SAPIENZA
UNIVERSITÀ DI ROMA

Università degli studi di Roma La Sapienza

Facoltà di Ingegneria dell'Informazione,
Informatica e Statistica

Tesi di Laurea Magistrale in
Ingegneria informatica

MANAGING QOS IN TRANSACTIONAL APPLICATION

Relatore
Prof. Bruno Ciciani
Correlatore
Ing. Roberto Palmieri

Candidato
Fabio Mariotti

Anno Accademico 2011/2012

A mia mamma e mio papà...

Contents

1 Quality of Service and Service Level Agreement 10

1.1 Introduction 10

1.2 QoS in transactional applications 11

1.2.1 Performance parameters in QoS 11

1.2.2 Application Contention Factor 12

1.3 Transactional Memories 13

1.4 Service Level Agreement 13

1.4.1 SLA Components 14

1.4.2 SLA Lifecycle 15

1.4.3 WS Agreement 17

1.4.3.1 Conceptual Model 17

1.4.3.2 Agreement model 18

2 Cloud Computing 24

2.1 Cloud Computing Model 25

2.2 Cloud Computing Structure 27

2.3 Types of Cloud Computing 29

2.3.1 Public Cloud 29

2.3.2 Private Cloud 29

2.3.3 Hybrid Cloud 29

2.4 Benefits of Cloud Computing 30

2.4.1 Elastic resources in Cloud Computing 33

2.4.2 NoSQL Key/Value store system 33

2.4.3 Horizontal and vertical scaling 34

3 QoS and SLA in Cloud Computing **35**
3.1 Motivation 35
3.2 State of the art 36
3.3 Cloud Elasticity and Databases 36
3.4 Cloud-TM project 37

4 Contribution **39**
4.1 Introduction 39
4.2 Infinispan platform 39
4.2.1 Java Map and ConcurrentMap extension . . 40
4.2.2 Transactional class distinction: user and provider advantages 41
4.2.3 Infinispan Interceptors 42
4.2.4 Implementing class distinction in transactions 44
4.3 Exposing statistics through JMX 46
4.4 Developing a QoS framework for the cloud 48
4.5 SLA negotiation application 49
4.5.1 QoS in data-centric applications 49
4.5.2 SLA Template: SLA@SOI model 50
4.5.3 SLA negotiation engine 58
4.5.4 SLA Application protocol 62
4.5.4.1 Application protocol in detail . . . 63
4.5.4.2 Login and registration 63
4.5.4.3 Submit SLA request 63
4.5.4.4 User application upload 64
4.5.4.5 User prediction panel 64
4.5.4.6 Administrator application panel . . 67
4.5.4.7 SLA Application main component 67
4.5.5 SLA Application charts library 68
4.5.5.1 Connection to the server 69

5 Evaluation with Infinispan benchmark tool **70**
5.1 Radargun application benchmark 70
5.2 TPC-C benchmark 71

5.2.1 Benchmark model 71
5.2.2 Transactional classes in TPC-C benchmark . 73
5.3 FutureGrid platform 73
5.4 Running benchmark on FutureGrid 74

6 Conclusions **76**

A CloudTM SLA Web Application installation manual **78**

Bibliography **80**

Abstract

Before the advent of Cloud Computing platform, in order to ensure that an application SLA was not violated, a resource overprovision policy was often adopted. This meant that all possible resources that an application could require in the worst case (e.g. in the peak time of the daily usage) were allocated statically to that specific application. Using this policy is very easy lead to a largely suboptimal utilization of the resources. In fact being statically allocated, during off peak hours, a number of resources can remain unused at run time, causing a waste of money. If we think now to a Cloud Computing architecture where we have the possibility to dynamically configure our infrastructure in real time acquiring the resources with a pay-as-use model, we see a whole new scenario in front of us. To honor the QoS chosen by a user through a SLA, an application can acquire at run time the cloud resources, and release them when no longer needed, minimizing costs guaranteeing the QoS requested. In view of these considerations is born the Cloud-TM project, an in-memory transactional data platform that leverages the elasticity of cloud resources in order to dinamically varying the scale of the platform in real-time to meet demands of varying workloads. To manage the negotiation of the QoS in transactional application I contributed to the design and develpment of a framework that shall be used by Cloud-TM platform to manage the QoS API layer of the platform and to the introduction of a fine grain control over the QoS of transactions.