SEMINAR 6

- What are the articles about?
  1. In the first article, it can be found information about Lambda Architecture, how it appears, how it works, and why it is needed. Also, benefits and constraints about this Lambda Architecture are presented.
  2. In the second article, it can be found data about large migrations, what migration is in general, why this process is considered to be hard. Also, the 4 steps of data migration are present with a proper example.
- What Lambda Architecture tries to solve?

  As I understood from the article, Lambda Architecture tries to solve problems with analytics and data reprocessing. Lambda Architecture emphasizes retaining the input data unchanged. Reprocessing is one of the key challenges of stream processing but is very often ignored. Lambda Architecture somehow “beats the CAP theorem” by allowing a mixture of different data systems with different trade-offs. Also, Lambda Architecture deserves a lot of credit for highlighting the problem with the code change.
- What is the critique given against Lambda Architecture?

  The problem with Lambda Architecture is that maintaining a code that aims to produce the same result in 2 complex distributed systems is very hard. Also, Debugging and running 2 systems simultaneously is also complicated. There are some possible solutions, for example, the usage of frameworks in order to abstract those 2 distributed systems, but it doesn’t actually solve the problem, it just makes the system a little bit better.
- Explain the keywords
  - □ CAP theorem - let’s start by presenting the 3 major goals of microservices design: consistency, availability, and partition tolerance. Having this, the CAP theorem states that in a distributed system only 2 out of 3 of these goals could be delivered.
  - □ MapReduce - this is a framework, that is designed to process a large amount of data in parallel, having large clusters. It is done in a reliable and fault-tolerant manner. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples
  - □ Migration - moving objects (in the case of this article millions of objects) from one database to another.

- Explain the 4 steps of data migration.
    1. Dual writing - this step involves creating a new database table. After this, the data should be written in both initial and newly created table in order to keep them in sync. The next point for this step is to add missing data to the new database from the old one, in order to hold identical information.
    2. Changing all read paths - starting from this step all data should be read from the newly created tables. In the article, it is described how they verified if the data is consistent and reading from new tables is safe. They used a tool(GitHub's Scientist) and verified there could be any inconsistency.
    3. Changing all write paths - this step means changing the order of how data is written to the databases. Until now, data was written to the old store and then copied to the new one. Then they decided to reverse the order , and to start writing the data into the new store and then archive it in the old store. An important aspect is that they kept the old store because they have the possibility to make incremental updates and observe each change carefully.
    4. Removing old data - this step is pretty obvious because it means removing code that writes to the old store and new tables are now the only source.
- What can we achieve if we follow the pattern for data migration provided in the article?
    1. No interruptions on read/write operations of data.
    2. Incremental changes.
    3. Transparency and observability.