Adaptation in Cloud Computing

Term Project

Rohit Kumar

Sections

1. Preliminaries
2. Problem statement
3. Literature Survey
4. Preferred approach
5. My ideas on possible extensions
6. Key takeaways

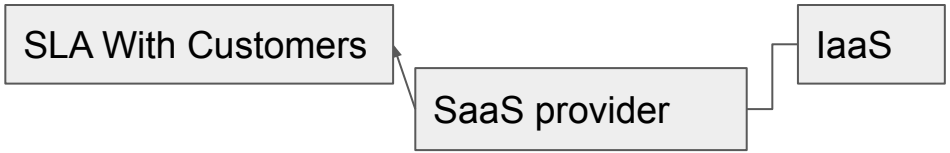
Preliminaries

Cloud Elasticity

Cloud Elasticity is defined as

the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible.

E.g



- ➔ Software-as-a-Service (SaaS) providers relying on Infrastructure-as-a-Service (IaaS)
- ➔ have the capability to quickly cope with highly and unpredictable demands by finely allocating resources accordingly
- ➔ Therefore meeting Service Level Agreements (SLAs) previously established with their customers

Horizontal elasticity

- Basically refers to the on-demand adding of a new VM
- Horizontal scaling is applicable for applications that have a clustered architecture.
    - with a gateway or a master node that distributes requests between the worker nodes (or VMs).
    - Adding or removing node from cluster is the elastic mechanism.
- Cost depends on the ease with which nodes can join or leave the cluster.
- We will notate as HS_{Infra} throughout the ppt.

Vertical elasticity

- Adding / deleting RAM or CPU to/from VM system.
- Modern hypervisors support online VM resizing allowing one to add CPU or memory resources to a VM without bringing it down.
- **Note 1:** vertical scaling is limited by the amount of free CPU cores and memory available on the physical server hosting the virtual machine.
- **Note 2:** Modern hypervisors also support **live migration** allowing virtual machines to be migrated from one physical server to another.
- We will notate as VS_{Infra} throughout the ppt.

Software elasticity

Software Elasticity is the capability of a software to adapt itself (ideally in an autonomic manner) to meet

- ➔ demand changes

  and/or

- ➔ infrastructure resources limitations
- ➔ We will notate as HS_{Soft} and VS_{soft} throughout the ppt.
  - ◆ HS_{Soft} - add or remove software components on the fly
  - ◆ VS_{Soft} - change the offering of existing components.

Three layer software model

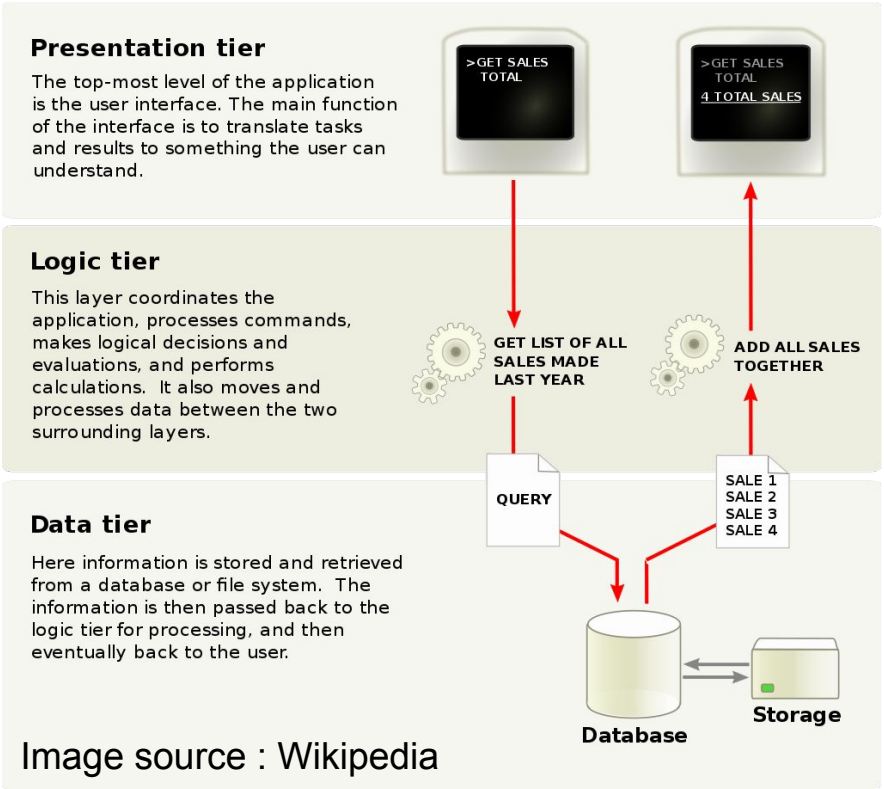


Image source : Wikipedia

- Modular view of software.
- Each unit can be handled in cloud environment in seperate ways.
- E.g. Google App Engine can be used only for logic tier/layer and Amazon Database service for data layer for the same software.

Difference b/w Elasticity mode and Method

Elasticity mode

- Mode (policy) refers to the needed interactions (or manner) in order to perform elasticity actions
- It handles the case of how the need for adaptation is calculated.
- Reactive mode
  - Static threshold
  - Dynamic threshold
- Proactive mode
  - Time series analysis:
  - Model solving mechanisms
- Model predictive control (MPC)
  - Reinforcement Learning (RL)
  - Control theory

Elasticity Methods

- Once elasticity policy have decided for more resources, elasticity methods handles how the demanded resource will be provided.
- To deploy the elasticity solutions, one or hybrid of the following methods is implemented to meet the demand.
    - Infrastructure
        - Horizontal scaling
        - Vertical scaling
    - Software level
        - Software Elasticity

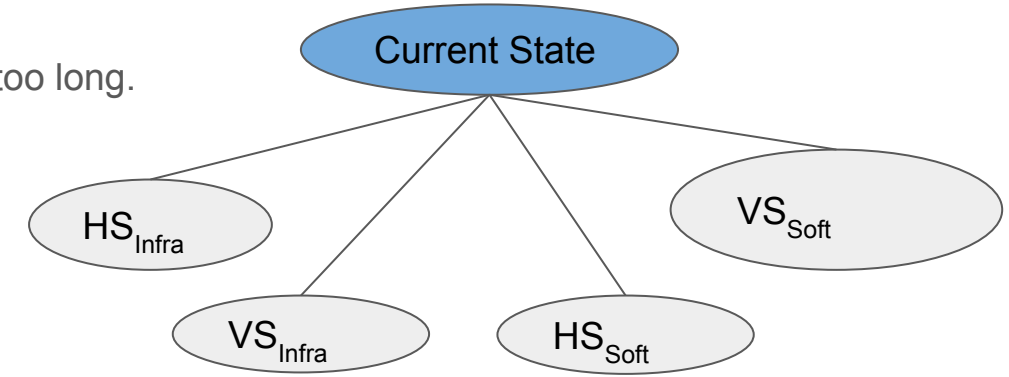
Problem Statement

Cloud Computing

- The cloud supports online provisioning / deprovisioning of resources provides a flexible system of resources for the softwares.
- It helps in robust handling of dynamic traffic that service provider face from day to day.

Problem Statement

- The problem is to select one of the possible method such that it is optimal.
    a. Optimal in the sense that it satisfies over the following recognised challenges
        i. Resources are limited
        ii. Resource initiation time can be too long.
        iii. Partial usage waste.



Literature Survey

Papers Reviewed

1. How to Adapt Applications for the Cloud Environment
    a. Presents a review of field.
    b. Three layer software development model as base and studies the adaptation layerwise.
        i. Data layer
        ii. Business layer
        iii. Presentation layer
    c. Addresses the research questions and open challenges with respect to adaptation of each layer.

Papers Reviewed

- SmartScale: Automatic Application Scaling in Enterprise Clouds
- Summary :
  - Addresses the choice of horizontal scaling and vertical scaling as an optimisation problem
  - Optimisation to minimize the joint cost of horizontal scaling and vertical scaling
  - Uses binary search to find the minima.

Papers Reviewed

- SmartScale: Automatic Application Scaling in Enterprise Clouds
- Issues not recognised :
    - Resource initiation time - what if vertical and horizontal scaling initiation time is high ?

Papers Reviewed

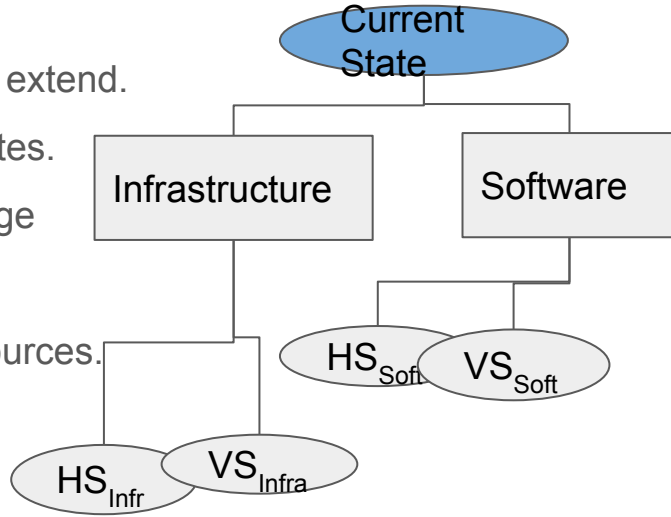
- Experimental Analysis on Autonomic Strategies for Cloud Elasticity
  - Addresses the various challenges in cloud computing adaptation
    - Resource limitation, initiation time, partial usage wastage.
  - E.g consider the case that there is peak demand of resources for minutes but initiation time is in of the order of minutes.
    - Think of the implications.
  - Addresses a new adaptation that could help with the high initiation time
    - Software elasticity
  - Uses MAPE-K loop.

Papers Reviewed : Experimental Analysis on Autonomic Strategies for Cloud Elasticity

- There are two types of resources at our hand to use for adaptation
  - Infrastructure and Software elasticity
  - Each of the above have their Horizontal and vertical Components.

HS_{Infra}
VS_{Infra}
HS_{Soft}
VS_{Soft}

- Infrastructure resources
  - are limited. I.e there is a threshold beyond which we can't extend.
  - initiation time is significant i.e maybe of the order of minutes.
  - Pricing is per instance-hour : leads to partial usage wastage
- Software resources
  - Initiation time is negligible compared to infrastructure resources.



a

Papers Reviewed : Experimental Analysis on Autonomic Strategies for Cloud Elasticity

Benefits of this complementary usage of IaaS and SaaS elasticities

- Alleviate the use of infrastructure resources.
- Improve responsiveness of scaling
- Improve expression capabilities of elasticity.

Table I
CLOUD ELASTICITY SCALING ACTIONS

Scaling Dimension	API Name	Description
Infrastructure Horizontal Scaling (HS_{infra})	Scale Out Infrastructure (SO_{infra})	Add VM(s) to the pool
	Scale In Infrastructure (SI_{infra})	Remove VM(s) from the pool
Infrastructure Vertical Scaling (VS_{infra})	Scale Up Infrastructure (SU_{infra})	Increase offering (Off_{vm}) of existing VM(s)
	Scale Down Infrastructure (SD_{infra})	Decrease offering (Off_{vm}) of existing VM(s)
Software Horizontal Scaling (HS_{soft})	Scale Out Software (SO_{soft})	Add software component(s) to the application
	Scale In Software (SI_{soft})	Remove software component(s) to the application
Software Vertical Scaling (VS_{soft})	Scale Up Software (SU_{soft})	Increase offering (Off_{comp}) of existing component(s)
	Scale Down Software (SD_{soft})	Decrease offering (Off_{comp}) of existing component(s)

Papers Reviewed : Experimental Analysis on Autonomic Strategies for Cloud Elasticity

The model

A. Autoscaling Service.
    a. Manages the elasticity of resources.
    b. Person Incharge of this - CA : Cloud Administator.
    c. CA needs a way to monitor and act on both Iaas and SaaS resources,

Papers Reviewed : Experimental Analysis on Autonomic Strategies for Cloud Elasticity

The model

B. Model overview

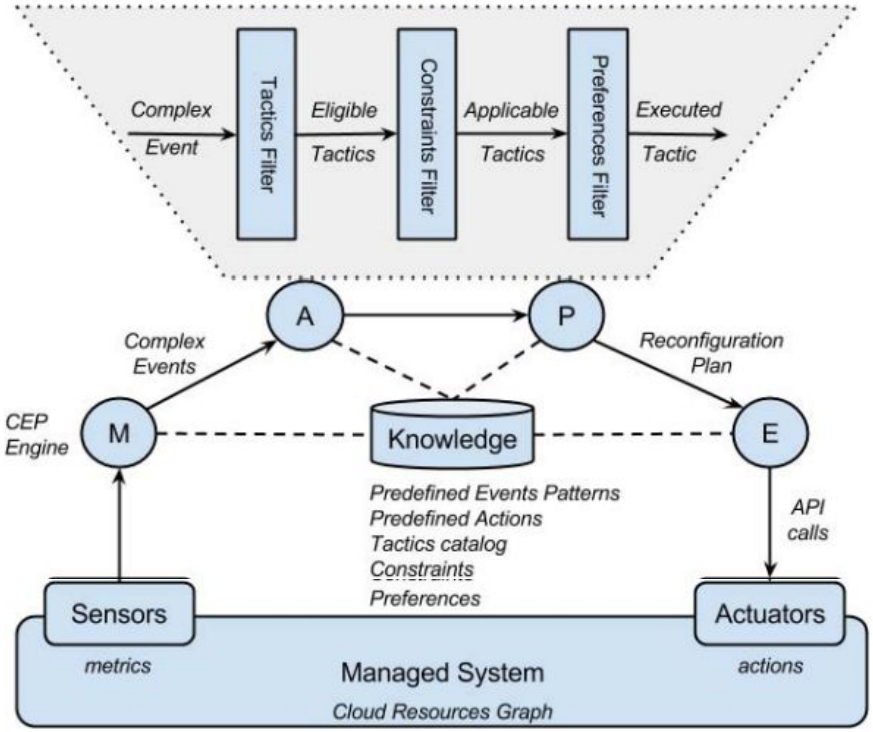


Figure 4. Cloud Resources Model Overview

Papers Reviewed : Experimental Analysis on Autonomic Strategies for Cloud Elasticity

C. Managed system

- An n-tier web applications
- Each tier is hosted on a VM
- VM's are hosted on a set of Physical Machines (PM)

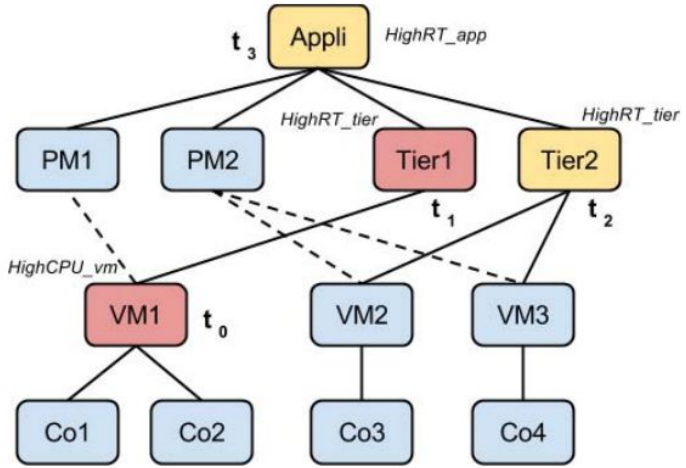
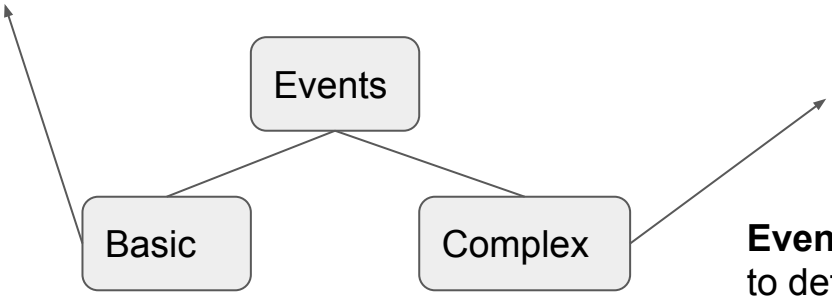


Figure 5. Instance of cloud resources model (managed system)

Papers Reviewed : Experimental Analysis on Autonomic Strategies for Cloud Elasticity

D. Monitor : Events

- Each cloud resource has a set of metrics indicating its health.
- The monitor phase collect and aggregate multiple sensor values over time.
- This helps system to make timely relevant information about the system health in form of *events.*
- An example of event:
  - if CPU consumption of one VM has exceeded a maximum threshold, generate a HighCPU_vm event.



Complex events are generated by composition of basic event.

Event patterns :
to detect complex relationships between events overtime by defining correlation rules also known as Event Patterns.

CA needs to observe and define these.

Papers Reviewed : Experimental Analysis on Autonomic Strategies for Cloud Elasticity

E. Execute : predefined actions

- Actions : Basic actions and complex actions
- The goal of the complex actions is to absorb the infrastructure resource initiation time.

a SD_{soft} (e.g., switching the Offcomp from Video HD to Image - step 2')

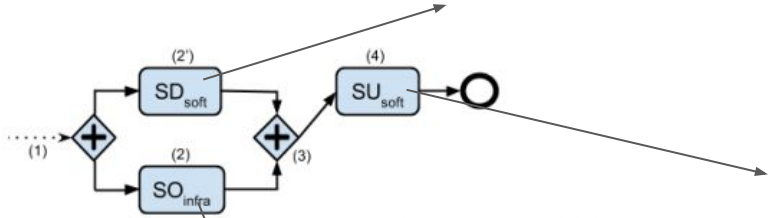


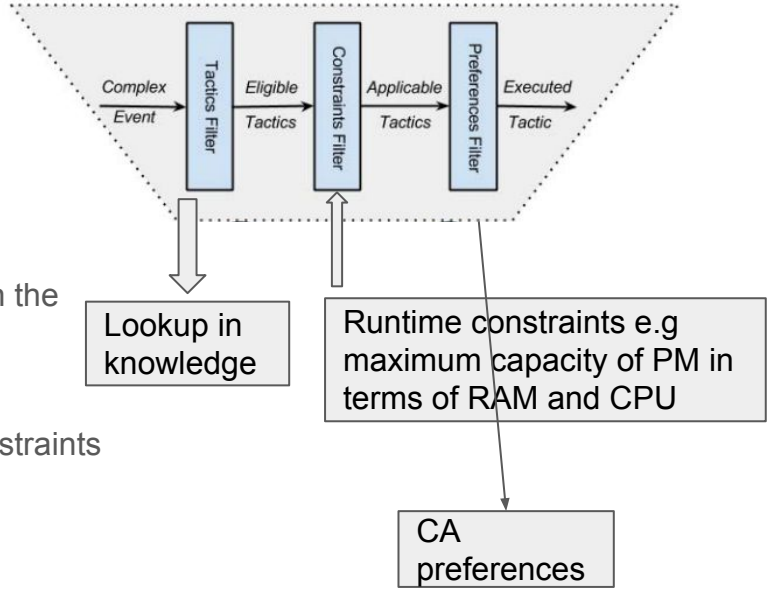
Figure 6. Complex action example: $(SD_{soft} \parallel SO_{infra})\ ;\ SU_{soft}$

SU_{soft} (e.g., switching back to Video HD - step 4)

SO_{infra} (e.g., adding one VM to the pool - step 2)

Papers Reviewed : Experimental Analysis on Autonomic Strategies for Cloud Elasticity

F: Analyse and Plan : Reconfiguration decision



Tactic filter : A tactic is actually an event-action pair defined by CA and representing a well-known IF-THEN statement.

- the Tactics Filter takes a look at the event-action mapping stored in the Knowledge which was given by CA

Constraints Filter – Context: The runtime context can induce some constraints that may prevent to execute eligible tactics.

Preferences Filter – Strategy: based on CA preferences, final action to take.
Some criteria: Cost, QoS, QoE, Elasticity time, Responsiveness

Papers Reviewed : Experimental Analysis on Autonomic Strategies for Cloud Elasticity

G: Knowledge

- data shared among all MAPE phases such as
  - current cloud resources graph with associated metrics values,
  - runtime constraints
  - preferences expressed on the system.
- event patterns and predefined actions
- Tactics

Additional Papers

1. A Survey on Cloud Computing Elasticity - Link
2. Elasticity in Cloud Computing: What It Is, and What It Is Not - Link
3. Horizontal and Vertical Scaling of Container-Based Applications Using Reinforcement Learning - Link
4. Elasticity in Cloud Computing: State of the Art and Research Challenges - Link

Preferred Approach

- Smart scale does not take into account the infrastructure resource initiation time.
- There have been few work in cross-layered adaptation.
- Integrating the SaaS layer elasticity with infrastructure elasticities is one possible approach to handle it.

My ideas on possible extensions

- Discussed paper
  - There is no optimisation in resource selection at runtime.
  - CA have a quite big role.
    - He have to decide preferred actions by test runs at different constraints and other factors.
    - Need of a team to manual handle the optimisation and changing system parameters iteratively.

⇒ Smartscale paper proposes one such algorithm that can be used.

⇒ The human intervention can be minimised or the decision algorithm can be based on some algorithm that adapts itself with time.

⇓

RL

Cross-layered optimisation

- The resource initiation time is a physical limit in the cloud systems.
- Algorithmically, cross-layered approach to handle the issue seems a valid direction to work towards.
- There is not much work in this direction.
  - One other work is *Rainbow*
- Using RL with human in loop can be a viable solution that
  - Minimises the human intervention.
  - Optimises for the resource selection.

Key Takeaways

- Cloud paradigm has changed the way we work around now. E.g google drive, onedrive etc.
- The change for the softwares is quite huge.
- It poses several options and we need to select best for us, as we are paying for the same.
- Hence, a need for optimisation in resource selection.

- SmartScale - infrastructure layer based only.
- Infrastructure resource initiation time is high.
- Cross-layered elasticities is one possible solution.