Introduction

1. Describe (at least five) features of cloud computing. Justify your answer with details.

    - Elasticity: Using 1000 servers for 1 hour costs the same as 1 server for 1000 hours, Same price to get a result faster.
    - Measured service: Resource usage can be monitored, controlled, reported (thus being charged "pay-as-you-go").
    - Rapid elasticity: Quick scale up or scale down of resources through elastic provisioning or the release of capabilities in near real time, can change immediately according to user needs.
    - Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model like AWS, lower spending due to marginal effects
    - Broad network access. Capabilities are available over the network and accessed through heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations)
    - On-demand self-service: Unilateral provisioning of computing resources, such as server time, storage or network bandwidth, without requiring human interaction with service providers.
    - Resource planning: in traditional data center, the usage of service is not perfect balanced, when demand is more than resource at a time, service provider will lost user which mean lose revenue, and idle time resources are not being used which is low utilization, with cloud computing, There will be no such cons

2. Compare the differences between IaaS, PaaS, and SaaS.
3. Compare the benefits of cloud data centers with traditional data centers.
4. Describe the benefits of virtualization for cloud computing.

    Server consolidation: 10:1 in many cases:

    Resource on the physical server can be split into different VM with different sizes to meets user demand, the host workflow.

    Enable rapid deployment: an image can run on any hardware: we can copy image into software user needs, without VM, hardware configuration is usually fixed, upgrade or downgrade of the physical server is not easy, with image we can quickly deploy machine on the server

    Dynamic load balancing: move hotspot to under-utilized hardware

    Disaster recovery: move affected VMs to other hardware

5. Describe the four cloud deployment modes and compare their differences.

Data Center

6. Compare the concepts of region, availability zone, and data center.

A region is a separate geography unit such as central US, east US.
Availability Zones are unique physical locations within an Azure region. To ensure resiliency, there's a minimum of three separate zones in all enabled regions.
Each zone is made up of one or more data centers equipped with independent power, cooling, and networking.

7. What is PUE? How is PUE calculated? Describe the factors that may affect PUE.

   PUE is power usage effectiveness, it calculated by total power divided by server power. Factors includes IT equipment power (we need cooling), time and location, technology improvement.

8. What are the difficulties to scale up the computer?

Virtualization 1 & 2

9. Describe in detail the responsibility of the VM hypervisor.
10. Compare the differences between Type-1 and Type-2 hypervisor
11. Why a VM cannot execute privileged instructions directly?
12. Compare the differences between trap-and-emulate and binary translation?
13. Why is the container more lightweight than the VM?

    Containers are isolated but share OS and where appropriate bins/libraries (VM doesn't share OS and Bins/Libraries, all independent) which result significantly faster deployment, much less overhead, easier migration faster restart.

14. How does the Docker engine isolate resources among containers?
15. Why is Kubernetes not PaaS?
16. In Kubernetes, what is Pod/ReplicaSet/Deployment?

Hadoop

17. What is the typical size of data blocks in GFS/HDFS? Why?
18. How does GFS/HDFS maintain data reliability?
19. Describe the topology-aware block placement policy in HDFS.
20. What is the pipelined write in HDFS?
21. Why does HDFS not place all block replicas in different racks?
22. Using map() and reduce(), calculate $1+8+27+...+n^3$.
23. How does MapReduce exploit data locality in GFS/HDFS?
24. Why does MapReduce need to ensure that all key-value pairs with the same key are shuffled to the same reducer? How is it achieved?

    Example with very long vector that each entry in the vector can be considered to be a partitioned by, we can use same map function to do aggregation in parallel, the purpose of the reduce function is to aggregate the intermediate data obtained from the map function. a text input can be made to different lists with key value, for same key we can apply some function to

the value (adding, sum etc.…), in shuffling same key will be grouped together, in step shuffling to reducer, it will produces new key value by reduce function, if something goes wrong in key-value pairs, it will produces incorrect result.
Hadoop is designed for this problem by using sorting and shuffling, map function sort key-value pairs to corresponding group by the key, to determine which reduce function should sent to

25. Why the jobtracker in Hadoop 1.x can be a bottleneck? How is it addressed in Hadoop 2.x?
26. Compare the differences of scheduling policies in YARN?

Spark

27. What is an RDD? Why is it fundamental in Spark?
28. Describe in detail the three ways to create an RDD.
29. Compare the difference between RDD transformations and actions?
30. How can Spark achieve better performance in iterative jobs than MapReduce?
31. Calculate $1+8+27+...+n^3$ in spark.

Resource Scheduling

32. Given a link of the capacity of 10 Gbps, to send 3 flows with arrival rates of 3 Gbps, 6 Gbps, and 6 Gbps, compute the fair rates with the max-min fairness. If the weight of the first flow becomes 2 while the weights of the other flows remain as 1, compute their fair rates again
33. In a cluster of 12 CPU cores and 48 GB of memory, one job asks a resource scheduler to launch tasks of 1 CPU core and 8 GB of memory, and another job asks to launch tasks of 4 CPU cores and 2 GB of memory. What are the dominant resources of such two tasks? What are their dominant shares? With the dominant resource fairness, how many tasks can be launched in the cluster for the two jobs?
34. Why is the task with a finer granularity preferred by a cloud-scale resource scheduler?
35. Compare the differences between the designs of Borg and Mesos.

Virtualization 3

36. How does FaaS allow agile deployment and fast startup?
37. Why stateful applications are bad cases for serverless computing?

Key-value Store

38. Compare the differences between iterative and recursive queries.
39. What is strong consistency? How does quorum consensus guarantee strong consistency when there is no node failure or network partition?
40. What is consistent hashing and why is it desirable for the distributed hash table?
41. How does the “finger table” in Chord save the number of hops to locate a key?
42. How does hinted handoff work in Dynamo?
43. What is eventual consistency? How does Dynamo maintain eventual consistency?

Machine Learning

44. List the limitations of a kernel function in CUDA.

iterate all data
data is getting bigger
update data might be slow

45. Compare the differences between data parallelism and model parallelism.
    In data parallelism, the same model is replicated across multiple processors or nodes.
    Deal data quicker, but in model too large, Communication overhead also increased

    Pros 优点

    ▸Model agnostic
    ▸Horizontal scaling

    Cons ▸Very communication heavy 缺点

    ▸Noisy (async version)

    In model parallelism, different parts of the model are placed on different processors or nodes, and each processor computes its part of the model independently.
    Hard to partitioning

46. Compare the differences between synchronous and asynchronous training.

    consistent and reproducible training behavior

    synchronous training can be slower, requires all workers to finish their computations before updating the model parameters

    Workers independently of each other, faster

    asynchronous training can lead to inconsistent and less predictable training behavior.

47. Compare the communication overhead of single-master all-reduce, parameter-server all-reduce, and ring all-reduce.

    (P-1)*N

    (P-1)N/P

    2*(P-1)*N

Data Center Networking

48. Why is there over-subscription in the traditional data center network?
    Too much traffic being inspected
49. Briefly describe the features of FatTree and its benefits compared to the traditional topology.

FatTree is a network topology commonly used in data center environments. It is designed to provide high bandwidth, low latency, and fault tolerance.

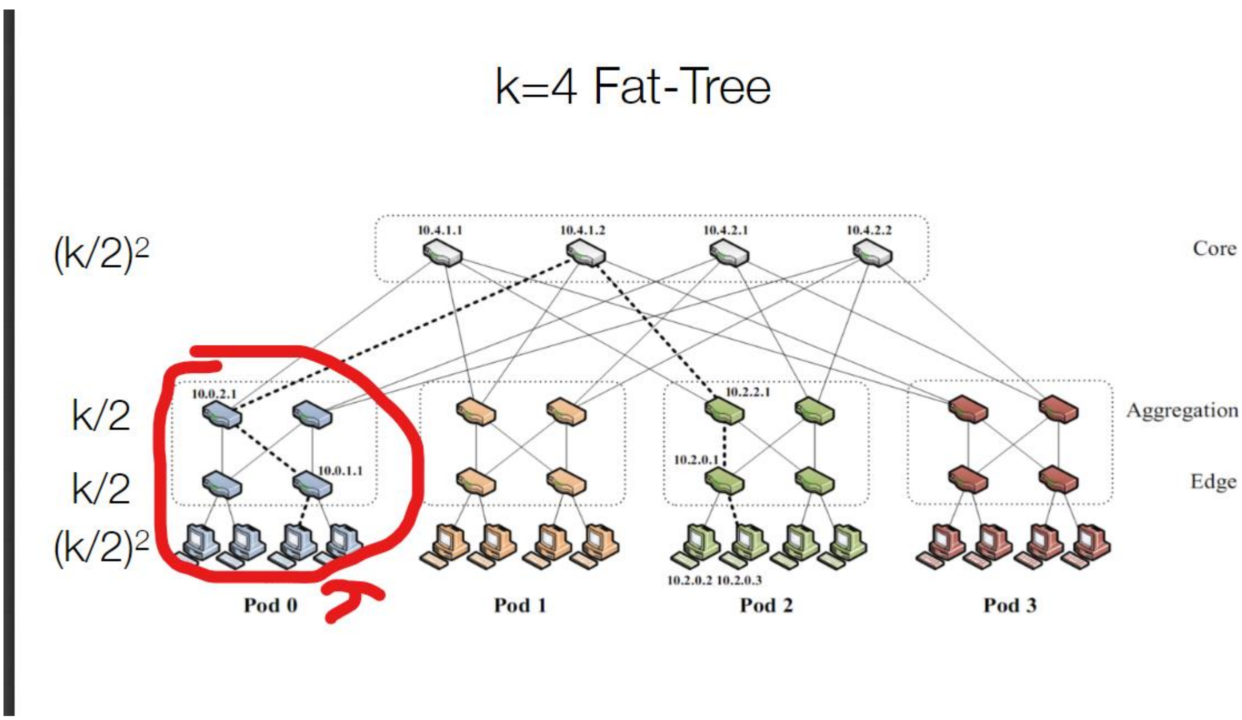
Clos topology:
Cheaper
Easier to scale
NO/low oversubscription
Higher path diversity

50. How many servers can be connected to a FatTree topology when k=64? How many switches are there in each layer?

Server:
each pod consists of (k/2)^2 servers & 2 layers of k/2 k-port switches

edge, aggregation and core

each edge switch connects to k/2 servers & k/2 aggr. switches
each aggr. switch connects to k/2 edge & k/2 core switches
(k/2)2 core switches: each connects to k pods
图片从下往上看



k/2 ^2 = 1024 server
agg = edge = 512
1024 core connect to each pods

51. Compare the differences between flow classification and flow scheduling?

1.Flow classification: Denote a flow as a sequence of packets with the same entries; pod switches forward subsequent packets of the same flow to same outgoing port, and periodically reassign a minimal number of output ports

▸Eliminates local congestion

▸Assign traffic to ports on a per-flow basis instead of a per-host basis, Ensure fair distribution on flows

2. Flow scheduling: Pay attention to routing large flows, edge switches detect any outgoing flow whose size grows above a predefined threshold, and then send notification to a central scheduler. The central scheduler tries to assign non-conflicting paths for these large flows.

▸Eliminates global congestion

▸Prevent long lived flows from sharing the same links – Assign long lived flows to different links

消除本地拥堵

按流量而不是按主机分配流量到端口，确保流量的公平分配

消除全局拥堵

防止长时间的流量共享相同的链路 - 将长时间的流量分配给不同的链路

52. Why is it difficult to deploy new network protocols on the Internet? How does SDN address such issues?

Operating a network is expensive
Buggy software in the equipment
The network is “in the way”
Over specified
Few people can innovate
Closed equipment

SDN(Software Defined Networking):
Horizontal Open interfaces Rapid innovation
Horizontal Open interfaces Rapid innovation Huge industry
Consistent, up-to-date global network view
At least one Network OS probably many. Open- and closed-source
Open interface to packet forwarding

53. What is incast? How does DCTCP address the incast problem?
many-to-one traffic patterns cause a network congestion collapse.

Extract multi-bit feedback from single-bit stream of ECN marks
–Reduce window size based on fraction of marked packets

54. Why does MPTCP perform worse when the number of connections per host is too low or too high?

MPTCP spreads application data over multiple subflows

too few connections, then the multipath benefits are not realized.
Too many connections can overload, leading to reduced performance.

Streaming Analytics

55. Explain the concepts of spouts and bolts in Apache Storm, and provide examples of each.
    Spouts:
    A Storm entity (process) that is a source of streams
    Often reads from a crawler or DB
    Twitter Streaming API

    Bolts:
    A Storm entity (process) that
    ▸ Processes input streams
    ▸ Outputs more streams for other bolts
    IRichBolt
56. Describe the role of the Nimbus and Supervisor daemons in the Apache Storm cluster.

    Master node - Nimbus daemon 分配任务

    ▸ Responsible for

    ▸ Distributing code around cluster

    ▸ Assigning tasks to machines

    ▸ Monitoring for failures of machines

    Worker node – Supervisor daemon 执行任务

    ▸ Runs on a machine (server)

    ▸ Listens for work assigned to its machines

    ▸ Runs “Executors”(which contain groups of tasks)

57. Explain DStream in Spark Streaming and how it relates to RDDs in Spark.

    Discretized Stream is Run a streaming computation as a series of very small, deterministic batch jobs(continuous stream of data)

    Chop up the live stream into batches of X seconds
    ▪ Spark treats each batch of data as RDDs
    and processes them using RDD
    operations
    ▪ Finally, the processed results of the RDD
    operations are returned in batches

Erasure Coding for Distributed Storage

58. Compare erasure coding with replication in data storage. Discuss the advantages and disadvantages of each approach in the context of distributed storage systems.
    Erasure Coding

    Huge Positive 容错能力强，存储开销少 但有很多缺点
    • Fault tolerance with less storage overhead!
    Many drawbacks
    • Encoding overhead
    • Decoding overhead
    • Updating overhead
    Deleting overhead
    • Update consistency
    • Fewer copies for serving reads
    • Larger minimum system size
    (Immutable data
    Data is stored for a long time after being written
    Storing lots of data (when storage overhead actually matters this is true)
    Low read rate)

    Replication - 简单常用但花费大量储存

    • Store multiple copies of the data
    • Simple and very commonly used!
    • But, requires a lot of extra storage
59. When is reconstruction necessary in a distributed storage system?
    rolling upgrade of disk
    load balancing
    transient unavailability and performance failure

60. In a Local Reconstruction Code, how does the local parity block reduce the reconstruction overhead of data blocks?

In a Local Reconstruction Code (LRC), data blocks are coded into local parity blocks in groups. This reduces the bandwidth and I/O required for reconstruction and reduces the number of coded blocks needed to be read during reconstruction

61. Evaluate and compare the performance of an LRC(12, 2, 2) and an RS(10, 4) in terms of storage overhead, reconstruction overhead, and fault tolerance.

RS(10,4)

reconstruction overhead = 10 (10,4)

storage overhead = 1.4 14/10 = 1.4

fault tolerance: 4

LRC(12,2,2):

reconstruction overhead = 6 (12/2=6) (12,2,4)

storage overhead = 1.333   12+2+2/12 = 1.3

fault tolerance: 86% Of 4

RS(12+4)>LRC(12+2+2) >RS(6+3)

LRC(12,4,2)

(12+4+2):

12/4 = 3

1.5

RS = Reed-Solomon

storage 4

Tolerating 2 failures

•3x replication = 3x storage overhead

•RS(4,2) = (4+2)/4 = 1.5x storage overhead•

Tolerating 4 failures

•5x replication = 5x storage overhead

•RS(10,4) = (10+4)/10 = 1.4x storage overhead

•RS(100,4) = (100+4)/100 = 1.04x storage overhead