Craig O'Loughlin 04/23/23

CS 470: Final Reflection

Cloud Presentation: https://youtu.be/QRAaBk1LYcA

Experiences and Strengths

In this final course we covered both containerization via Docker and development of cloud applications through AWS. The focus was on converting a working application built on the MEAN tech stack into a cloud solution. We broke this application into component containers and orchestrated their deployment via Docker Compose. We then reconfigured the application components into cloud pieces on AWS using S3, Lambda, API Gateway, and DynamoDB.

As a newly graduated software developer, one of my strengths is my familiarity with new tech stacks such as this. Working with cloud services is, and will continue to be, important for the field of software development across many applications. Also, on top of understanding these systems, I also have gained an understanding of *why* we would want to use (or not use) certain technologies such as the cloud for our applications.

I am now open to a variety of working roles in the software field at the entry level. I understand that as a graduate I do not know everything but am ready to apply my skills and continue to grow. I am prepared to begin a role working with web development across the stack, across data science projects, and even embedded systems and desktop native applications. This program has covered a wide range of topics and has given me a broad understanding across the board, along with teaching me how to learn quickly and effectively.

Planning for Growth

Scaling a web application can be difficult when you are managing your own hardware. Understanding how much hardware is enough for current demand, how much will be needed in the future, and how the data structure in your application may need to stretch or change, are all important considerations and may present roadblocks.

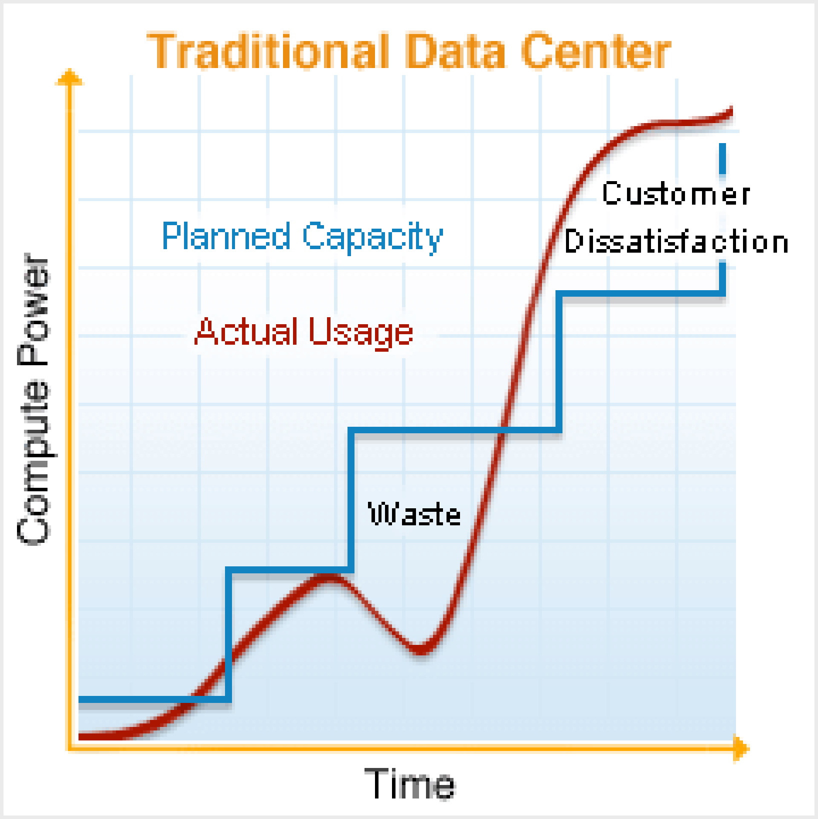
Serverless computing through cloud services presents a complete solution to scaling issues. No hardware is managed, no frameworks are managed, and a highly flexible database can be used. These systems can scale themselves as needed, from 0 to 1,000s of requests per second, or more. A dedicated server that is managed yourself is far less flexible, of course, but can be highly tuned to your exact needs. This can become important in large or mission critical applications.

The pricing of cloud services varies, but for a serverless setup using Lambda, for example, cost is based on runtime of the application microservices. This can be less predictable than running your own server hardware but can represent large cost savings for applications that have highly variable demands, low demands, or gaps in demand. In those cases, large capacity does not need to be available constantly, and you do not have to pay to maintain such.

Scalability and costing should be considered as early on as possible. It is never known exactly how much demand an application will eventually need. With a cloud approach, elasticity in availability (that always matches exactly what you need) (figure 1) and pay-for-service make sense, especially for small applications. However, at larger scales, it may be worthwhile to handle serving capabilities and costs yourself. In between, it is often difficult to forecast exactly how much serving power you need at any time. This results in having to buy more than you need and is a con towards a traditional approach.

Figure 1

Elasticity vs Tiered Service



Note: From Kutera, R. (2012). https://www.researchgate.net/figure/Capacity-vs-usage-comparison-in-on-premise-and-on-demand-models-Source-awsamazoncom_fig5_316276786

Overall, cloud services are not always the best solution for your application, but often offer large advantages in service cost and flexibility which are making them more and more popular among business application development.