CSCI 540
Lecture Summary
Ibrahim Barada

I started the lecture by recalling the databases that we covered during the semester and reiterated their use cases. Then, I introduced the idea of a polyglot system and the importance of orchestrating a robust polyglot system in modern applications. To keep things practical, I used a real-life generic booking system as our starting point. The goal in the lecture is to design a booking system that scales to thousands of users, executes jobs in real-time, and most importantly, is accurate and sound.

We started building the system together as a team where I would ask a question and then based on the answers from the audience we decide whether the suggestion would work or not. We started by constructing the frontend layer, then microservices and finally all the focus was on databases.

We covered relational, document store, caching and column store databases based on our previous lectures throughout the semester. Then, I introduced new technologies (not covered extensively in class) such as ElasticSearch, RabbitMQ, S3 and Kafka. I tried to focus more on the new technologies by explaining their common use cases and demonstrating their use case in our generic system. I also illustrated how we can replace some technologies with other ones and the system will still work fine.

During the lecture, I covered the following databases:

- PostgreSQL (Relational)
- MongoDB (Document-oriented)
- Hadoop (Column Store)
- ElasticSearch (Distributed full-text search engine)
- Redis (Key-value)
- RabbitMQ (message-broker)
- S3 (File Store)
- Kafka (Event streaming platform)

The lecture was meant to be interactive and engaging and I wanted to build the system with the audience. This is a bit tricky because you need to adjust on the fly. For example, let’s say you asked a question on what database would fit this scenario and you get an answer that is good but not quite what you prepared for.

Therefore, I had to prepare a lot to make sure whatever the audience recommended, I can accommodate it in the architecture. I also had to know why, or why not, certain technologies would work in each scenario. This approach is a little exhaustive in the sense that you need to learn about what’s out there in the industry but it kept me a step ahead of the audience and prepared me to answer their questions.

My main two goals of this lecture was to show the class what a polyglot system looks like and illustrate some great tools that they can potentially use in their own projects. Accordingly, I tried to cover around 9 technologies and several data architecture concepts in 50mins.Therefore, we didn’t deep dive into any of the technologies but everyone has an idea on what each technology is supposed to do. I believe that with this approach, students can deep dive into the technology based on their interest and future work.

I think the main reason people got excited is due to the fact that I started the lecture with something that the audience knew beforehand. I was able to get their attention and show them something that they already know in a different environment. This also allowed us to introduce more advanced concepts (search engines, streaming, queueing) more easily.

An important lesson that I learned is time management. Although I did practice and made sure I’m on time, I didn’t fully account for the audience's questions. As a result, I ended up condensing the final part covering Rabbitmq. Next time, I’ll make sure to keep enough room for questions and feedback and prioritize my topics

Finally, I believe that overall the lecture was successful and rewarding. I did learn a lot preparing for the lecture as well as during lecturing and post lecture feedback.