云计算技术与工程

负载均衡技术

徐敏贤

副研究员
云计算研究中心
中国科学院深圳先进技术研究院
http://www.minxianxu.info/vcc

造化钟神秀，阴阳割昏晓。荡胸生曾云，决眦入归鸟。
——（唐）杜甫

内容回顾

- Q1: 什么是数据中心?

- A simplistic view
  - Scaled-up/out version of machine rooms for enterprise computing
  - A large collection of commodity components
    - PC-based servers (CPUs, DRAM, disks),
    - Ethernet networking
    - Commodity OS and software stack
    - 10 to 100 of thousands of nodes
  - High-bandwidth networking (10Gbps → 40Gbps → 100Gbps)
  - Power delivery, cooling, UPS (50 – 200MW)
  - System software for DC management (centralized or distributed)
  - Software that implements internet services

内容回顾

- Q2: 数据中心的主要开销（TCO）有哪些?

- Total Cost of Ownership (TCO) components
  - Capital expenses (CAPEX)
    - Facilities, compute, storage, networking
  - Operational expenses (OPEX)
    - Energy, maintenance, employees, …

内容回顾

- Q3: 数据中心通常采用什么指标（metrics）度量服务的性能？如何提升这些指标？

- Throughput
  - User requests per second (RPS)
  - Scale-out addresses this (more servers)
- Quality of Service (QoS)
  - Latency of individual requests (90th, 95th, or 99th percentile)
  - Scale-out does not necessarily help
- Interesting notes
    - The distribution matters, not just the averages
    - Optimizing throughput often hurts latency
      - And optimizing latency often hurts power consumption
  - At the end, it is RPS/$ within some QoS constraints

内容回顾

- Q4: 哪些原因会导致数据中心服务的性能变化（variability）？

- Interference:
  - Different apps or different requests from the same app are sharing resources (compute, memory, network, etc.)
  - Background tasks: profiling/monitoring daemons, garbage collection, log compaction, data reconstruction, etc.
- Heterogeneity:
  - DC does not have a single platform (different generations, configs)
  - Performance varies across platforms
- Power limits & energy management:
  - Machines have to throttle when running over their Thermal Design Power
  - Machines go to low-power modes when underutilized/unused
- Input load:
  - Diurnal input load (high during day, low during night)
  - Unexpected spikes (need to provision for them)
  - Varying sizes of input dataset for analytics jobs (e.g., Hadoop)

目录 Contents

Why Are Load Balancing Important?

- The cloud needs quality of service monitoring to access the offered services for fulfilling user demands and to maintain the service level agreement
- Cloud may pose several issues and challenges
  - **Load balancing**
  - Performance analysis and modelling
  - Security and privacy issues
  - …….
- Load balancing allows avoiding the situation of overloading or underutilization during task computation

Perfectly Balanced

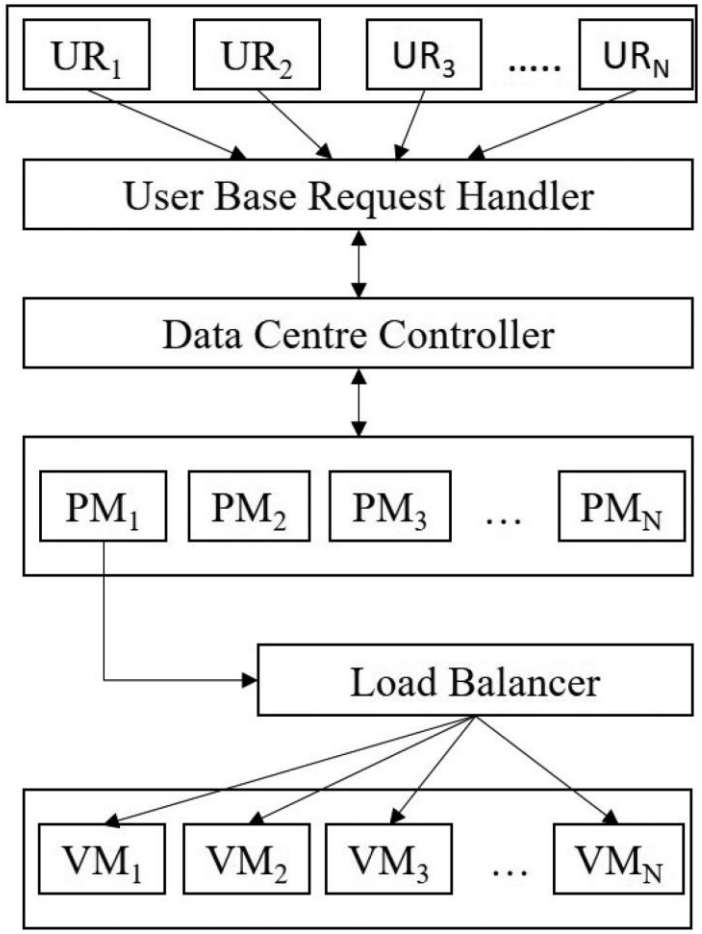
Thanos perfectly balanced as all things should be.

-- 《Avengers: Infinity War》

Need of Load Balancing

- Distribute the workloads equally on available resources
- Provide continuous service in case of failure of services' components
- Minimize the resource time of tasks and improve resource utilization
- Provide scalability and flexibility for those application whose size may increase in future
- Provide priority to jobs that need instant execution as compared with another jobs

Traditional Load Balancing Model



UR – User Request
DC – Data Centre
PM – Physical Machine
VM – Virtual Machine
N – No. of PM, VMs, UR

Q: Model is simple, but any challenges?

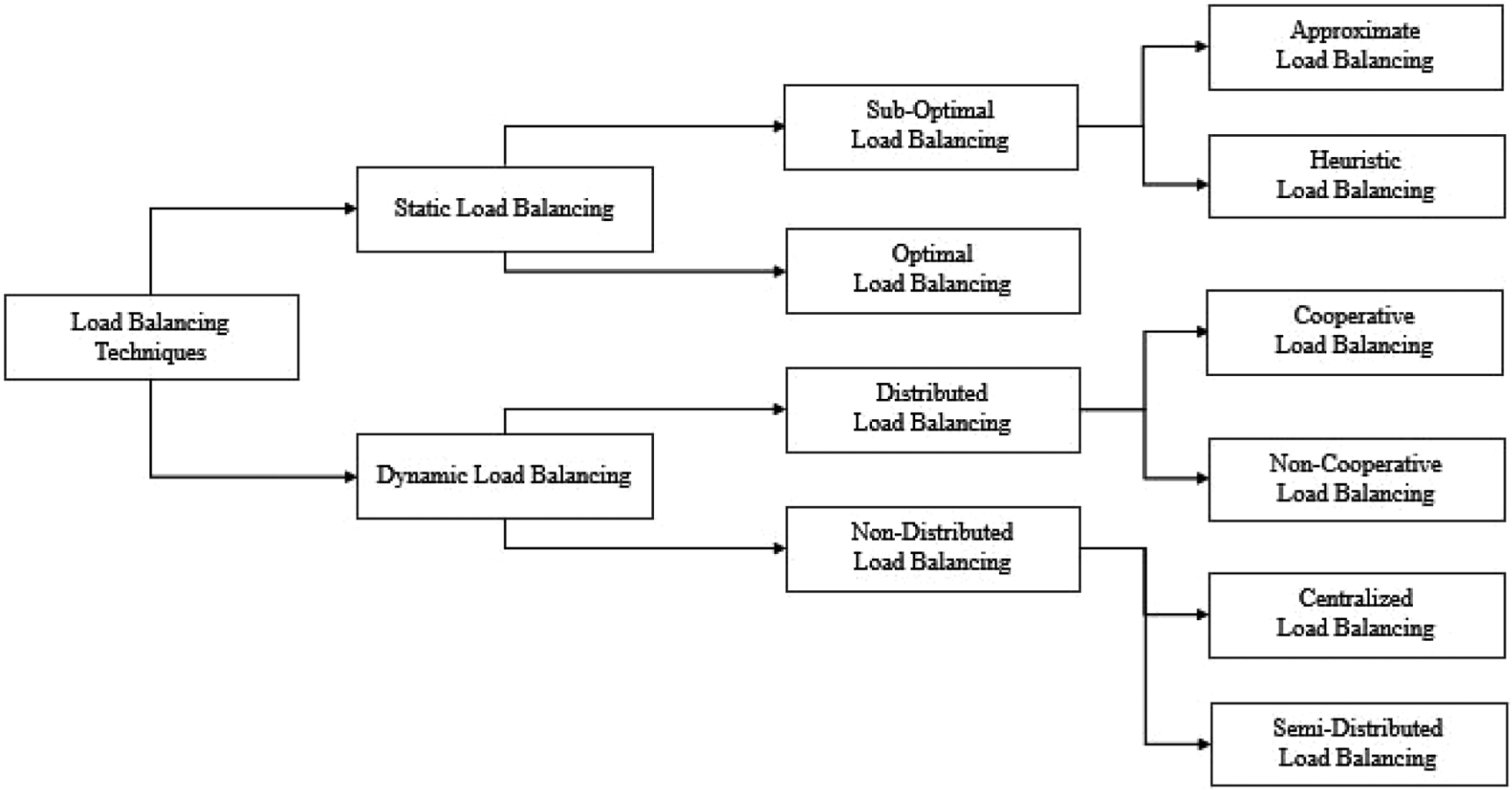
Challenges in Load Balancing in Cloud

- Geographical Distributed Nodes
  - Spatially distributed nodes are treated as a single location system for efficient execution of user requests
  - Consider factors such as network delay, communication delay, distance between the distributed computing nodes.
- Single Point of Failure
  - If the central device crashes, then it will affect the overall computing environment
- Virtual Machine Migration
  - VMs are independent in nature and have different configurations

Challenges in Load Balancing in Cloud

- Heterogeneous Nodes
  - User requirements change dynamically which requires executing them
- Storage Management
  - Full data replication schemes are not very efficient
  - Partial replication can be sufficient but dataset availability should be concerned
- Load-Balancer Scalability
  - Consider quick changes in demands in terms of computing power, storage, system topology
- Algorithm Complexity
  - A complex algorithm will reduce the performance and efficiency of the cloud system.

Classification of load balancing strategies



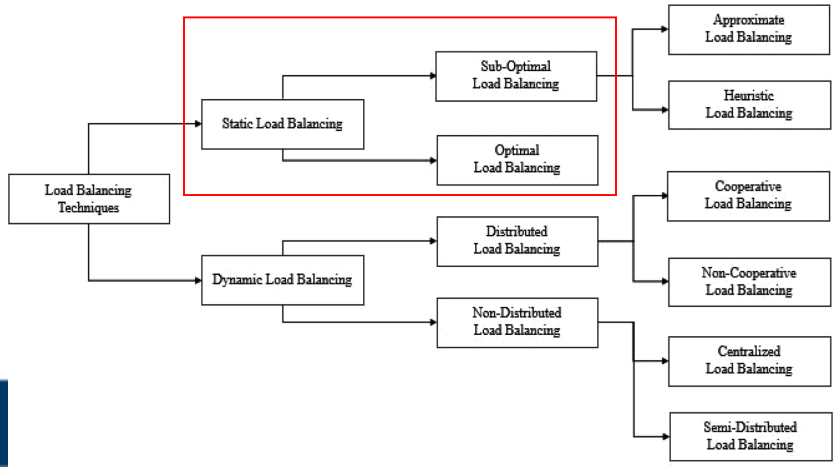
Kumar et al, Issues and Challenges of Load Balancing Techniques in Cloud Computing: A Survey, ACM Computing Survey, 2019

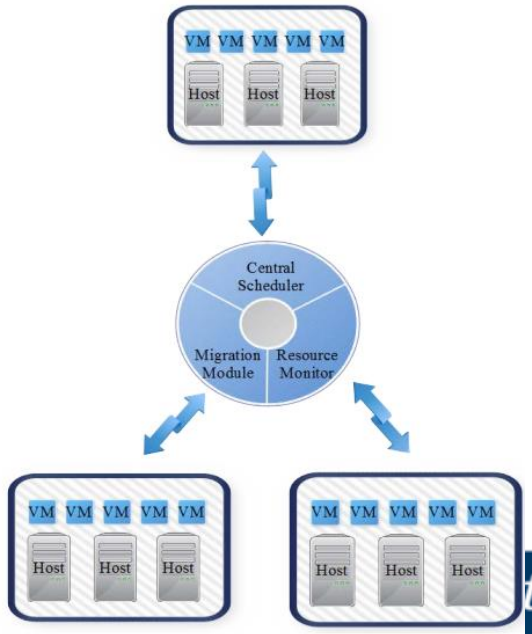
Process Initiation

- Sender Initiated
  - If a node is overloaded, it looks for other nodes that are lightly loaded to share the workload. The sender initiates the process to find the underloaded nodes when nodes get congested.
- Receiver Initiated
  - Receiver or lightly loaded nodes look for heavily loaded nodes to share the workload.
- Symmetric
  - both sender-initiated process and receiver-initiated process techniques are combined.

Static Algorithms

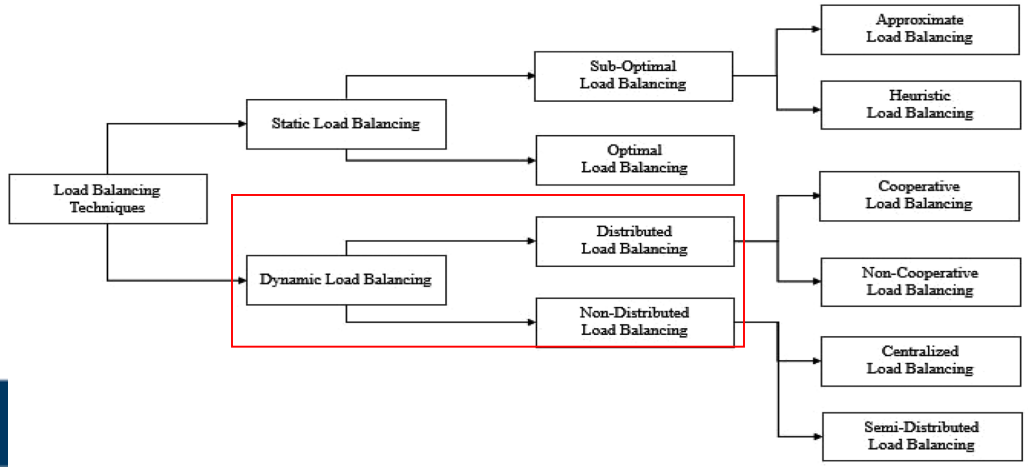
- Static
  - Follow a fixed set of rules that are not dependent on the system's current state
  - E.g. Min-Min, Max-Min, Round-Robin, Shortest Job First……
  - Q: Advantages and disadvantage?
- Optimal
  - Collect information about resources and sends tasks to the load balancer, which does an optimal allocation in minimum time.
- Suboptimal
  - Unable to determine optimal decisions

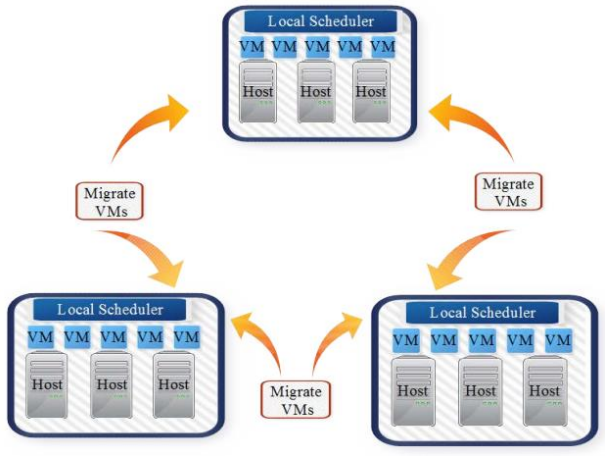




Dynamic Algorithms

- Dynamic
  - Consider the current state of the system and make a decision
  - Agent-based, dynamic threshold, markov chain, ant colony optimization……
  - Advantages and disadvantage?
- Distributed
  - All nodes maintain an information base for communication to distribute and redistribute the tasks efficiently
- Non-distributed
  - a single node or some nodes make the decision for load distribution





Load Balancing Metrics

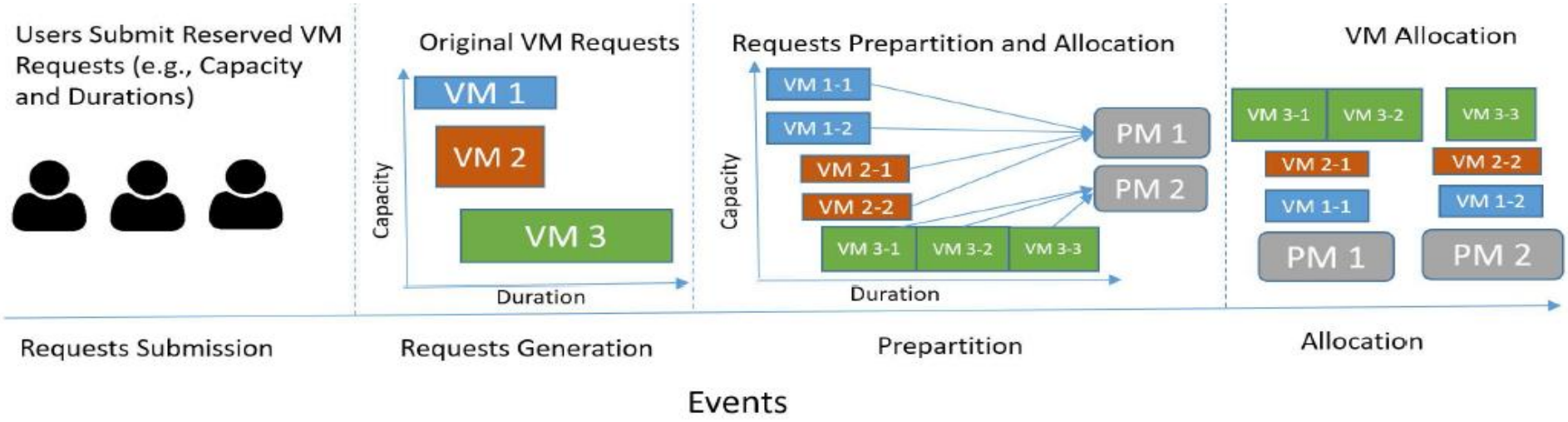
- Performance
- Response Time
- Throughput
- Scalability
- Fault Tolerance
- Migration Time
- Resource Use
- Degree of Imbalance
- Makespan
- ……

Load Balancing Policies

- Selection Policy
  - Identify the tasks that should be transferred from one node to another based on the migration overhead
- Location Policy
  - Determine the computing nodes that are underloaded or free and transfers tasks to them for processing
- Transfer Policy
  - Discover the circumstances in which tasks are required to transfer from a local node to another local/remote node
- Information Policy
  - Keep all resource information in the system

Prepartition (JCST 2023)

- Key idea: Partitioned requests before real scheduling
- Online approach for VM load balancing
  - Approximation ratio: $1 + \frac{1}{k} - \frac{1}{mk}$
- Offline approach for VM load balancing
  - Approximation ratio $(1+\frac{1}{k})$



Overload Control in WeChat (SoCC '18)

- For WeChat, the total amount of requests to the entry services is normally $10^{10} \sim 10^{11}$ on a daily basis
- Deployed 3000 services over 20000 machines
- Chinese Lunar New Year, the peak amount of workload can rise up to around 10 times of the daily average

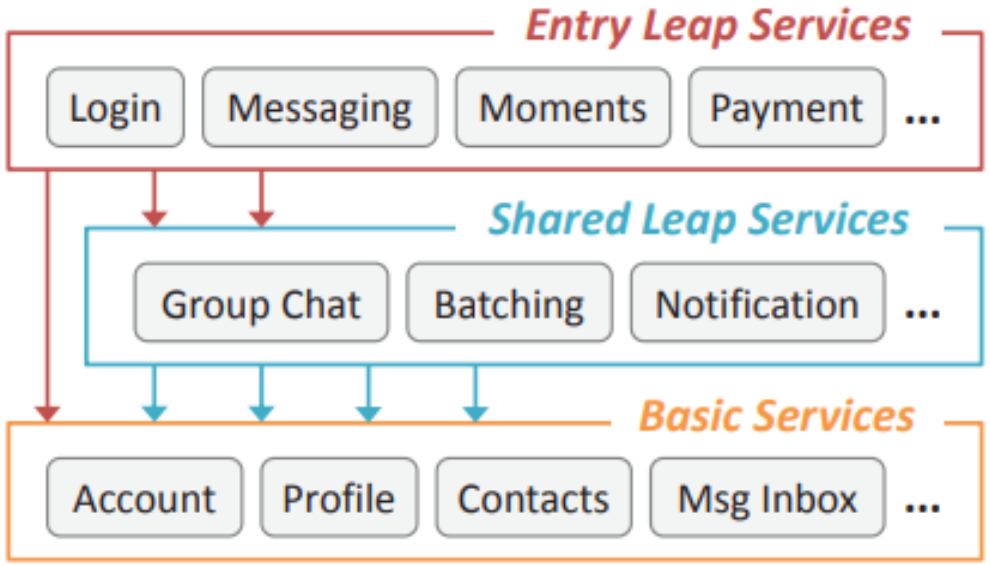
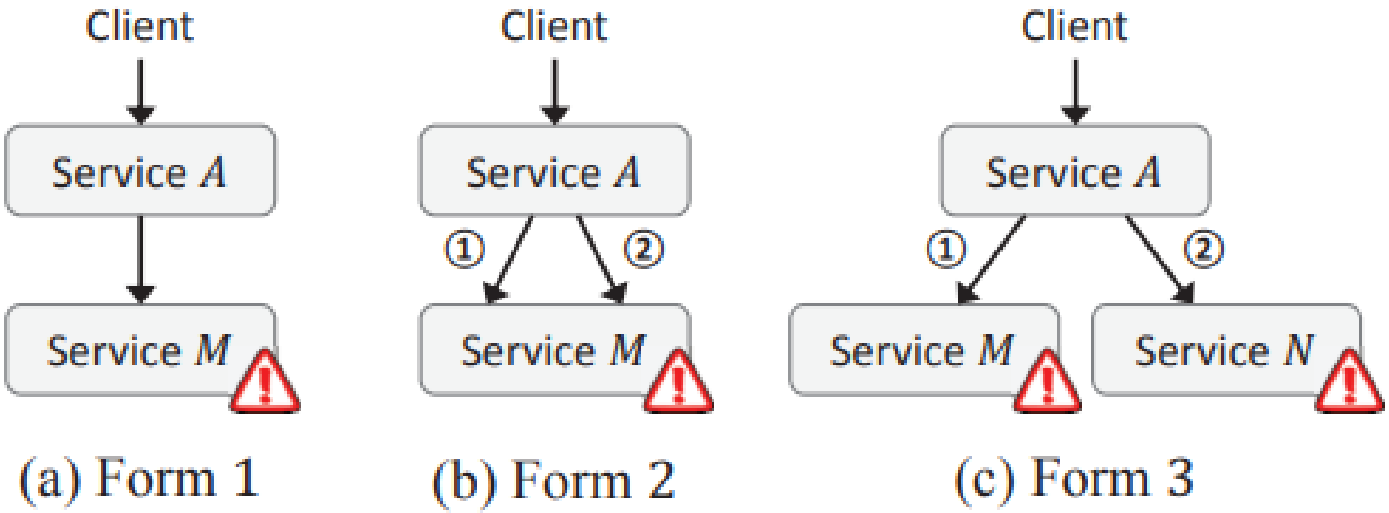


Figure 1: WeChat's microservice architecture.

Overload Scenario in WeChat

- Subsequent Overload



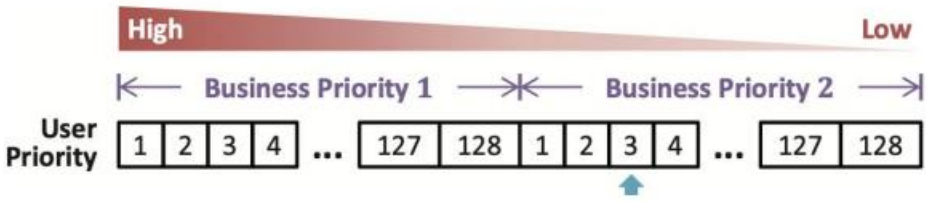
(a) Form 1 (b) Form 2 (c) Form 3

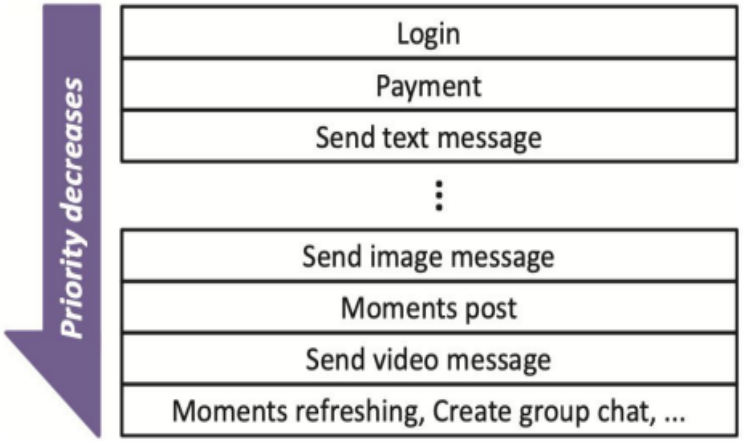
Challenge & Insight

- No single entry point for services request and with complex call path
- Excessive request aborts waste the computational resources
- Excessive request affects user experience (due to the high latency of service response)

DAGOR: Service Admission Control

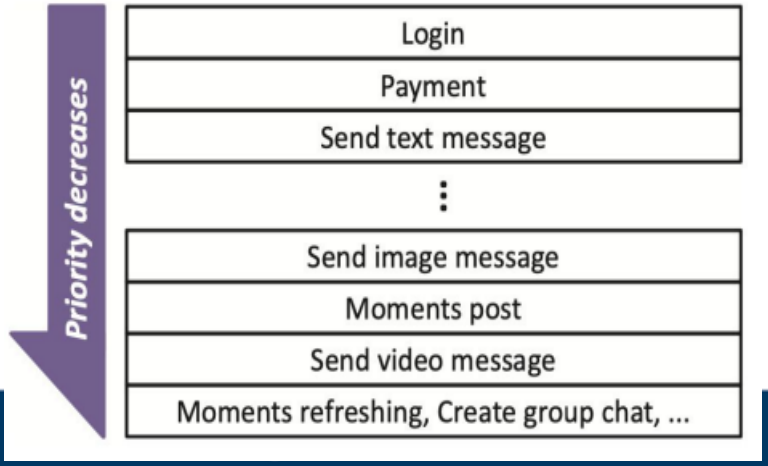
- Business-oriented Admission Control
- User-oriented Admission Control
- Session-oriented Admission Control
- Adaptive Admission Control
- Collaborative Admission Control



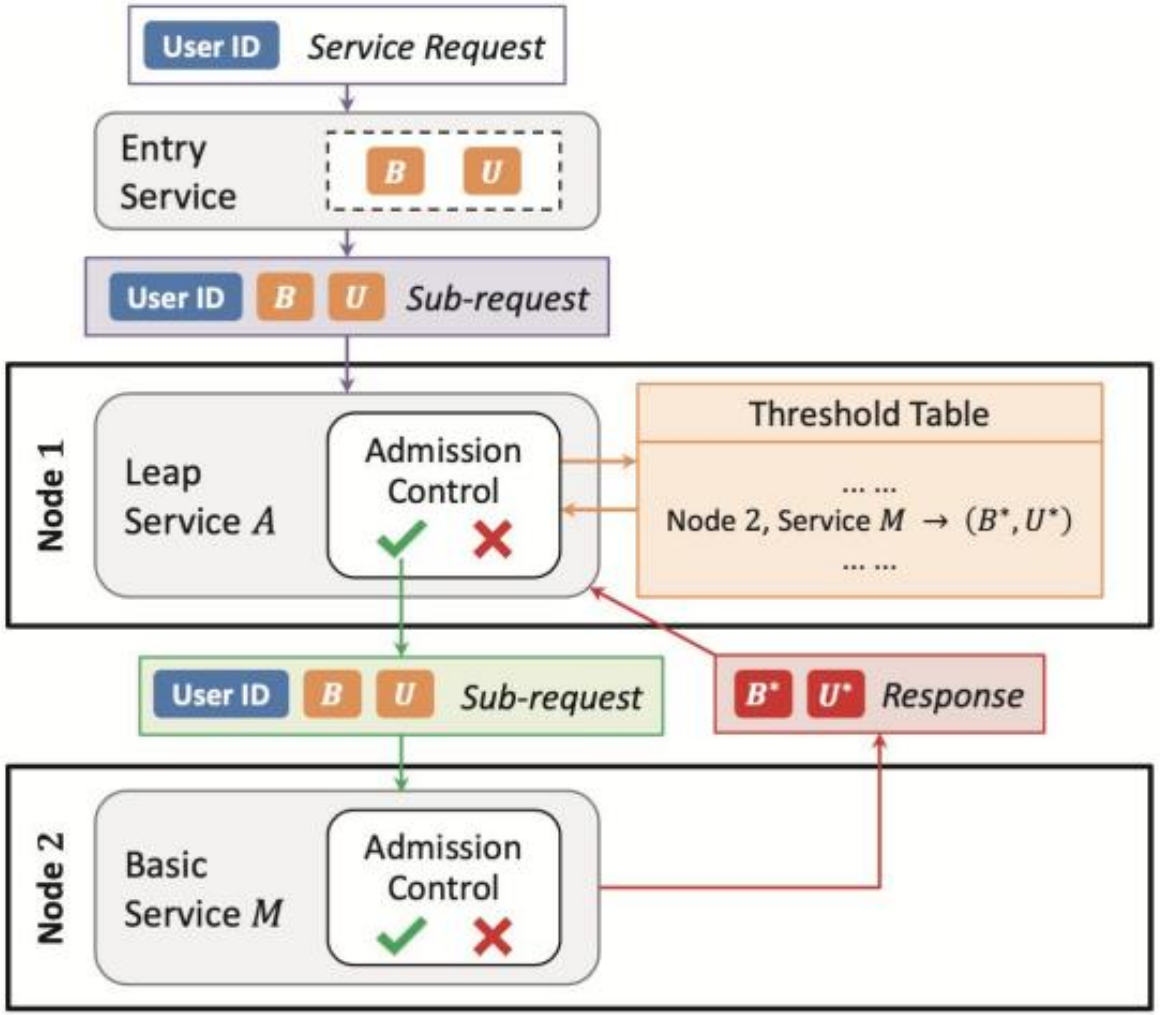


Business-oriented Admission Control

- Prioritized based on their business significance
- Subsequent requests inherit the same business priority
- Advantages:
  - Service agnostic: business priority is independent to the business logic of any service
  - Easy to maintain: business priority is assigned in the entry services & reflect the changes of basic and leap services

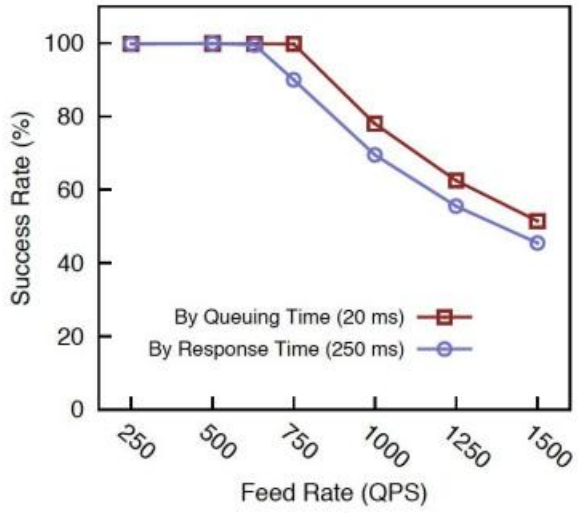


Service Admission Control Workflow

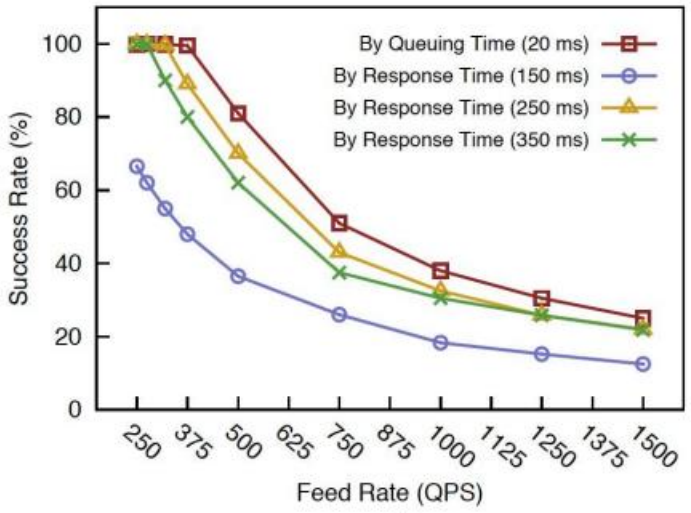


Evaluation – Queuing time vs. Response time

- Overload control in the large-scale microservice architecture must be **decentralized** and **autonomous** in each service
- The algorithmic design of overload control should take into account a **variety of feedback** mechanisms
- An effective design of overload control is always derived from the comprehensive profiling of the processing behavior in the actual workload



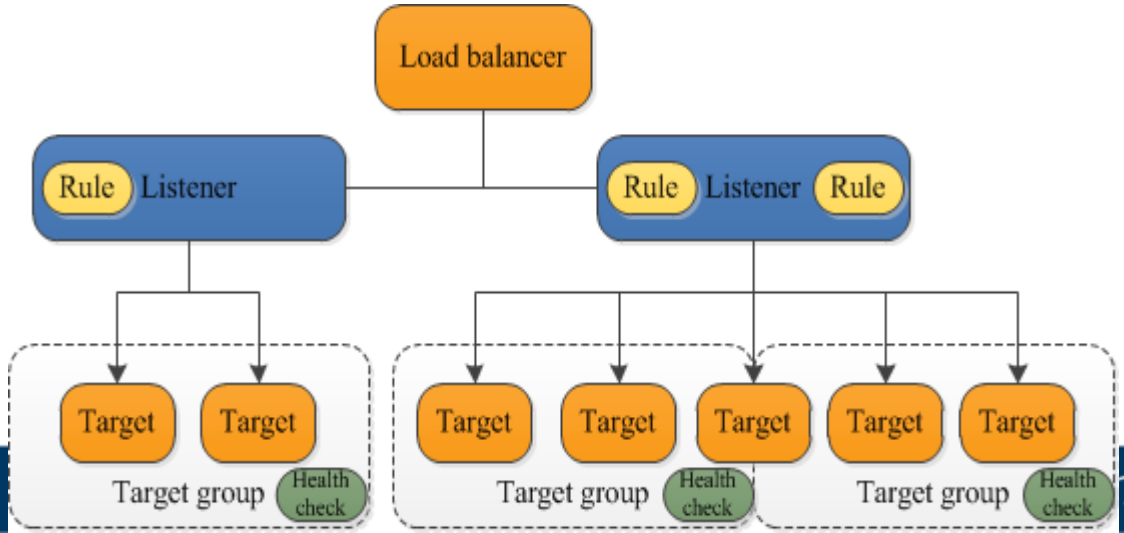
(a) $\mathcal{M}^1$



(b) $\mathcal{M}^2$

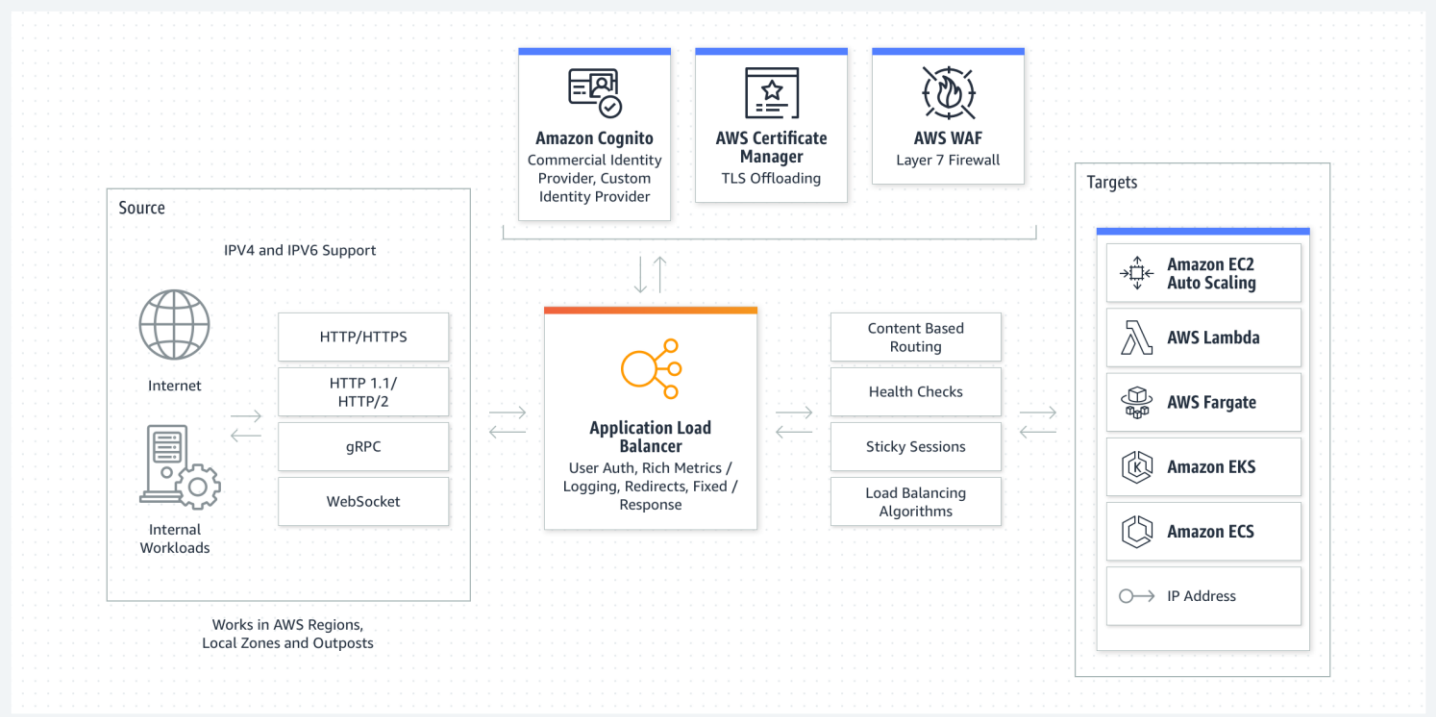
AWS Load Balancing Practice

- AWS Load Balancer Controller
  - A controller to help manage Elastic Load Balancers for a Kubernetes cluster, including Application Load Balancers, Network Load Balancers and etc.
  - A *load balancer* serves as the single point of contact for clients.
  - A *listener* checks for connection requests from clients
  - Each *target group* routes requests to one or more registered targets



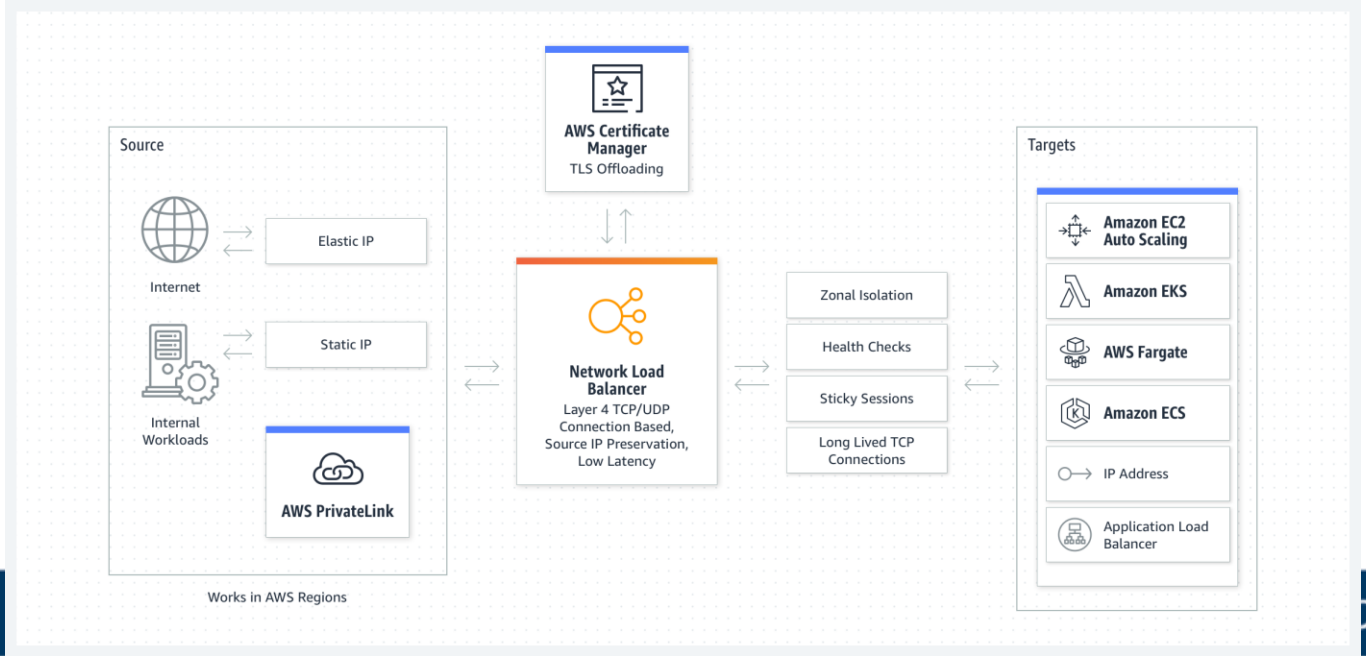
AWS Application Load Balancer

- Elastic Load Balancing automatically distributes incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones
- Elastic Load Balancing scales load balancer as the incoming traffic changes over time. It can automatically scale to the vast majority of workloads.



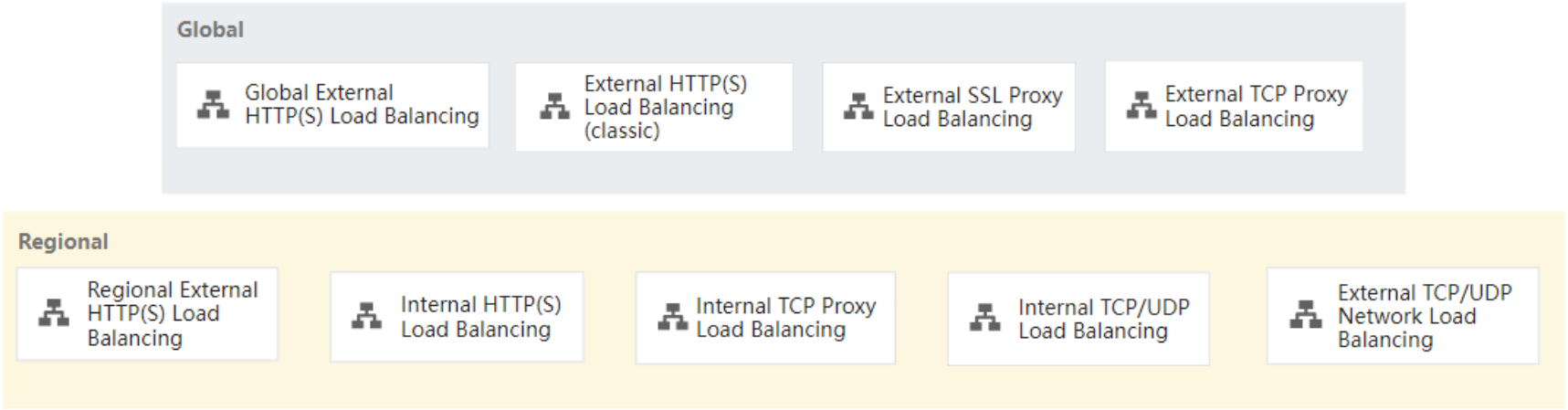
AWS Network Load Balancer

- A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second.
  - For TCP traffic, the load balancer selects a target using a flow hash algorithm based on the protocol, source IP address, source port, destination IP address, destination port, and TCP sequence number.
  - The TCP connections from a client have different source ports and sequence numbers, and can be routed to different targets.
  - Each individual TCP connection is routed to a single target for the life of the connection.



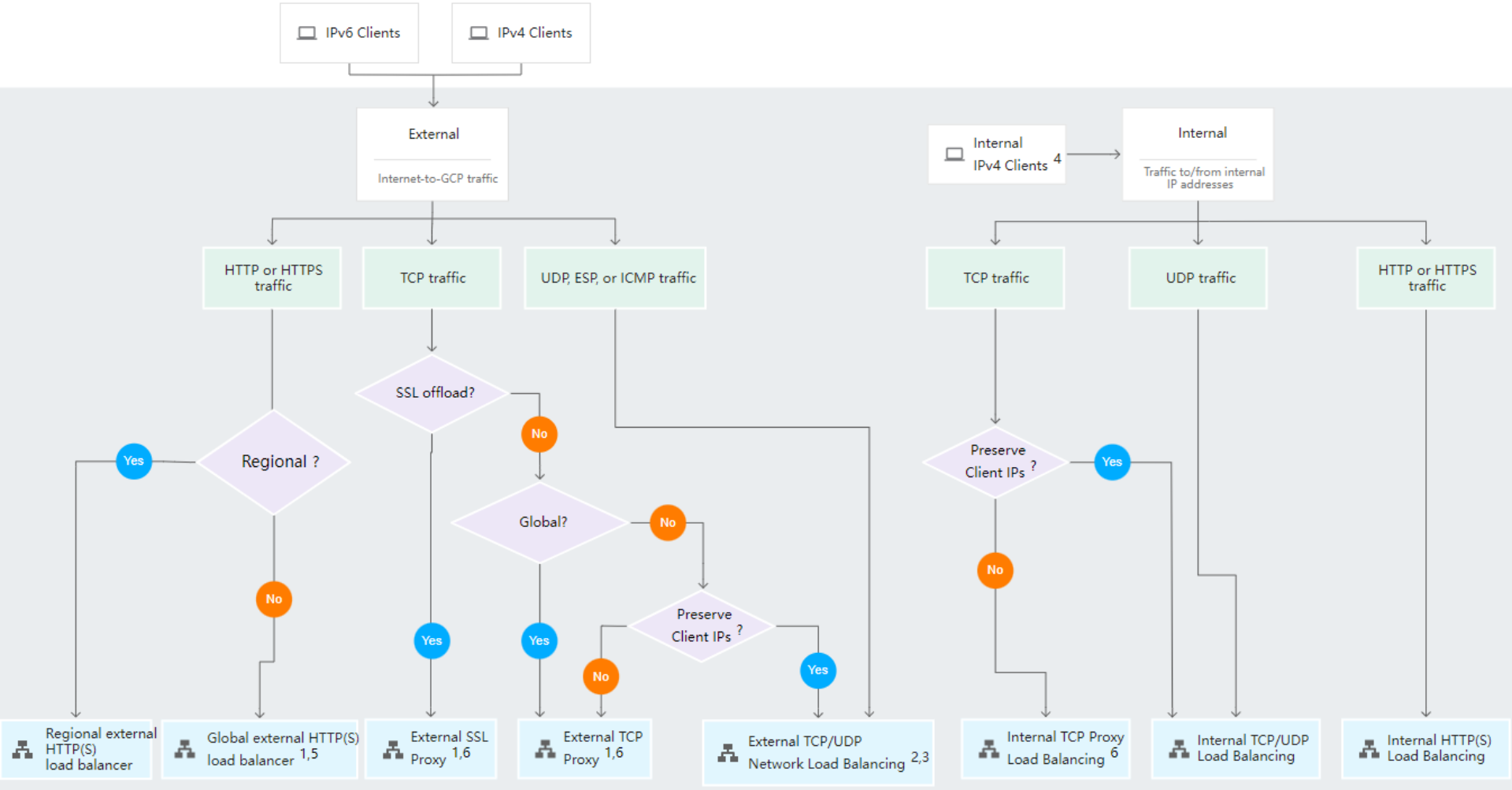
Google Cloud Load Balancing

- Load Balancing features of Google Cloud
  - Single anycast IP address
  - Software-defined load balancing
  - Seamless autoscaling
  - Layer 4 and Layer 7 load balancing
  - External and internal load balancing
  - Global and regional load balancing
  - Advanced feature support



Decision of a Load Balancer

- A decision tree is suggested by Google



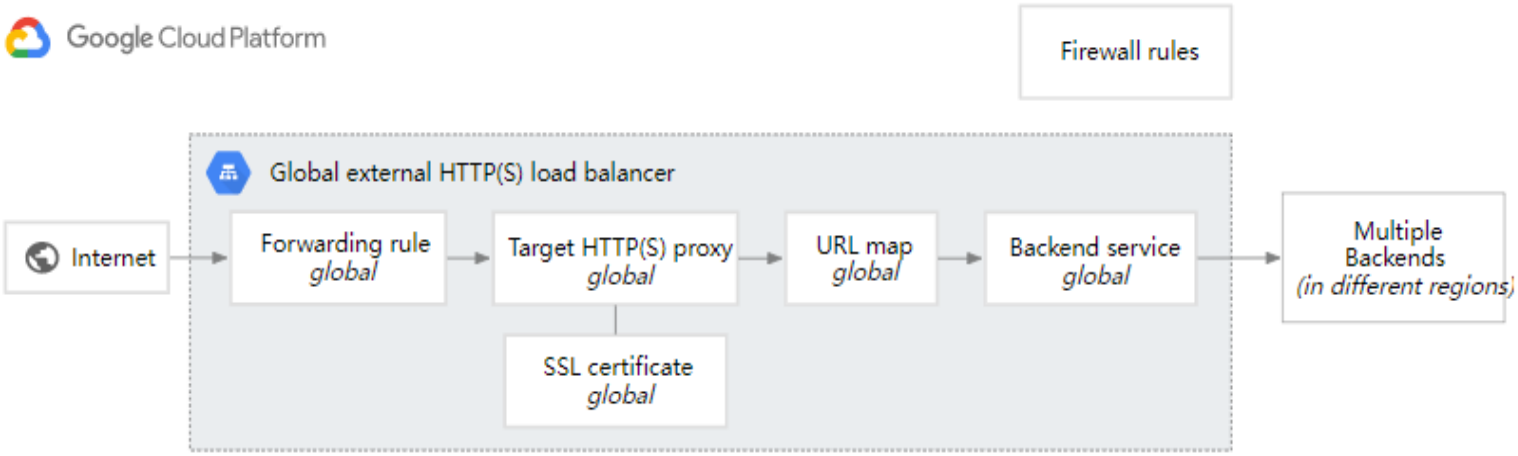
Relevant technologies of Google Cloud Load Balancer

- Google Front Ends (GFEs) are software-defined, distributed systems that are located in Google points of presence (PoPs) and perform global load balancing in conjunction with other systems and control planes.
- Andromeda is Google Cloud's software-defined network virtualization stack.
- Maglev is a distributed system for Network Load Balancing.
- Envoy proxy is an open source edge and service proxy, designed for cloud-native applications.

Load balancer	Underlying technology
Global external HTTP(S) load balancer	GFEs, Envoy
Global external HTTP(S) load balancer (classic)	GFEs
Regional external HTTP(S) load balancer	Envoy, Maglev
Internal HTTP(S) load balancer	Andromeda, Envoy
External TCP proxy load balancer	GFEs
External SSL proxy load balancer	GFEs
Internal regional TCP proxy load balancer (Preview)	Andromeda, Envoy
External TCP/UDP network load balancer	Maglev
Internal TCP/UDP load balancer	Andromeda

External HTTP(S) Load Balancing

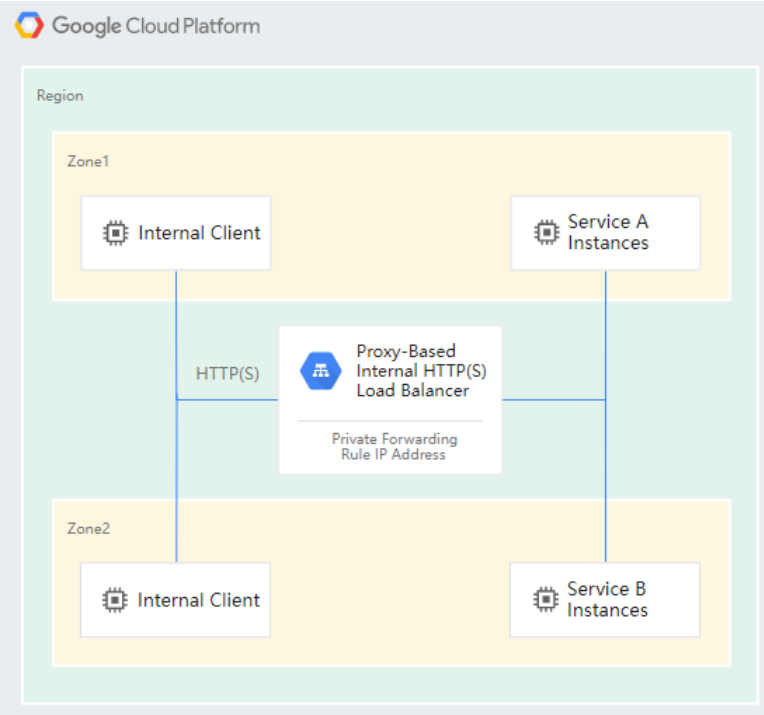
- External HTTP(S) Load Balancing is a proxy-based Layer 7 load balancer that enables you to run and scale your services behind a single external IP address.
- Forwarding rules route traffic by IP address, port, and protocol to a load balancing configuration consisting of a target proxy, URL map, and one or more backend services.



Global external HTTP(S) load balancer components

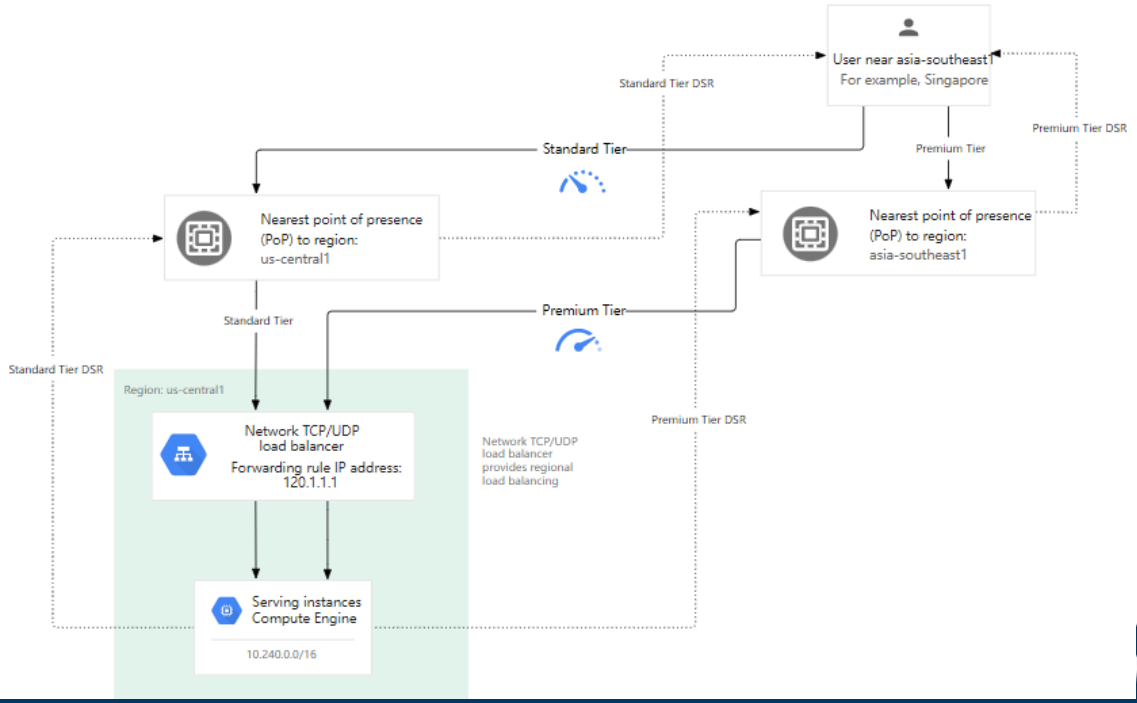
Internal HTTP(S) Load Balancing

- Google Cloud Internal HTTP(S) Load Balancing is a proxy-based, regional Layer 7 load balancer that enables you to run and scale your services behind an internal IP address.
  - An internal IP address to which clients send traffic
  - One or more backend services to which the load balancer forwards traffic



External TCP/UDP Network Load Balancing

- Google Cloud external TCP/UDP Network Load Balancing is a regional, pass-through load balancer. A network load balancer distributes external traffic among virtual machine instances in the same region.
- A network load balancer can receive traffic from:
  - Any client on the internet
  - Google Cloud VMs with external IPs
  - Google Cloud VMs that have internet access through Cloud NAT or instance-based NAT



Assignment Option 2

- For the teams who want to do the assignment (option 2) with Kubernetes:
- Read Kubernetes Cookbook:
  - https://github.com/k8s-cookbook/recipes
- Tutorial
  - https://kubernetes.io/zh-cn/docs/home/”
- Demo