Business Driven IT Management in Cloud Computing Systems

Dipesh Chauhan (supervisors: Prof. Alan Fekete, Dr. Vladimir Tosic)

School of Information Technologies

FACULTY OF ENGINEERING & INFORMATION TECHNOLOGIES

The Problem

- System management of cloud computing systems is complex because of the diversity, complexity, and scale of cloud systems
- System management is typically done using technical metrics (e.g. CPU utilisation), which may not optimise business metrics (e.g. profit)
- Current cloud management tools have many limitations (e.g. vendor lock-in)
- Management actions and events/metrics are different across cloud types and vendors

The Goal

- Provide a vendor-independent cloud management solution that automatically chooses and executes management actions that are best from the business perspective

The Solution

- Extended WS-Policy4MASC to support cloud management actions and events/metrics
- MiniZnMASC extension through proof of concept prototype that managed various cloud systems (e.g. EC2 and Rackspace) and allows for seamless switching

Methodology

- Reviewed 8 most-relevant academic papers & 10 major industrial cloud management tools
- Reviewed functionality of major cloud systems (e.g. Amazon EC2, Microsoft Azure, Salesforce) across IaaS, PaaS, and SaaS
- Identified and classified (according to CRUD - create, read, update, delete) core common management actions and events/metrics
- Mapped cloud vendor functionality to create a generic classification of cloud adaptions and events
- Designed, implemented, and tested a policy language and policy-driven management middleware

Evaluation

- Generality – Mapped generic methods to vendor specific methods (Table 1). Created a high level policies that allowed for seamless switching between various cloud vendors.
- Feasibility – Proof of concept prototype that connect MiniZnMASC to EC2 and Rackspace

WS-Policy4MASC Methods	Amazon EC2 Methods
Severs	
ListServers()	describeInstances()
CreateServer(ImageId, SizeId)	runInstances(RunInstanceRequest)
DescribeServer(ServerId)	describeInstances(DescribeInstancesRequest)
UpdateServer(ServerId, SizeId)	modifyInstanceAttribute(ModifyInstanceAttributeRequest)
DeleteServer(ServerId)	terminateInstances(TerminateInstancesRequest)
StartServer(ServerId)	startInstances(StartInstancesRequest)
StopServer(ServerId)	stopInstances(StopInstancesRequest)
RebootServer(ServerId)	rebootInstances(RebootInstancesRequest)
Images	
ListImages()	describeImages()
CreateImage(ServerId)	createImage(CreateImageRequest) registerImage(RegisterImageRequest)
DescribeImage	describeImages(DescribeImagesRequest)
DeleteImage(ImageId)	deregisterImage(DeregisterImageRequest)
Sizes	
ListSizes()	-
Processes	
ListProcesses(ServerId)	SSH top
StartProcess(ServerId, Command)	
TerminateProcess(ServerId, ProcessId)	SSH kill -9 <ProcessId>
RestartProcess(ServerId, ProcessId)	-
Objects	
UploadObject(ServerId, Path, Object)	SCP
ObjectExists(ServerId, Path)	SSH ls
GetObject(ServerId, Path)	SCP
DeleteObject(ServerId, Path)	SSH rm
Metrics	
ListMetrics()	listMetrics()
GetMetricValue(ServerId, Metric)	getMetricStatistics(GetMetricStatisticsRequest)

Table 1: Mapping between WS-Policy generic methods and Amazon

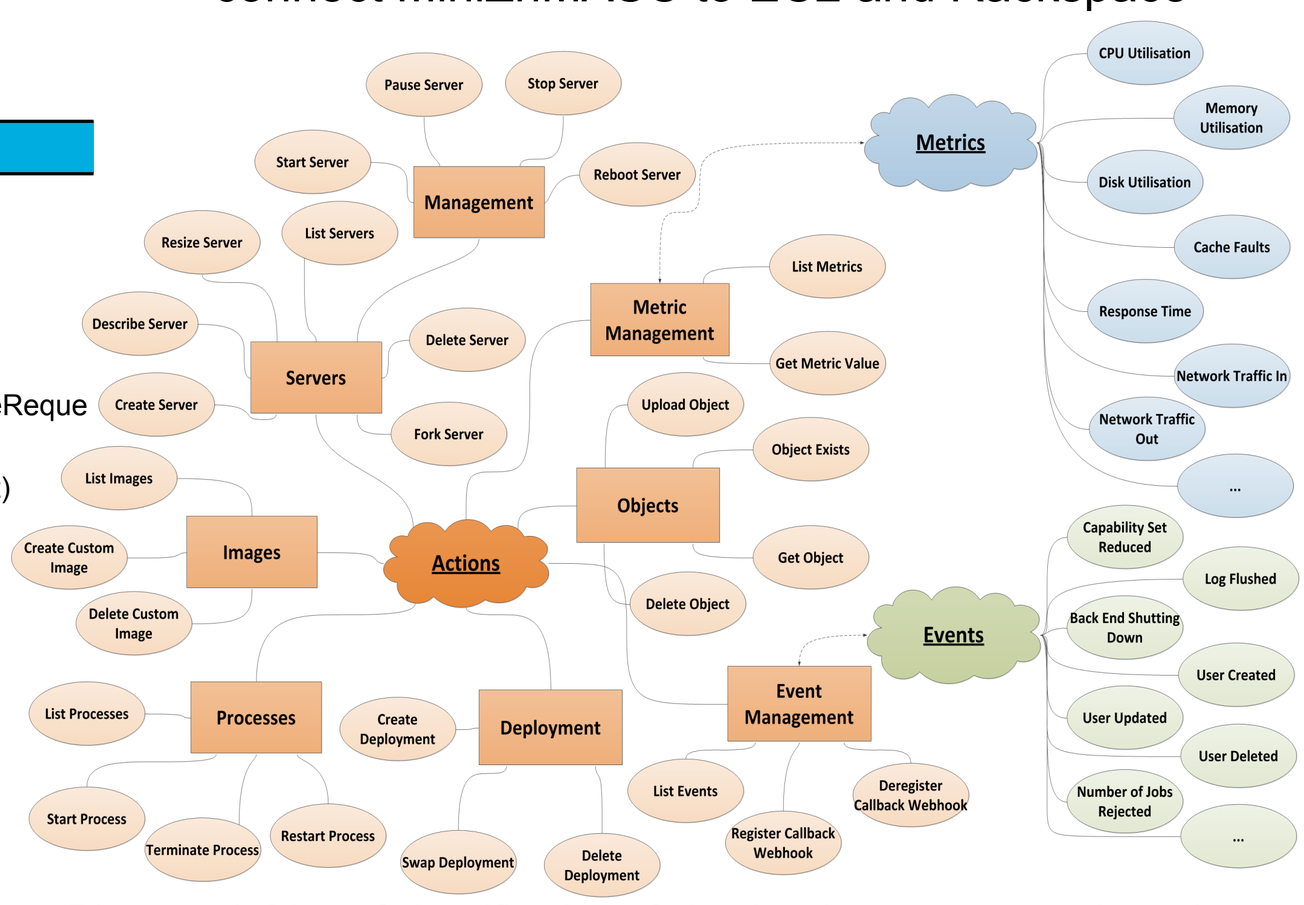

Diagram 1: Map of classification of cloud actions, events and metrics

Diagram 2: High level overview of end-to-end interaction and integration of MiniZnMASC with the cloud

THIS RESEARCH IS SPONSORED BY NICTA