2023

Distributed data

LEERDOEL ONDERZOEK

SEM PLASMEIJER

Contents

1. Data Complexities

1.1 Big data

Big data is defined by the big three V's: volume, velocity, and variety. Volume is the total amount of data, usually defined in terabytes or petabytes. To fill the volume, you need a large amount of data sources, variety describes the amount of different data sources from sensors, IOT to blogs and social media posts. Last is velocity Big Data is defined by its near real-time data processing speed. These 3 V's combine to make Big Data different compared to regular data sources which it might be based on.

1.2 Distributed data processing models

Data can be stored and retrieved in many different architecture designs each bring with them their own advantages and disadvantages. In this chapter we will discuss what these designs are and how they impact the overall design.

1.2.1 Data-centric architecture development

Lambda and Kappa are two styles of Big Data architecture styles which fall under this model to deliver near real-time views on data but they both have a different way to achieve this real-time model.

Lambda separates its data in two distinct layers before it is viewable. First it creates a batch layer which contains a portion of the data which is long-lived with its scope decided by the owner. Second is a speed layer which is used to compute data which is real-time. The reason for separating the two comes down to be able to create complex computations on the batch layers without slowing down the real-time view on the speed-layer which will handle the simpler request through both its own layer as the batch layer.

Kappa removes the layered system and streams all data through a real-time layer and the results are placed in serving layer ready to be queried. If possible, both cases can be handled on the same engine removing the need to maintain two complex layers mentioned in the Lambda example. Furthermore, due to the removal of the two layers all data can be accessed through a single serving point.

1.2.2 Data event sourcing

Event sourcing is the concept of storing mutations and creations of data rather than the data itself. This has a view benefits, first it allows to us to view the total history of changes an object has gone through. Within this event log each data object will build upon the previous and as such it comes with multiple use cases. It allows for a complete rebuild where we delete the application and rerun all previous events to recreate it. It allows for rerunning parts of events and branching off into different event lines at certain points. And it allows for replaying all events with small modifications from a certain event taking in the change into all future events.

2. CAP

- **Consistency:** All nodes in the system have the same view of the data at any time (similar to the C in ACID).
- **Availability:** The system always responds to requests.
- **Partition tolerance:** The system remains operational if network problems occur between system nodes.

CAP theorem is the theory where a system cannot contain all three of the letters within it. A consistent system which is always available cannot be partitioned well due to the traditional system it requires. A system which is always available and is partition tolerant cannot guarantee consistency because each part needs to be updated separately leading to inconsistency over time. And if a system needs to be consistent even when a partition is updated needs to be down till all partitions are updated.