Fine-Grained Elasticity Support for Cloud Applications: The CELAR Approach

(Marios Dikaiakos)

University of Cyprus
Cloud Computing

- Ubiquitous, convenient, on-demand network access
- Shared pool of configurable resources
- Rapid provisioning and release
- Minimal management effort
- Minimal interaction with the service provider

The NIST Definition of Cloud Computing, NIST, 2011
## Cloud Models

### Service Models
- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

### Deployment Models
- Private
- Community
- Public (IaaS)
- Hybrid

*The NIST Definition of Cloud Computing, NIST, 2011*
A Public Utility

...Computing may someday be organised as a public utility, just as the telephone system is organised as a public utility...

John McCarthy, 1961

(Cloud) Computing is dubbed as the 5th utility

R. Buyya et al., 2009
THE HISTORY OF THE CLOUD

John McCarthy introduces mainframe timesharing

1969

ARPANET by J.C.R. Licklider

1970

Virtualisation software launched

1991
WWW launched

1997

Cloud Computing is defined by Prof. Ramnath Chellappa

1999

Amazon launches Elastic Compute cloud (EC2), Simple Storage Service (S3)

2006

2013

£78bn Worldwide Public Cloud Services Market*

2014

£103.8bn estimated global cloud spending**

* Gartner, ** Constellation Research

celar

Marios Dikaiakos, University of Cyprus
Amazon Web Services is a $5 billion business, and it’s growing 50% a year.

Amazon Web Services quarterly sales and operating income

Revenue

Operating income

Quartz | qz.com

Data: Amazon
5 Essential Cloud Characteristics

The NIST Definition of Cloud Computing, NIST, 2011
OUTLINE

• Cloud Computing
• Main topics:
  • Elasticity
  • Application Management
• CELAR Architecture
• Elasticity and Monitoring - JCatascopia
• c-Eclipse and CAMF
• Conclusions
Elasticity

- Ability of a system to **expand** or **contract** its dedicated resources to meet the current demand

- Stakeholders state that **elasticity** (54%) and **cost reduction** (48%) are driving cloud adoption [FOC Survey 2013]
Three types of Elasticity

- **Horizontal**: On-demand provisioning
  - E.g.: Get more VMs

- **Vertical**: Resource re-configuration
  - E.g.: Main memory ballooning

- **Live Migration**: 
  - E.g.: Move to another provider
Elasticity controller

If (metricA \(>\) X) then add VM
else if (metricA \(<\) X) then remove VM
else if (metricB \(>\) Y) then increase RAM

...
OUTLINE

• Cloud Computing

• Main topics:
  • Elasticity
  • Application Management

• CELAR Architecture
• Elastic Monitoring - JCatascopia
• c-Eclipse and CAMF
• Conclusions
Emerging cloud applications

• Increasing complexity

• Dynamic behaviour

• A variety of deployment platforms with different:
  • configuration mechanisms
  • offered services
  • availability and pricing
  • elasticity capabilities
Use Case I: Cancer Genome Detection

- CPU and disk I/O intensive
- Memory intensive
- Disk I/O and memory intensive
“Analysing Cancer Genomics in the Elastic Cloud” Smowton, Balla, Antoniades, Miller, Pallis, Dikaiakos, Xing CCGridLife2015, May 4, Shenzhen, CN
Use case II: DataPlay

**Requirements:**
- Response time < 1.5s

**Elasticity capabilities:**
- Horizontal and Vertical Scaling
  - Allocate/Deallocate VM with more/less resources hosting Rendering and User Control instances
  - Add/Remove resources from VMs hosting Rendering and User Control instances

**Elasticity capabilities:**
- Horizontal and vertical scaling of single data storage system
  - Allocate/Deallocate VM with more/less resources hosting data processing node
- Horizontal and vertical scaling of entire data layer
  - Allocate/Deallocate separate data storage system instance

**Overall Requirements:**
- Minimize Cost

**Requirements:**
- Response time < 1.5s

**Elasticity capabilities:**
- Horizontal and Vertical Scaling
  - Create/Destroy Data Processing service instance running in a VM hosting other Data Processing instances
  - Create/Destroy Data Processing service instance running on a separate VM

**Elasticity capabilities:**
- Deployment elasticity
  - Deploy and Run in a virtual machine hosting another DATA PLAY components, or a standalone VM

**Requirements:**
- I/O performance >= 100MB/s
Use Case II: DataPlay (serious games)
Be the first to discover a new data story with
NHS Spending 2012 suggested correlations with 261,811 datasets

A&E Spend London in correlation to Crime Rate London

Primary datasource: NHS Spending 2012 - London
Secondary datasource: Met Police Crime Rate 2012
Pattern id: 145044111

Want to explore this correlation further, maybe find a deeper pattern in the data? Click the "Explore" button now.
Cloud application management challenges

- Time-consuming - requires manual effort
- Steep learning curve
- Relies on vendor-specific tools
- Offers limited portability
- Migration entails significant cost

Cloud management platforms

- Integrated software stacks for the management of Cloud environments [Gartner]
- Goal: ease the description & deployment of applications over Cloud infrastructures
- A critical component to the overall success of a cloud initiative [Gartner, CMP Landscape, 2012]
- Expected to provide application portability: “define once, deploy anywhere”
Cloud management platforms

- Requirements
  - Self-service interface
  - Provision system images
  - Metering and billing
  - Include service catalogs
  - Support configuration of resources
  - Policy-based workload optimisation
  - Application Monitoring
  - etc
OUTLINE

• Cloud Computing
• Main topics:
  • Elasticity
  • Application Management
• CELAR Architecture
• Elastic Monitoring - JCatascopia
• c-Eclipse and CAMF
• Conclusions
Fully Automated
Intelligent Decision Making Algorithms
Multi-Dimensional Control
Multi-Grain Elasticity Control
Multi-layer Scalable Monitoring
Application Management
Vendor Neutrality
Open-Source
www.celarcloud.eu
CELAR Architecture

CELAR is deployed around the Cloud Infrastructure

www.celarcloud.eu
OUTLINE

• Cloud Computing
• CELAR main topics:
  • Elasticity
  • Application Management
• CELAR Architecture
• Elasticity and Monitoring - JCatascopia
• c-Eclipse and CAMF
• Conclusions
Elasticity controller

In general, elasticity constraints are too complex for users and based on low-level metrics
Elasticity Control Estimation and Evaluation

• How should we interpret a sudden drop in request throughput at the business tier of a 3-tier cloud service?

A. There are **less clients**; this makes the business tier inefficiently utilised
   • Right Decision: Remove an Application Server

B. Video storage **backend under-provisioned**, requests are getting queued at business tier
   • Right Decision: Add another Database Node

Elasticity Controller with simple IF-THEN-ELSE policies based on metric violations cannot determine the right ECP to improve QoS or cost
Current Elasticity Controllers

- **Manual or semi-automated elasticity control**

- **Vendor-specific**

- Elasticity modelled as a one-dimensional property
  - No control over **cost, performance** and **quality**

- Only “horizontal” elasticity control
  - e.g. add/remove virtual instances
CELAR Elasticity Building Blocks

- Conceptualising and modelling elastic objects and execution environments
- Defining and capturing elasticity primitive operations associated with elastic objects
- Programming elastic objects and actions
- Runtime deployment, control and monitoring techniques for elastic objects

Source: Hong-Linh Truong (TUW), Cloudcom ‘14
Elasticity in CELAR

Dependency Graph capturing application structure and enabling a multi-level specification of elasticity: Application unit, service topology, application

Elasticity requirement specification - SYBL: Monitoring, constraint, strategy

Multi-dimensional Modeling & analysing Elasticity of Cloud Services - MELA: Resources, cost, quality

Multi-level elasticity control through rSYBL
Elasticity Policies Specification in SYBL

- **SYBL** language enables elasticity requirements description for Cloud applications
- Elasticity specification at different levels
  - Component, composite component, application
- Two types of SYBL elasticity requirements:
  - **Constraint**: “Constraint 1: CPU\_Usage < 80%”
  - **Strategy**: “Strategy 1: CASE Violated (Constraint 1) : Scale\_Out”

---


#SYBL.CloudServiceLevel
Cons1: CONSTRAINT responseTime < 5 ms
Cons2: CONSTRAINT responseTime < 10 ms
WHEN nbOfUsers > 10000
Str1: STRATEGY CASE fulfilled(Cons1) OR fulfilled(Cons2): minimize(cost)

#SYBL.ServiceUnitLevel
Str2: STRATEGY CASE ioCost < 3 Euro : maximize( dataFreshness )

#SYBL.CodeRegionLevel
Cons4: CONSTRAINT dataAccuracy>90% AND cost<4 Euro
Monitoring for Elastic Control

- MAPE-K control loop (Monitoring, Analysing, Planning, Executing using Knowledge)

Cloud Monitoring Challenges

- Monitor heterogeneous types of run-time information and resources
- Extract metrics from multiple levels of the cloud
  - Low-level metrics (i.e. CPU usage, network traffic)
  - High-level metrics (i.e. application throughput, latency, availability)
- Metrics collected at different time granularities
- Non-intrusiveness
Cloud Monitoring Challenges

- **Cloud Platform Independence**
  - If a cloud service is portable then it can be moved to another platform due to better *pricing schemes, availability, QoS*, etc.

- **Monitoring System?**
  - Portable
  - Easily configurable on new platform
Cloud Monitoring Challenges

- **Interoperability**
  - Distribute a cloud service across multiple providers due to better resource locality, availability or security concerns

- **Monitoring System?**
  - Operate and collect metrics seamlessly across multiple providers

42% are interested in adopting hybrid cloud. Estimated to rise to 55% by 2016 [GIGAOM 2014]
Cloud Monitoring Challenges

• Elasticity Support
  • Detect configuration changes in a cloud service

• Monitoring System?
  • Detect configuration changes automatically without restarting monitoring process or part of it and without any human intervention

Application topology changes (e.g. new VM added)

Allocated resource changes (e.g. new disk attached to VM)
Cloud Monitoring State-of-the-art

• Cloud-specific Monitoring Tools
  • Limited to a specific number of cloud platforms
  • Commercial and proprietary -> limited portability and interoperability

• General Purpose Monitoring Tools
  • Suitable for only slowly changing fixed infrastructures
  • Limited application-level monitoring support

• No elasticity support
  • Detect configuration changes in:
    • Application topology (e.g. new VM added)
    • Allocated resource (e.g. new disk attached to VM)
JCatascopia Monitoring System

• Open-source
• Interoperable
• Scalable

• Multi-Layer Cloud Monitoring
  • Customisable and Extensible by Users
  • Metric Subscription Rule Language

• Platform Independent
  • Operates on any cloud platform
  • Metric collection, distribution or storage are independent to underlying infrastructure

• Elastic by design

JCatascopia Architecture
Monitoring Probes

- The actual metric collectors managed by Monitoring Agents
- Collect system-level and application performance metrics
- Push or Pull mechanism to forward metrics to their corresponding Agent
JCatascopia Probes

- JCatascopia Probe API
- Application developers can implement their own Probes
- Dynamically deployable to Monitoring Agents
- Push and Pull metric delivery mechanism
- Filtering mechanism at Probe level
  - Minimizes communication and storage overhead
  - Probe developers can create their own metric filters
Monitoring Agents

- Light-weight monitoring instances
- Deployable on physical nodes or virtual instances
- Responsible for the metric collection process
- Aggregate and distribute collected metrics (pub/sub)
Monitoring Server

- Receives metrics from Monitoring Agents
- Aggregates, filters, processes and stores metrics in Database
- Handles user metric and configuration requests
- Hierarchy of Monitoring Servers for greater scalability
Dynamic Agent Discovery

Benefits
- Monitoring Servers are agnostic of Agent network location
- Agents appear dynamically

Eliminated the need to
- Restart or reconfigure Monitoring System
- Depend on underlying hypervisor
- Require directory service with Agent locations
Dynamic Agent Removal

• Heartbeat monitoring to detect when Agents:
  • Removed due to scaling down elasticity actions
  • Temporary unavailable (network connectivity issues)
Metric Subscription Rule Language

- **Aggregate single instance metrics**
  
  \[
  \text{SUM(errorCount)}
  \]

- **Generate high-level metrics at runtime**
  
  \[
  \text{DBthroughput} = \text{AVG(readps+writeps)}
  \]

---

**Subscription Rule Example**

Average DBthroughput from the low-level metrics readps and writeps of a database cluster comprised of \(N\) nodes:

\[
\text{DBthroughput} = \text{AVG(readps + writeps)}
\]

\[
\text{MEMBERS} = [\text{id1}, \ldots, \text{idN}]
\]

\[
\text{ACTION} = \text{NOTIFY(\langle25,>75\%)}
\]
Adaptive Filtering

- Simple fixed range filter windows are not effective:
  - i.e. filter $\text{currentValue}$ if in window $\pm R$
  - No guarantee that any values will be filtered at all

- Adaptive filter window range
  - Window range ($R$) is not static but depends on percentage of values previously filtered

Diagram:
- Collect Samples
- Check percentage of filtered values
- Adjust Window Range ($R$)
Evaluation

• Validate JCatascopia functionality and performance

• Compare JCatascopia to other Monitoring Tools
  • Ganglia
  • Lattice Monitoring Framework

• Testbed
  • Different domains of Cloud applications
  • Various VM flavors
  • 3 public Cloud providers and 1 private Cloud
## Testbed

<table>
<thead>
<tr>
<th>Cloud Provider</th>
<th>VM no.</th>
<th>VM Flavor</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRNET Okeanos public Cloud1</td>
<td>15</td>
<td>1GB RAM, 10GB Disk, Ubuntu Server 12.04 LTS</td>
<td>12 VMs Cassandra&lt;sup&gt;7&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 VMs YCSB Clients&lt;sup&gt;8&lt;/sup&gt;</td>
</tr>
<tr>
<td>Flexiant FlexiScale platform2</td>
<td>10</td>
<td>2 VCPU, 2GB RAM, 10GB Disk, Debian 6.07 (Squeeze)</td>
<td>HASCOP&lt;sup&gt;9&lt;/sup&gt;</td>
</tr>
<tr>
<td>Amazon EC23</td>
<td>10</td>
<td>m1.small with CentOS 6.4 (1VCPU, 1.7GB RAM, 160GB)</td>
<td>HASCOP</td>
</tr>
<tr>
<td>UCY Nephelae Private Cloud4</td>
<td>60</td>
<td>2 VCPU, 2GB RAM, 10GB Disk, Ubuntu Server 12.04 LTS</td>
<td>HASCOP</td>
</tr>
</tbody>
</table>

We have deployed on all VMs JCastascopia Monitoring Agents, Ganglia gmmonds<sup>5</sup> and Lattice<sup>6</sup> DataSources

1 https://okeanos.grnet.gr/     4 http://linc.ucy.ac.cy/Nephelae/
2 http://www.flexiscale.com/     5 Ganglia v3.1.7
3 http://aws.amazon.com/ec2/      6 Lattice v0.6.4
7 http://cassandra.apache.org/
8 [B.F. Cooper, 2010]
9 [A. Papadopoulos, 2013]
# Testbed - Available Probes

<table>
<thead>
<tr>
<th>Probe</th>
<th>Metrics</th>
<th>Period (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>cpuUserUsage, cpuNiceUsage, cpuSystemUsage, cpuldle, cpuIOWait</td>
<td>10</td>
</tr>
<tr>
<td>Memory</td>
<td>memTotal, memUsed, memFree, memCache, memSwapTotal, memSwapFree</td>
<td>15</td>
</tr>
<tr>
<td>Network</td>
<td>netPacketsIN, netPacketsOUT, netBytesIN, netBytesOUT</td>
<td>20</td>
</tr>
<tr>
<td>Disk Usage</td>
<td>diskTotal, diskFree, diskUsed</td>
<td>60</td>
</tr>
<tr>
<td>Disk IO</td>
<td>readkbps, writekbps, iotime</td>
<td>40</td>
</tr>
<tr>
<td>Cassandra</td>
<td>readLatency, writeLatency</td>
<td>20</td>
</tr>
<tr>
<td>YCSB</td>
<td>clientThroughput, clientLatency</td>
<td>10</td>
</tr>
<tr>
<td>HASCOP</td>
<td>clustersPerIter, iterElapTime, centroidUpdTime, pTableUpdTime, graphUpdTime</td>
<td>20</td>
</tr>
</tbody>
</table>
When in need of **application-level monitoring**, for a small runtime overhead, **JCatascopia** can **reduce** monitoring network traffic and consequently **monitoring cost**.
Multi-Tier Monitoring

JCatascopia Metric Rule Language and Mechanism

\[ \text{avgActiveConnections} = \text{AVG(busyThreads)} \]
\[ \text{MEMBERS} = [\text{id1}, \ldots, \text{idN}] \]
\[ \text{ACTION} = \text{NOTIFY(<70, } \geq 140) \]

\[ \text{avgCPUUsage} = \text{AVG(1-cpuIdle)} \]
\[ \text{MEMBERS} = [\text{id1}, \ldots, \text{idN}] \]
\[ \text{ACTION} = \text{NOTIFY(<30, } \geq 85) \]
JCatascopia: Portability and Interoperability

Online Gaming

Multi-Tier Video Streaming

Multi-Graph Clustering in the Cloud

XDB In-Memory Data Analytics

SCAN Genome Pipeline
JCatascopia: Scalability Evaluation

1 Monitoring Server
MySQL DB
JCatascopia: Scalability Evaluation

1 Monitoring Server
1 Cassandra Node
JCatascopia: Scalability Evaluation

1 Monitoring Server
2 Cassandra Nodes
JCatascopia: Scalability Evaluation

1 root Monitoring Server and 2 Intermediates
When archiving time is high, we can direct monitoring metric traffic through multiple Monitoring Servers, allowing the monitoring system to scale.
JCatascopia Release

- Open-source under **Apache 2.0 Licence**
  - **Github**: https://github.com/CELAR/cloud-ms

- JCatascopia Website (docs, examples, videos, publications, etc.)
  - http://linc.ucy.ac.cy/CELAR/jcatascopia

- Packaging (**JARs, tarballs, RPMs and Chef recipes**) available in CELAR repo

- JCatascopia **Probe Library** and Java Probe API
  - https://github.com/dtrihas/JCatascopia-Probe-Library
  - System-level monitoring probes (for both Linux and Windows)
  - Application-specific probes (Tomcat, Cassandra DB, HAProxy, Postgres DB, RabbitMQ)

- Supporting **2 Different Database Backends** (MySQL, Cassandra DB)
OUTLINE

• Cloud Computing
• CELAR main topics:
  • Elasticity
  • Application Management
• CELAR Architecture
• Elasticity and Monitoring - JCatascopia
• c-Eclipse and CAMF
• Conclusions
Cloud Application Management

- Goal: ease the description & deployment of applications over Cloud infrastructures.
- However:
  - Most frameworks are vendor-specific and fail to address one of the main challenges in cloud application management, that is the vendor neutrality [Gartner]
  - Proprietary
  - Provide limited support for elasticity
c-Eclipse

• Open-source (on top of Eclipse platform)
• Manages Complete Cloud Application Life Cycle
• Platform independent
• Intuitive graphical drag-and-drop UI
• Adopts open Cloud specifications
• Separates general-purpose functionality from IaaS platform-specific details (portable/extensible)
• Supports an open language for describing Cloud applications’ elasticity requirements
TOSCA

- TOSCA provides a language to describe:
  - Application components & relationships (*topology*)
  - Application management procedures (*orchestration*)

Orchestration can be inferred from Types’ semantics i.e. interfaces
TOSCA

- Specifies an XML Binding
  - Implemented in c-Eclipse

- Specifies **CSAR** (Cloud Service Archive) an exchange format to package Cloud applications

CSAR Format
c-Eclipse Architecture

- Graphical modeling of application’s topology
- Drag-and-drop interface
c-Eclipse Architecture

Associated with visual elements by the Modeling Tool
c-Eclipse Architecture

- c-Eclipse specific types for Nodes, Relationships etc.
c-Eclipse Architecture

- Packages application descriptions as CSAR archives
Application Modeling Tool
Graphical TOSCA Modeling
Cloud Project View

- c-Eclipse organizes files in a structured hierarchy
  - Just like any other Eclipse project
- Folders are placeholders for files required throughout application’s lifecycle i.e.
  - Content needed to realize a deployment (executables, configuration files, VM images etc.)
- Folders’ structure is automatically created on project’s creation
Palette

- **Connections**: Different relationship types can be specified i.e. “Depends On”, “Connects To”
- **Application Components**: Application component types + composite component
- **Images**: Provider’s images & user’s custom built images
- **Monitoring Probes**: Monitoring metrics available by the provider’s monitoring system or by the integrated to c-Eclipse monitoring system
- **Elasticity Actions**: Provider supported elasticity actions & user’s custom elasticity actions
- **User Applications**: User’s custom created applications
- **Key Pairs**: Generated by the user, used for accessing the deployed components
- **Deployment Scripts**: User’s custom configuration scripts
Palette elements can be dragged-and-dropped onto the canvas.

More details can be specified through the Properties View.
Cloud Provider Selection

- Users can select Cloud providers for deploying their applications
  - Authentication credentials must be provided
  - c-Eclipse uses vendor’s API to retrieve required info
Elasticity Policies Specification

- c-Eclipse facilitates the specification of applications’ elasticity policies
- Applications can scale at runtime based on user defined policies

![c-Eclipse Properties View: Elasticity Policies Specification](image-url)
DataPlay in c-Eclipse

Example application: Online Gaming
Cancer Genome Detection in c-Eclipse
Images
Flavors
Resizing Actions
Monitoring Probes and Metrics
Deployment over different Cloud Infrastructures

- Applications’ deployment request are sent
- Applications are up and running on 2 infrastructures
- The status of the two deployments is shown in the Application Deployments View of c-Eclipse
c-Eclipse
... now becoming ...

Cloud Application Management Framework (CAMF)
official Eclipse Project since October 2014

https://projects.eclipse.org/projects/technology.camf
http://linc.ucy.ac.cy/CAMF
CAMF Extended Architecture
Summary and Conclusions

• Elasticity works! Videos with demos are available online (Youtube channel: CELAR Cloud)

• Elastic Monitoring can be performed in a platform-independent and scalable way, non-intrusively.

• CAMF represents a powerful vendor-neutral environment for the Cloud application lifecycle.

• In progress: Information System and Cloud Analytics.

• Open-source and available on github
Acknowledgments

- G. Pallis, Assistant Professor
- D. Antoniades, Postdoctoral Fellow
- N. Loullouides, Ph.D. Candidate
- S. Sofokleous, Senior Research Engineer
- D. Trihinas, Ph.D. Candidate
- A. Foudoullis, Research Engineer
- A. Balla, Senior IT Engineer
- A. Kastanas, B.Sc. Student
THANK YOU!

DOWNLOAD celar NOW

https://github.com/celar

Video demos online on YouTube channels CELAR EU, LINC-UCY

Marios Dikaiakos
mdd@ucy.ac.cy
@dikaiakos