrays use built-in software to automatically copy data from one storage array to another.)

Since the replication software is installed locally on the host or machine image, it gains the advantage of observing the active changes to data inside the host or VM.

8. In MapReduce, what are the functions of map, reduce and combine tasks?
Map Task: Break down into five map tasks, where each mapper works on one of the five files and the mapper task goes through the data and returns the maximum temperature for each city.
Reduce Task: All five of these output streams would be fed into the reduce task, which combine the input results and output a single value for each city, producing a final result set.

(MapReduce is a parallel data processing model for processing and analysing massive scale data. A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks.)

9. Describe three applications that can benefit from the MapReduce programming model?

10. What are the limitations of the MapReduce programming model? [research on the Web – especially reasons why Google moved from MapReduce to Cloud Dataflow]

MapReduce got too cumbersome once the size of the data reached a few petabytes. Also, tasks that have a dependency on each other cannot be parallelized, which is not possible through MapReduce.

11. What are the various criteria for Service Level Agreements (SLA)?

Criteria	Details
Availability	Percentage of time the service is guaranteed to be available
Performance	Response time, Throughput
Disaster Recovery	Mean time to recover
Problem resolution	Process to identify problems, support options, resolution expectations
Security and privacy of data	Mechanisms for security of data in storage and transmission

Week 9. Cloud Application Architecture and Design

1. How can a Cloud app be made scalable?
    - ◆ Loose coupling of components (避免组件耦合)
    - ◆ Asynchronous communication (异步通信)
    - ◆ Stateless design (无状态)
    - ◆ Database choice & design (数据库选择与设计)
        - NoSQL has better horizontal scaling capability
        - NoSQL supports unstructured data
        - NoSQL supports MapReduce for processing massive scale of data.
2. What are the design considerations to make a Cloud app reliable?
    - ◆ No single point of failure
    - ◆ Trigger automated actions on failures
    - ◆ Robust (Graceful degradation) – some components remain available during failure
    - ◆ Logging – help in diagnosis and prediction of failure
    - ◆ Replication – multiple copies of data allows organisations to function during failure.
3. What are the various layers of the Service-Oriented Architecture (SOA)?
    - ◆ Business Systems: custom-built apps & legacy systems (e.g., ERP, CRM & SCM)
    - ◆ Service Components: Allow layers above it to interact with business systems and realise the functionality of services being exposed
    - ◆ Composite Services: higher level business processes created by orchestrating Service Components
    - ◆ Orchestrated Business Services: higher-level business services created by orchestrating Composite Services
    - ◆ Presentation Services: top-most layer, includes Uis exposing services and business processes to users
    - ◆ Enterprise Service Bus: integrates services via adapters, routing, transformation & messaging mechanis
4. What is the benefit of loose coupling in the Cloud Component Model (CCM)?
    - ◆ possible to scale each component independently
    - ◆ low upgrade & maintenance costs
    - ◆ easy to change and test for each independent components
5. What are the functions of the Model-View-Controller (MVC) model?
    - ◆ MVC separates data(Model), logic(Controller) and interface(View), which is a good design pattern to achieve decoupling.
        - **Model**: manages the data and the behaviour of the apps and processes events sent by Controller. Has no info about Views & Controllers. Handles requests for info and instructions to change state.

- **View**: Handles interface shown to the user. The user interacts with App via the Views, which present info that Model and Controller passes for presentation to the user. Also handles user requests.
- **Controller**: The bridge between Model and View. Processes user requests. Updates Model when the user manipulates the View. Also updates View when Model changes.

6. What are the benefits of and limitations of non-relational (No-SQL) databases over relational (SQL) database?
   - Pros:
     - NoSQL is **easy to scale**, while SQL has many constraints for their table, which is hard to scale.
     - High **availability** and **data tolerance** provided by data replication
     - No fixed schema, **support complex structure or unstructured data and Object**, while SQL has limited support for complex data structures.
     - Almost all NoSQL systems are **free** (open source) software, while many SQL systems are expensive (e.g. MS SQL Server, Oracle)
     - **support MapReduce** for processing massive scale of data
   - Cons:
     - Limited support for **aggregation**, SQL is very good at aggregation.
     - Performance for **join** is poor than SQL
     - No **ACID** (atomicity, consistency, isolation, durability) guarantee, so NoSQL have less suitable for applications which require high consistency. But SQL has ACID guarantee.
     - **Lack of consistent model**, so hard to migrant to another system. But SQL can easily migrant from one to another.
7. **What is the ACID guarantee? (我觉得今年大概率会考)**
   - Atomicity guarantees that each transaction is treated as a single "unit", which either succeeds completely or fails completely. It would not cause any half done and half fail.
   - Consistency ensures that a transaction can only bring the database from one valid state to another valid state.
   - Isolation ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially.
   - Durability guarantees that once a transaction has been committed, it will remain committed even in the case of a system crash, completed transactions would be stored in non-volatile storage.

WEEK 11

1. What is the benefit of decoupling Cloud-based application components?

同上

2. What is the use of message queues in Cloud-based applications?

In modern cloud architecture, applications are decoupled into smaller, independent building blocks that are easier to develop, deploy and maintain. Message queues provide communication and coordination for these distributed applications.
improving performance, reliability and scalability

3. How can Cloud-based application components be scaled independents?

Scale-up is done by adding more resources to an existing system to reach a desired state of performance. For example, a database or web server needs additional resources to continue performance at a certain level to meet SLAs.

More compute, memory, storage, or network can be added to that system to keep the performance at desired levels. When this is done in the cloud, applications often get moved onto more powerful instances and may even migrate to a different host and retire the server it was on. Of course, this process should be transparent to the customer. Scaling-up can also be done in software by adding more threads, more connections, or in cases of database applications, increasing cache sizes.

4. What are the benefits of non-relational database over relational databases?

同上

WEEK 12 Big Data analytics in cloud computing

1. What are the characteristics of Big Data?
   volume, velocity (速度) and variety

2. What is the stopping criteria used in k-means clustering?
   state stops change anymore

3. What is the difference between density reachability (密度可达性) and density connectivity (密度连通性)?
   Page 12

4. What is the difference between precision and accuracy
   Precision: the fraction of objects classified correctly;

   $$\text{Precision} = TP/TP+FP$$

   Accuracy: percentage of correct positive and negative predictions;

   $$\text{Accuracy} = TP+TN/TP+FP+FN+TN$$

5. What is the naïve assumption made by the Naïve Bayes algorithm?
   It is about the independence of feature attributes - the value of a particular feature is independent of the value of any other feature, given the class variable;

6. What is information gain?

7. What is an ensemble learning method?
   Random Forest (RF) is an ensemble learning method based on randomised decision trees;

8. What is a maximum margin hyper plane in SVM?

9. What is the difference between content-based recommendation and collaborative filtering?
   item-based or content-based recommendation: Provide recommendations to users for unrated items based on the characteristics of the item

   collaborative filtering: the ratings given by the user and other users to similar items;

WEEK12 Cloud Computing Security

Common approaches to cloud security: authorization (授权), authentication (身份验证), identify access management (识别访问管理), data security, data integrity, encryption key management

Cloud Security Alliance (CSA) (云安全联盟) - Trusted Cloud Initiative (TCI) (可信云计划) Reference Architecture (参考架构):
a methodology and a set of tools that assessments of where internal IT admins and cloud providers are in terms of security capabilities and to plan a roadmap to meet the security needs of their business.

Authentication: confirming the identity of the entity requesting access to some protected information

Authentication: mechanisms SSO, SAML Token, OTP

Two common SSO approaches: SAML SSO and Kerberos

SAML SSO: uses SAML which is an XML based open standard data format for exchanging security information between an identity provider and a service provider (使用SAML，这是一种基于XML的开放标准数据格式，用于在身份提供者和服务提供者之间交换安全信息)

Kerberos: uses tickets for authenticating client to a service that communicate over an un secure network (使用票证对通过不安全网络进行通信的服务进行客户端身份验证)

OTP: uses passwords which are valid for single use only for a single transaction or session

WEEK12 The Multimedia Cloud

1. What is the difference between HTTP Live Streaming and HTTP Dynamic Streaming?
   HTTP Live Streaming: HTTP streaming to iOS devices or devices **supporting HLS format** –optional encryption with AES128 (HTTP流式传输到iOS设备或支持HLS格式的设备 - 使用AES128进行可选加密)

   HTTP Dynamic Streaming: **On-demand and live adaptive bit-rate video** stream of MP4 media over HTTP (HTTP上的MP4媒体的按需和实时自适应比特率视频流)

2. What are the benefits of HTTP Live Streaming and HTTP Dynamic Streaming over RTMP?
   RTMP: Real time message protocol
   With using RTMP, it has a high quality, low latency, media streaming with support for live and on-demand and full adaptive bit-rate (自适应比特率)

3. What is the function of a pipeline (管道) in the Amazon Elastic Transcoder?
   NOT SHOWN IN PDF
   FROM Amazon: 管道是**管理转码任务的队列**。
   Pipelines are queues that **manage your transcoding jobs**.

   在创建任务时，您需要指定要将任务添加到的管道。Elastic Transcoder 按照添加任务的顺序开始处理管道中的任务。

一个常见配置是创建两个管道 — 一个管道用于标准优先级任务，另一个管道用于高优先级任务。大多数任务都会进入标准优先级管道；仅在您需要立即对文件转码时使用高优先级管道。

在您创建任务时如果管道中有其他任务，Elastic Transcoder 会在资源可用时开始处理新的任务。一个管道可以同时处理多个任务，并且完成任务所需的时间因要转换的文件的大小和任务规范而存在显著差异。因此，任务不一定会按照创建的顺序完成。

您可以临时暂停一个管道，它会停止处理任务。如果您想要取消一个或多个任务，这非常有用。您只能在 Elastic Transcoder 开始处理任务前执行取消操作。