

Cloud storage

Learning objectives

The following is a checklist that students can use during their learning in order to self-assess their mastery of the material.

- a. Can you describe the limitations of traditional (local) file systems?
- b. Can you explain what object storage is?
- c. Can you explain what the benefits of object storage are?
- d. Can you contrast or relate object storage with block storage, a file system, and the key-value model?
- e. Can you explain the three different ways, on the physical level, to deal with more data (scale up, scale out, write better code)?

- f. Can you explain why scaling out is less expensive than scaling up?
- g. Can you explain what aspects of the design of object storage enable scaling out and why?
- h. Do you know what a data centre is made of (racks, server nodes, storage nodes, etc.)?
- i. Do you know the difference between storage, memory, CPU and network and how the three are paramount in a cluster?
- j. Do you know rough, typical numbers (per-node storage capacity, memory, number of cores, etc.)?
- k. Do you know how storage and memory technologies (HDD, SSD and RAM) compare in terms of capacity and throughput?

- l. Can you sketch the (key-value) data model behind object storage? Can you spot instances of the key-value model in several other technologies seen in the lecture?
- m. Do you know the difference between data and metadata?
- n. Do you know the order of magnitude that can be achieved in terms of number of objects, and object size?
- o. Can you name a few big players for cloud-based object storage (vendors, consumers)?
- p. Can you describe the features of S3 and Azure Blob Storage on a high level? Do you know what a bucket and object are? What block blob storage, append blob storage and page blob storage are and how they work?

- q. Can you describe what the most important SLA (Service Level Agreement) parameters mean (e.g., latency, availability, durability) as well as their typical range?
- r. Can you describe a typical use case for object storage?
- s. Can you explain what a REST API is, what resources and methods are? Do you know the most important status codes (200, 404, 500, ...)? Do you know the most important methods? Do you know what a URI is? (Note that the difference between URI, IRI, URN, URL has become irrelevant lately and for the purpose of the course)
- t. Do you know what each letter stands for in CAP?
- u. Can you explain why, for large amounts of data, CAP becomes relevant over ACID?

Distributed file systems

Learning objectives

- a. Can you explain the difference between block storage and object storage?
- b. Can you explain the difference between the (logical) key-value model and a file system?
- c. Can you contrast block storage and object storage in terms of maximum number of objects/files?
- d. Can you contrast block storage and object storage in terms of the maximum size of objects/files?

- e. Do you know the order of magnitude of a block size for a local filesystem and for a distributed file system? Can you explain the rationale behind them with respect to latency and throughput?
- f. Do you know where HDFS shines and why?
- g. Do you know that HDFS files are updated by appending atomically and why?
- h. Do you know how HDFS performs in terms of throughput and latency?
- i. Do you know the main benefits of HDFS (commodity hardware, etc.)?
- j. Can you contrast centralized architectures to decentralized (peer-to-peer) architectures?

- k. Can you explain the HDFS architecture, what a NameNode is and what a DataNode is, how blocks are replicated?
- l. Can you sketch how the various components communicate with each other (client, NameNode, DataNode)?
- m. Can you point to the single points of failure of HDFS and explain how they can be addressed?
- n. Can you explain how the NameNode stores the file system namespace, in memory and on disk? In particular, can you explain how the namespace file and the edit log work together at startup time and how they get modified once the system is up and running?

- o. Can you explain what a Standby NameNode is? (Note: it has many predecessors that only have historical relevance in the development of HDFS: Backup NameNode, Secondary NameNode, Checkpoint NameNode, etc, but this is not important for the course)
- p. Are you able to use the HDFS shell (creating directories, reading, uploading and downloading files, etc.)?