One more knowledge point to study:

According to the agreed proposal, we can close our course by finishing 60% of the key content. To satisfy this requirement, I am very keen to supplement a key knowledge point of cloud computing, i.e. Elasticity.

As you may remember, we have emphasized Scalability very much. All our study on Salability is actually to help justify the very important and possibly confusing concept Elasticity. Let's ask ourselves two questions to help distinguish between Scalability and Elasticity:

1. Recall our previous homework on scaling a system. Assume one finished the homework in one day, while another finished it within five days. It will be both ok as long as your homework was submitted on time. However, if scaling the system is a real job driven by customer requirements, which one' job is preferred with higher customer satisfaction?
2. Imagine we are in charge of two systems, and one needs to be scaled to double size (workers and especially resources) while another needs to be scaled to triple size. Can we just simply scale both systems to triple sizes to satisfy the two requirements? Functionally speaking, the answer is Yes. But should we do so? If not, why?

Try to think of these two questions before going through the following materials.

After thinking about those two questions, we can try to clarify Elasticity from Scalability. In general, scalability refers to the ability of a system to deal with the gradually increasing amounts of workloads in a graceful manner, which is indeed part of what elasticity needs. Nevertheless, good scalability does not necessarily ensure good elasticity. Without necessarily giving any definition, we consider two elements of elasticity of cloud services/applications/systems, i.e.:

- The faster scaling speed, the better elasticity.
- The lower scaling cost, the better elasticity.

Recommended Reference:

Elasticity in Cloud Computing: What It Is, and What It Is Not

There are two key technologies for realizing/implementing good elasticity in the cloud, one is virtualization and the other is auto-scaling.

When it comes to the cloud virtualization, the de facto solution is to employ the hypervisor-based technologies, and the most representative cloud service type is offering virtual machines (VMs). In this virtualization solution, the hypervisor manages physical computing resources and makes isolated slices of hardware available for creating VMs.

However, the evolution direction of cloud virtualization seems to be an alternative and lightweight solution, namely container-based virtualization. Unlike the hardware-level solution of hypervisors, containers realize virtualization at the OS level and utilize isolated slices of the host OS to shield their contained applications. In essence, a container is composed of one or

more lightweight images, and each image is a prebaked and replaceable file system that includes necessary binaries, libraries or middlewares for running the application.

Recommended Reference:

Cloud Container Technologies: A State-of-the-Art Review

Auto-scaling is a way to automatically scale up or down the number of compute resources that are being allocated to your application based on its needs at any given time. Please do not be confused by its name. According to the aforementioned elements of elasticity, you may notice that the "automatically" and "scale down" of auto-scaling are actually to speed up scaling and to reduce the cost of scaling, so auto-scaling is essentially the technology for Elasticity.

There are tons of auto-scaling implementations, or auto-scalers. Unfortunately, many successful ones are confidential business products in the industry, so it is impossible for us to learn how they are implemented. But no matter how an auto-scaler is implemented, its essential theory must follow the ancient while powerful MAPE loop:

- M – Monitoring
- A – Analyzing
- P – Planning
- E – Executing

Recommended Reference:

A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments

The final Practice for this knowledge point:

We use this practice to cover both Container and Auto-scaling to help understand Elasticity.

Task 1: Create your first container to give a message of Hello World, by following the formal documents at: https://docs.docker.com/get-started/#test-docker-installation

There are also third-party tutorials we can refer to:
https://docker-curriculum.com/
https://stackify.com/docker-tutorial/

Note that no need to go through the detailed tutorial. Just pick up what you need to read and practice. A very small set of Docker commands should be enough to finish this task.

Task 2: Get started with Kubernetes (using Python), by following the guidelines at:
https://kubernetes.io/blog/2019/07/23/get-started-with-kubernetes-using-python/

You are also free and encouraged to search more references about Kubernetes to read and refer to. Getting more familiar with those buzzwords will be helpful for your future work/interviews anyway. Kubernetes (or k8s) is a popular open-source container-orchestration system for

automating application deployment, scaling, and management. So it can be viewed as an auto-scaler. Note that, since we do not employ cloud resources (ideally unlimited resources) in this practice, the auto-scaling has a predefined limit (e.g., “replicas: 4” in the deployment.yaml in the guidelines).

*Task 3 (optional): Replicate your previous Master-Worker application to the Kubernetes environment, i.e. the worker is deployed in a container. Note that Kubernetes will help you automatically replicate your workers. Then let your master issue different amounts of workloads to your worker, and observe and analyze different performances.

NOTES:

1. You are not required to come to the campus to study and practice, but you are more than welcome to come to my office for discussions and help if you like.
2. My former students actually have done Task1 before, so you may also talk to @Juan Fecci, @Carlos Landero, @Jhon, @Francisco for discussions and help.
3. Do not forget to document what you have done in the practice, because I will have to use your documented report to understand and grade your learning performance.
4. The due of the final practice report will be 6pm on 31 January 2020 (Friday).