



Project Acronym: **OPTIMIS**  
Project Title: **Optimized Infrastructure Services**  
Project Number: **257115**  
Instrument: **Integrated Project**  
Thematic Priority: **ICT-2009.1.2 – Internet of Services, Software and Virtualisation**

## Cloud Optimizer User Guide

*Activity 4: Basic Service Operation*

*WP 4.2: Cloud Runtime Optimization*

<b>Due Date:</b>	M36
<b>Submission Date:</b>	30/05/2013
<b>Start Date of Project:</b>	01/06/2010
<b>Duration of Project:</b>	36 months
<b>Organisation Responsible for the Deliverable:</b>	Barcelona Supercomputing Center
<b>Version:</b>	1.0
<b>Status</b>	Final
<b>Author(s):</b>	J. Oriol Fitó BSC
<b>Reviewer(s)</b>	Sotiris Stamokostas (NTUA) Johan Tordsson (UMU)

---

Project co-funded by the European Commission within the Seventh Framework Programme		
<b>Dissemination Level</b>		
<b>PU</b>	Public	<b>X</b>
<b>PP</b>	Restricted to other programme participants (including the Commission)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission)	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission)	



## Version History

<b>Version</b>	<b>Date</b>	<b>Comments, Changes, Status</b>	<b>Authors, contributors, reviewers</b>
0.1	2013-03-31	First version	Mario Macías (BSC)
0.2	2013-04-10	Final version ready for submission	J. Oriol Fitó (BSC)
1	2013-30-05	Final version	Malena Donato (ATOS)



## Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>6</b>
1.1	GLOSSARY OF ACRONYMS.....	6
<b>2</b>	<b>CLOUD OPTIMIZER USER GUIDE .....</b>	<b>7</b>
2.1	RELEASE INFORMATION .....	7
2.2	INTRODUCTION.....	7
2.3	FUNCTIONALITIES.....	7
2.3.1	<i>Parsing IP manifest using the new API .....</i>	<i>7</i>
2.3.2	<i>TREC-based decision between local and remote deployment .....</i>	<i>7</i>
2.3.3	<i>Improved management of service resources at SP-level .....</i>	<i>8</i>
2.3.4	<i>Improved management of infrastructure status .....</i>	<i>8</i>
2.3.5	<i>Proactive behavior of TREC tools .....</i>	<i>9</i>
2.3.6	<i>Coordination of Holistic Management process in collaboration with TREC assessors and Low-Level Managers.....</i>	<i>10</i>
2.4	KNOWN LIMITATIONS .....	10
2.5	GETTING STARTED.....	10
2.5.1	<i>Using the Software .....</i>	<i>10</i>
2.5.2	<i>Testing the Software.....</i>	<i>11</i>
2.5.3	<i>Configuration .....</i>	<i>11</i>
2.6	FAQ .....	12
2.7	OTHER INFORMATION .....	12
2.7.1	<i>Directory Structure .....</i>	<i>12</i>
2.7.2	<i>Contributors.....</i>	<i>12</i>



## Index of Tables

Table 1 - Acronyms table.....	6
Table 2 - Release information .....	7
Table 3 - Methods offered to other components to let them query the database .....	9



## 1 Introduction

This document includes the user guide for the software component Cloud Optimizer.

Section 2 details the component's release information, its functionality and provided API. It also details its limitations, as well as its code information and directory structure. Its usage and testing procedures are also provided.

### 1.1 Glossary of Acronyms

Acronym	Definition
API	Application Programming Interface
BLO	Business Level Objective
IP	Infrastructure Provider
JAR	Java ARchive
REST	Representational State Transfer
SP	Service Provider
TREC	Trust, Risk, Eco-efficiency, Cost
VM	Virtual Machine

Table 1 - Acronyms table

## 2 Cloud Optimizer User Guide

### 2.1 Release information

Component Name	Release Number	Release Date
Cloud Optimizer	1.0-SNAPSHOT	2012-04-30

Table 2 - Release information

### 2.2 Introduction

The Cloud Optimizer is part of the OPTIMIS toolkit for Infrastructure Providers (IPs).

It combines the monitoring and assessment tools in the OPTIMIS Base Toolkit with various management engines in order to create a self-managed Cloud infrastructure driven by provider's business-level objectives (i.e. BLOs).

In particular to the implication of this component during services lifecycle, it exposes the IP external interfaces needed to deploy and undeploy services (i.e. a set of VMs). These operations can be requested by Service Providers, as well as by IPs in the Cloud bursting scenario. In addition, it offers the possibility to add, remove and restart services' VMs during their operation. It should be noted that these operations can be requested by Elasticity Engine and Fault Tolerance Engine.

Besides, it is responsible of maintaining a database containing all the information regarding physical and virtual resources of a Cloud infrastructure.

### 2.3 Functionalities

This release implements these new features and functionality:

- Parses IP manifests using the new API
- Takes decision of local or remote (bursting) deployment based on TREC and infrastructure status
- Improved management of service resources at SP-level
- Improved management of the internal database containing the state of the provider infrastructure
- Methods towards proactive behaviour of TREC tools
- Coordinates Holistic Management process in collaboration with TREC assessors and Low-Level Managers

#### 2.3.1 Parsing IP manifest using the new API

This component is able to parse IP manifests using the new API provided by the Service

#### 2.3.2 TREC-based decision between local and remote deployment

The Cloud Optimizer takes into account TREC assessments when deciding between local or remote deployment of VMs.



### 2.3.3 Improved management of service resources at SP-level

The management of service resources done by this component has been improved to match with their new structure.

### 2.3.4 Improved management of infrastructure status

The management of infrastructure status has been improved by adding some more fields into the internal database.

The following table details all the methods offered to other components to let them query the database:

Operation	Input	Output	Description
addPhysicalResource	Parameter 1: physical resource		It adds a physical resource into IP's management tables
getPhysicalResource	Parameter 1: node identifier	Physical resource	It returns a given physical resource
deletePhysicalResource	Parameter 1: node identifier		It deletes a physical resource from the IP's management tables
getNodesId	-	List of nodes identifier	Returns the list of IP's nodes identifier
addVirtualResource	Parameter 1: virtual resource		
updateVirtualResource	Parameter 1: vm identifier Parameter 2: node identifier		Updates the location (physical node) of a virtual resource
getVirtualResource		Virtual resource	Returns a virtual resource
deleteVirtualResource	Parameter 1: vm identifier		Deletes a given virtual resource from the IP's management tables
getVMsId	Parameter 1: physical node identifier	List of identifiers of VMs running in the given physical node	Returns the list of identifiers of VMs running in the given physical node
getNodeId	Parameter 1: VM identifier	Physical node identifier	Return the identifier of the physical node where a given VM is running





getVMName	Parameter 1: VM identifier	VM name	Returns the name of a given VM
getVMId	Parameter 1: VM name	VM identifier	Returns the identifier of a given VM
getVMServiceld	Parameter 1: VM identifier	Service identifier	Returns the identifier of the service to which the VM belongs to
getVMsIdsOfService	Parameter 1: service identifier	List of VMs identifier	Returns a list of identifiers of VMs composing a given service
getIpld	Parameter 1: physical node identifier	The identifier of the IP where the node is located	It is used to obtain the identifier of the IP where the node is located
getIpVmIpAddress	Parameter 1: IP identifier	The IP address of an external Infrastructure Provider (IP)	This method is used to know the IP address of any external Infrastructure Provider (IP)
setBLO	Parameter 1: Business-Level Objective		This method is used by the IP manager for specifying the objective functions as well as the TREC constraints

Table 3 - Methods offered to other components to let them query the database

### 2.3.5 Proactive behavior of TREC tools

This release includes the next methods which are offered to TREC tools in order to let them to pull TREC assessments when thresholds are reached:

- /hm/risk/vm/{vmId}
- /hm/eco/vm/{vmId}
- /hm/cost/vm/{vmId}
- /hm/trust/vm/{vmId}
  - Notifies when VM Trust/Risk/Eco/Cost goes beyond the given threshold (as specified by see method setBLO)
  - Parameters
    - vmId: identifier of the virtual machine
    - Actual value of the TREC
- /hm/trust/service/{serviceld}
- /hm/risk/service/{serviceld}

- /hm/eco/service/{serviceId}
- /hm/cost/service/{serviceId}
  - Notifies when Service Trust/Risk/Eco/Cost goes beyond the given threshold (as specified by see method setBLO)
  - Parameters
    - serviceId: identifier of the service
    - Actual value of the TREC
- /hm/trust/node/{nodeId}
- /hm/risk/node/{nodeId}
- /hm/eco/node/{nodeId}
- /hm/cost/node/{nodeId}
  - Notifies when Node Trust/Risk/Eco/Cost goes beyond the given threshold (as specified by see method setBLO)
  - Parameters
    - nodeId: identifier of the node
    - Actual value of the TREC
- /hm/trust/ip
- /hm/risk/ip
- /hm/eco/ip
- /hm/cost/ip
  - Notifies when the Trust/Risk/Eco/Cost of the whole IP goes beyond the given threshold (as specified by see method setBLO)
  - Parameters
    - Actual value of the TREC

### 2.3.6 Coordination of Holistic Management process in collaboration with TREC assessors and Low-Level Managers

This release contains the final version of the Holistic Management process, which optimizes the configuration of the LLMs in function of the IP manager policies and the real time TREC information.

## 2.4 Known limitations

N/A

## 2.5 Getting Started

### 2.5.1 Using the Software

There is a client to interact with this component, which is packaged as a JAR file and can be used by any user. The only thing that one needs to do is importing such file, which can be found into the project's repository: <http://optimis->

artifactory.atosorigin.es/artifactory/repo/eu/optimis/CloudOptimizerRESTClient/1.0-SNAPSHOT/CloudOptimizerRESTClient-1.0-SNAPSHOT.jar

### 2.5.2 Testing the Software

In a Maven environment, the test cases provided can be invoked by means of the following command: `mvn test`.

### 2.5.3 Configuration

Cloud Optimizer must be configured by editing the configuration file at `$(OPTIMIS_HOME)/etc/CloudOptimizer/config.properties`. Next are listed the properties that can be manually tuned for configuring CO:

- `config.ipvm_host`, `config.ipvm_port`
  - Specifies the hostname and port of the IP host that contains other components. Usually is 'localhost:8080'
- `config.svvm_host`, `config.svvm_port`
  - Specifies the hostname and the port of the Service Provider host.
- `config.svvm_host_sd`, `config.svvm_port_sd`
  - Specifies the hostname and the port of the Service Deployer host used for federation cases
- `manifest.location`
  - Location of the Manifest Template file used for Bursting Scenario
- `db.driver`
  - JDBC driver for database access. Usually is 'com.mysql.jdbc.Driver'
- `db.username`, `db.password`
  - Username and password for database access.
- `db.location=XXXX`
  - Identifier of the location of the database. It should be a descriptive name of the database (e.g. testing, development, etc...)
- `XXXX.url`
  - URL of the database of the CO. The XXXX is the value of the `db.location` property.
- `XXXX.url_trec`
  - URL of the database of the TREC assessors. The XXXX is the value of the `db.location` property.

## 2.6 FAQ

N/A

## 2.7 Other information

### 2.7.1 Directory Structure

This software component is divided in three main directories:

- *DataFormats*: specifies Java XML Bindings for different message formats of the Cloud Optimizer interfaces.
- *BusinessDescriptor*: specifies Java XML Bindings and helper methods for parsing and validating Business-Level Objectives.
- *CloudOptimizerRESTClient*: this software part is intended to be used by clients of this component. It is composed by several Java interfaces that act as clients of REST methods offered by a server.
- *CloudOptimizerServer*: this are the server's core functionality and RESTful interfaces which provide access to all the functionalities of this component.

### 2.7.2 Contributors

J. Oriol Fitó (BSC)

Jordi Guitart (BSC)

Mario Macías (BSC)