Managing and Monitoring Elastic Cloud Applications

Demetris Trihinas, Chrystalla Sofokleous, Nicholas Loulloudes, Athanasios Foudoulis, George Palilis, Marios D. Dikaiakos
Department of Computer Science, University of Cyprus
{trihinas, stalosof, loulloudes.n, afoudou01, gpallis, mdd}@cs.ucy.ac.cy
http://linc.ucy.ac.cy/CELAR/icwe2014/

Introduction

- **Elastic Cloud applications** have the ability to expand or contract their comprised resources to meet runtime demands
- **Managing and monitoring** Cloud applications’ lifecycle is a challenging endeavor
- Current Application Management Frameworks and Monitoring Tools lack in portability and interoperability

Contribution

We present two open-source tools for managing and monitoring elastic Cloud applications:

- **c-Eclipse**: Facilitates portable descriptions of elastic Cloud applications by adopting an open Cloud standard (TOSCA), and a language for elasticity requirements specification
- **JCatascopia**: Fully automated, platform independent, interoperable Cloud monitoring system, that considers application topology changes due to elasticity actions

Application Description

**c-Eclipse Application Management Framework**

- Portable application descriptions via a GUI
- On the fly translation of graphical descriptions into TOSCA
- Specification of elasticity capabilities/requirements

**Application Description Contents**

- Application Topology (components & relationships)
- Deployment Files (deployment/configuration scripts, executable files, VM images, key pairs)
- Deployment Preferences (VM flavors, number of components’ instances)

**Elasticity Strategies at different application levels**

If (CPU Utilization (NoSQL BH) > 80%) then AddVM

**Application Deployment**

- c-Eclipse sends deployment requests to the Cloud Manager
- Requests contain all necessary contextualization artifacts
- Deployments Tab shows details for each node status per IaaS

**Use - Case**

3-Tier Online Video Streaming Service

- **Load Balance Tier**: HAProxy
- **Application Server Tier**: Tomcat with Video Streaming Web Service
- **Database Tier**: Cassandra NoSQL

**Application (& Cloud) Monitoring**

**JCatascopia Monitoring System**

- Deployable in a fully automatic manner
- Deployable on any underlying platform
- Dynamically detects application topology changes due to elasticity actions
- Provides filtering capabilities
- Dynamically generates high-level app metrics

**JCatascopia**

**Pub/Sub Message Pattern**

- Monitoring Servers bind to network interface, awaiting for incoming requests
- Monitoring Agents initiate subscription by pinging the Monitoring Server and publish collected metrics

**Our Approach: Elasticity Management Platform**

1. **Graphically describe** application structure & management operations
2. **Specify elasticity strategies**
3. **Deploy application** over selected Cloud providers
4. **Monitor** Cloud platform resources and application’s performance
5. **Scale** deployed application based on elasticity strategies

**Application Server: High # of client connections → Add App. Server**

**Scaling Strategies**

- **Cassandra Ring**: High CPU utilization → Add Cassandra Node

**Initial Deployment**

1 Load Balancer
1 App Server
1 Cassandra DB Node

**After 1 hour**

1 Load Balancer
2 App Servers
6 Cassandra DB Nodes


This work was partially supported by the European Commission in terms of the CELAR 317790 FP7 project (FP7-ICT-2011-8).