Resource Management Policies

Neelam Singh

Programming Support of Google App Engine:

- Google App Engine primarily supports **Go, PHP, Java, Python, Node. js, . NET, and Ruby applications**, although it can also support other languages via "custom runtimes". The service is free up to a certain level of consumed resources and only in standard environment but not in flexible environment.

Policies and mechanisms for resource management

- A policy typically refers to the principal guiding decisions, whereas mechanisms represent the means to implement policies. Separation of policies from mechanisms is a guiding principle in computer science.
- **Cloud resource management policies can be loosely grouped into five classes:**
    - Admission control.
    - Capacity allocation.
    - Load balancing.
    - Energy optimization.
    - Quality-of-service (QoS) guarantees.

Policies Goal

- The explicit goal of an admission control policy is to prevent the system from accepting workloads in violation of high-level system policies; for example, a system may not accept an additional workload that would prevent it from completing work already in progress or contracted.
- Limiting the workload requires some knowledge of the global state of the system. In a dynamic system such knowledge, when available, is at best obsolete. Capacity allocation means to allocate resources for individual instances; an instance is an activation of a service. Locating resources subject to multiple global optimization constraints requires a search of a very large search space when the state of individual systems changes rapidly.
- Load balancing and energy optimization can be done locally, but global load-balancing and energy optimization policies encounter the same difficulties as the one we have already discussed. Load balancing and energy optimization are correlated and affect the cost of providing the services.

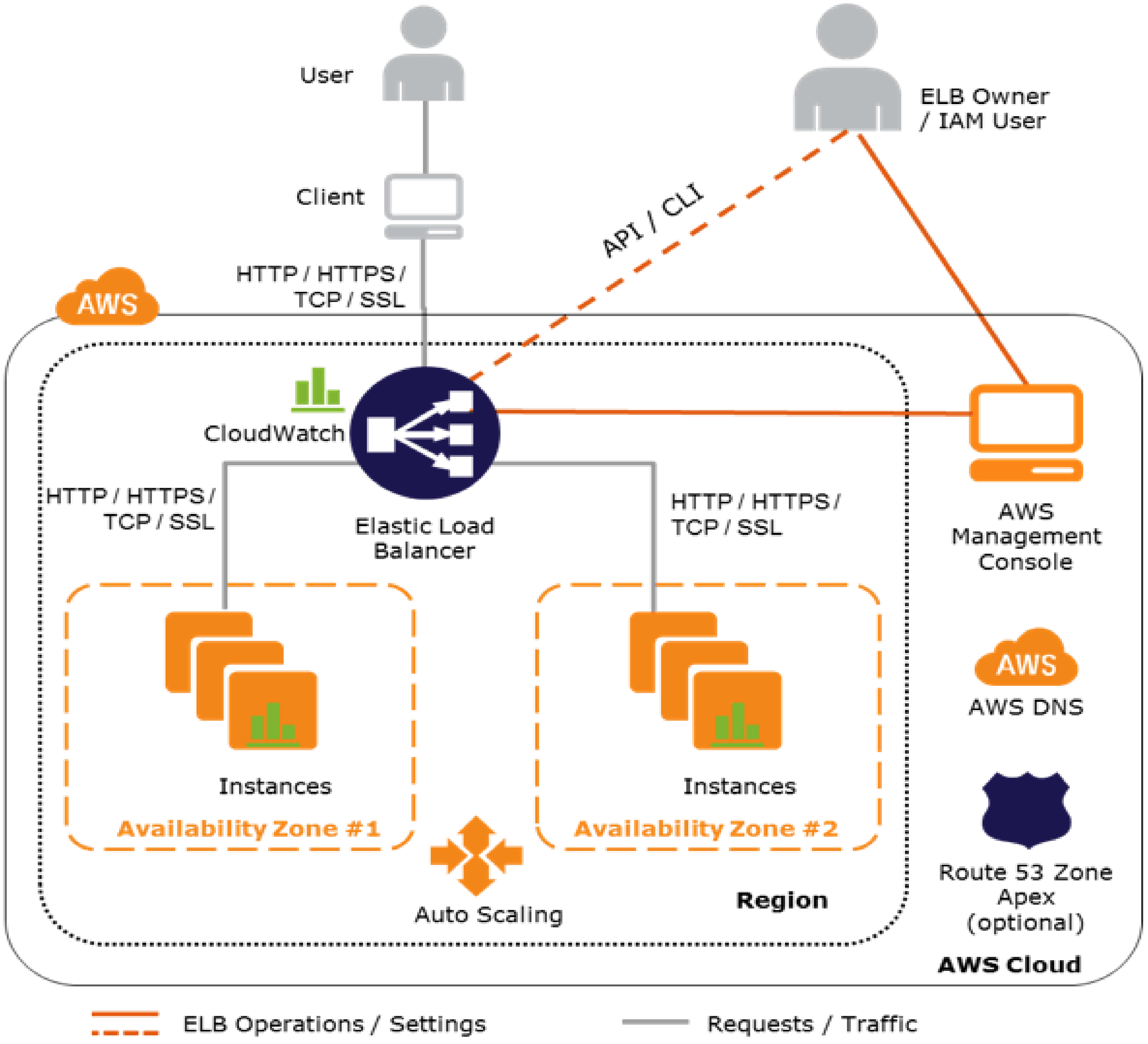
- Allocation techniques in computer clouds must be based on a disciplined approach rather than adhoc methods.
- The four basic mechanisms for the implementation of resource management policies are:
  - Control theory: Control theory uses the feedback to guarantee system stability and predict transient behavior but can be used only to predict local rather than global behavior.
  - Machine learning: A major advantage of machine learning techniques is that they do not need a performance model of the system. This technique could be applied to coordination of several autonomic system managers.
  - Utility-based: Utility-based approaches require a performance model and a mechanism to correlate user-level performance with cost.
  - Market-oriented/economic mechanisms: Such mechanisms do not require a model of the system, e.g., combinatorial auctions for bundles of resources

Applications of control theory to task scheduling on a cloud

- Control theory has been used to design adaptive resource management for many classes of applications, including power management, task scheduling, QoS adaptation in Web servers ,and load balancing.
- The classical feedback control methods are used in all these cases to regulate the key operating parameters of the system based on measurement of the system output; the feedback control in these methods assumes a linear time-invariant system model and a closed-loop controller.
- This controller is based on an open-loop system transfer function that satisfies stability and sensitivity constraints.
- The technique allows multiple QoS objectives and operating constraints to be expressed as a cost function and can be applied to stand-alone or distributed Web servers, database servers, high- performance application servers, and even mobile/embedded systems.

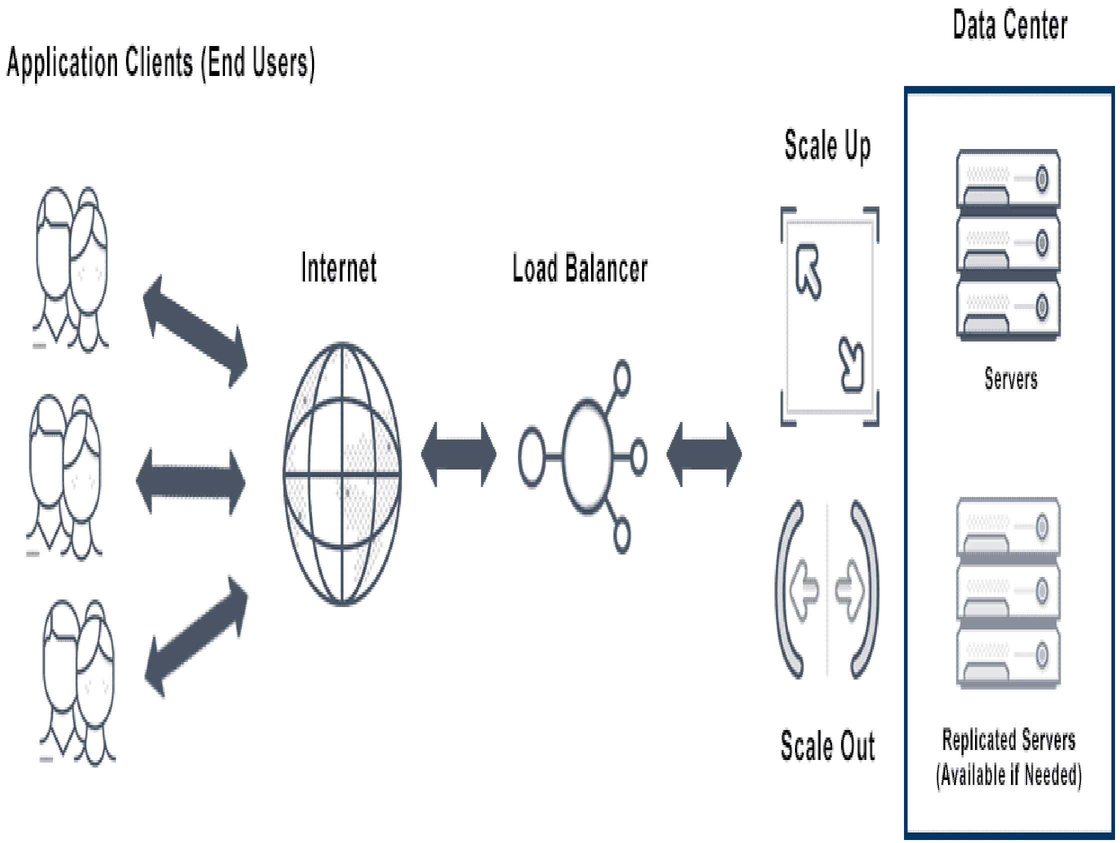
Elastic load balancing and auto scaling

- Elastic Load Balancing automatically distributes traffic from incoming applications through various destinations, such as Amazon EC2 instances, containers, IP addresses and Lambda functions. You can control the variable load of your application's traffic in a single zone or in several availability zones.
- Elastic Load Balancing offers three types of load balancers that have the necessary level of high availability, automatic scalability and security to make your applications tolerant of errors.



Auto scaling

- Auto scaling, also referred to as autoscaling, auto-scaling, and sometimes automatic scaling, is a cloud computing technique for dynamically allocating computational resources. Depending on the load to a server farm or pool, the number of servers that are active will typically vary automatically as user needs fluctuate.
- Auto scaling and load balancing are related because an application typically scales based on load balancing serving capacity. In other words, the serving capacity of the load balancer is one of several metrics (including cloud monitoring metrics and CPU utilization) that shapes the auto scaling policy.
- Auto scaling is a cloud computing feature that enables organizations to scale cloud services such as server capacities or virtual machines up or down automatically, based on defined situations such as traffic ir utilization levels. Cloud computing providers, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), offer autoscaling tools.


Application Clients (End Users)
Internet
Load Balancer
Scale Up
Scale Out
Data Center
Servers
Replicated Servers
(Available if Needed)