Lecture Note Session 14

CSH1F2 Introduction to Computer Science

Telkom University Sem-1 2018-2019 / Anton Herutomo (AHT)

https://www.linkedin.com/in/antonherutomo

Here you can see some general and practical principles and key terms in modern software development implemented using Agile methodology, cloud computing, and user experience orientation.

The important things here is **User Experience**. Everything is measured against, translated from, and directed toward user experience. Personally, I took many of the concepts from Google Cloud Platform, however it should be similar to other platform as well such as Amazon Web Services or Microsoft Azure.

You can find below key principles in subsequent subjects in School of Computing department in Telkom University. Try to relate key terms in this lecture note with related subjects such as software engineering, software testing, distributed systems, as well as applied subjects such as data engineering and artificial intelligence. I hope it will benefit you toward your technology startup or career journey.

Don't hesitate to contact me at anton.herutomo@gmail.com or antonh@telkomuniversity.ac.id for further discussion prompt. Hope you enjoy the dynamics with my guidance and thanks for being with me in this class, see you soon!

Consideration when architecting solutions using Cloud Platform:

Generic Principles

- **Agile steps** when do architecting cloud solutions: 1 begin simple and iterate, 2 plan for failure, 3 measure stuff.

- Requirements can be divided into 4 categories: 1 qualitative requirements, 2 quantitative requirements, 3 scaling requirements, 4 sizing requirements.

- Think thoroughly of your state information in your solutions. Example of state information: session information (should it be stored in local SSD filesystem, or in separate database such as Redis?

- For measurement: you should differentiate between SLIs, SLOs, and SLAs (SLI = Service Level Indicators, SLO = Service Level Objectives, SLA = Service Level Agreements)

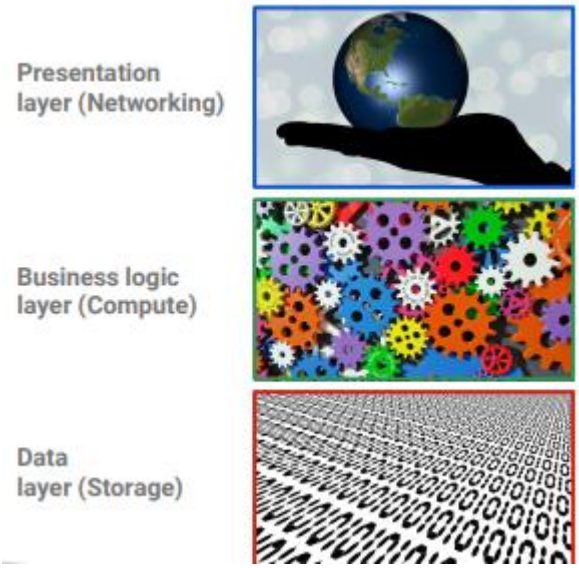

Fig 1 Design Architecture for Cloud Computing Platform

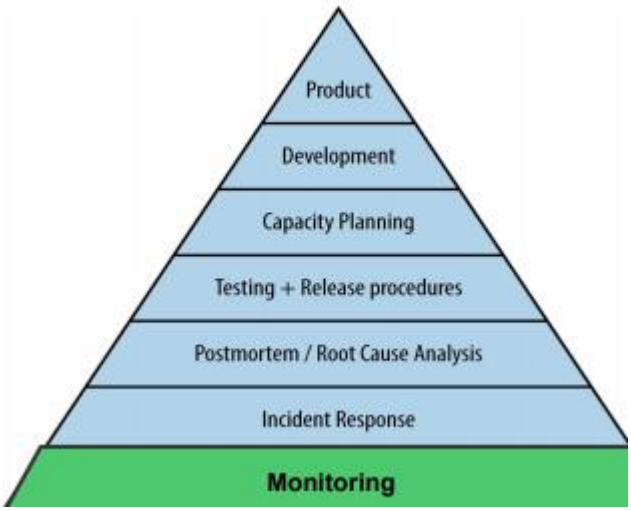

Fig 2 Monitoring is the Basis for Site Reliability Engineering (SRE)

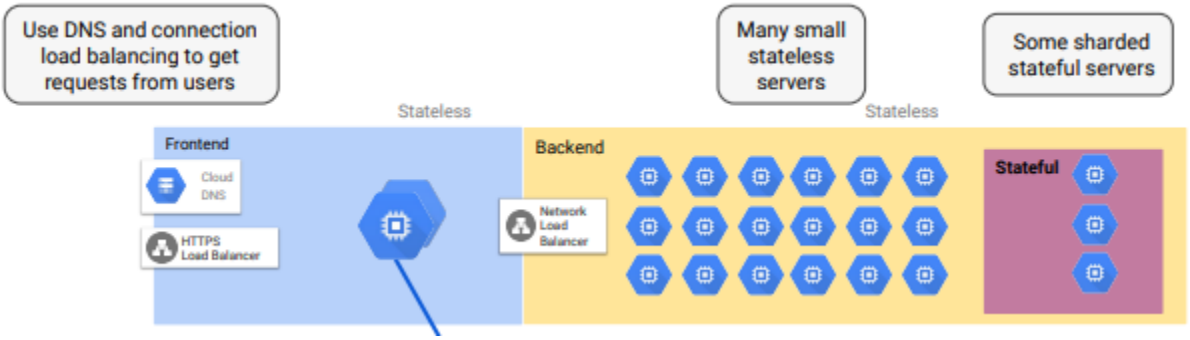

Fig 3 Large Scale Cloud Based Systems

Data Layer (Storage)

- From user point of view: they want to be able to **access the data when they need it**.
- Pay attention to data persistency (how to store data, should it be ephemeral or persistence) and access (how data are accessed).
- In data, there a well-known principle called **CAP theorem**. You can only pick 2 among these 3: 1 consistency, 2 availability, 3 partition tolerance.
- Terms: data ingestion = getting data into the cloud.
- Terms: mega maid = data migration tools for petabyte scale data (how much is 1 PB actually? :-D).

Fig 4 Storage Options in Google Cloud Platform

Business Logic Layer (Compute)

- Consider to design using **microservices** SOA (service oriented architecture) if you want your solutions to be scalable and **have many users of an atomic unit of functionality**.
- In microservice, the principle of software development is: horizontal scaling (small servers at first) … that is: **design first, dimension later**.
- Follow 12 Factor Design (http://12factor.net) Principles.

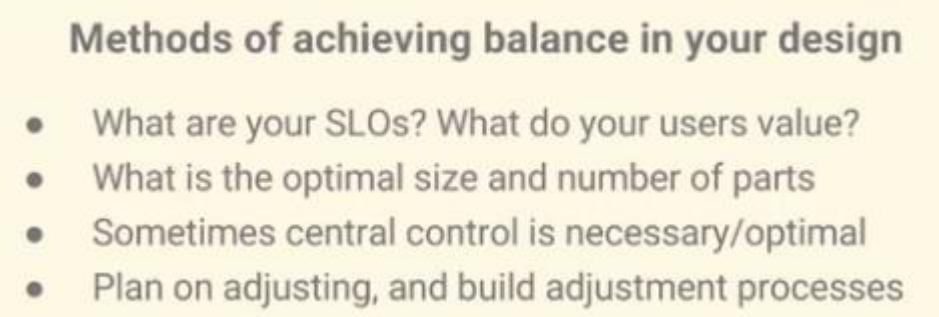

Fig 5 How to design cloud-based systems properly using SLOs (Service Level Objectives)

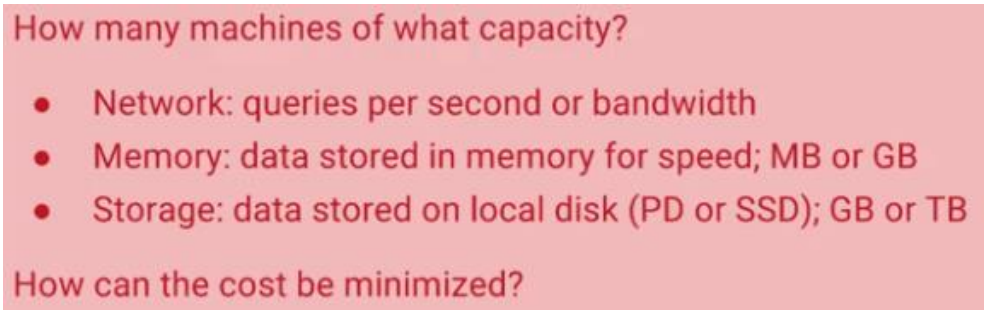

Fig 6 (virtual) machine consideration in cloud-based systems

Presentation Layer (Networking)

- Pay attention to 3 things when designing solutions: locations, load balancing, and caching.
- Location should be based on assumption that you cannot rely on just 1 cloud provider. Multi providers are preferred, for example: AWS and GCP, AWS and on premise (bare metal) data center, Azure and GCP.
- Consider this example: in 1 second, no more than 6-7 round trips between Europe and the US are possible, but approximately 2000 round trips can be achieved within a data centre.
- Load balancing can be selected using application layer (HTTP protocol), presentation layer (SSL), transport layer (TCP or UDP protocol), or network layer (IP protocol), if we see it from ISO OSI (Open Systems Interconnection) protocol stack.
- Caching is usually achieved using CDN (Content Delivery Networks) nowadays. Caching content at the edges of cloud provider's network provides faster delivery of content to your users while reducing costs.

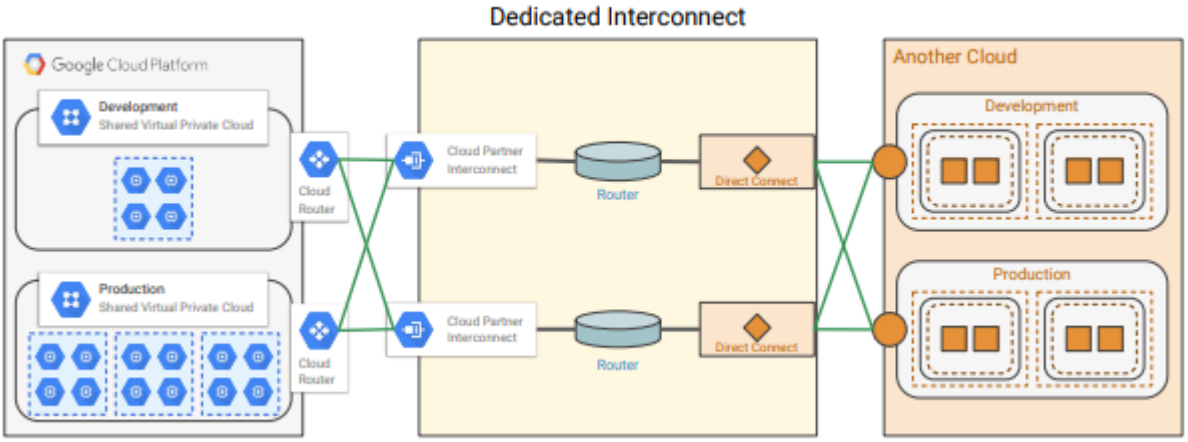

Fig 7 Example of multi infrastructure provider implementation