Multi-time Scale Distributed Capacity Allocation and Load Redirect Algorithms for Cloud Systems

Advisor: Dr. Danilo ARDAGNA
Co-Advisor: Dr. Barbara PANICUCCI

Thesis by:
Marco CASIERO - 734383
Stefano VETTOR - 734450

Academic Year 2009/2010
Contents

1 Introduction .......................... 5

2 State of the art ......................... 7
   2.1 Cloud Computing ..................... 7
   2.1.1 Infrastructure-as-a-Service (IaaS) .... 9
   2.1.2 Platform-as-a-Service (Paas) ........... 13
   2.1.3 Software-as-a-Service (SaaS) .......... 16
   2.2 Centralized and Distributed Solutions for Resource Management 16
   2.3 Distributed Techniques ................. 17
   2.4 Distributed Techniques for Requests Redirection Mechanisms . 19
      2.4.1 Request Routing Mechanisms for Distributed Web Sys-
             tems .................................. 20
      2.4.2 Web Server Routing Mechanisms ........ 21

3 Load Balancing Algorithm ............ 25
   3.1 Problem Statement ...................... 25
   3.2 Reference Framework and Design Assumptions ............... 29
   3.3 Workload Prediction Models .............. 31
   3.4 Optimization Problem Formulation ........... 33
      3.4.1 Capacity Allocation problem .......... 33
      3.4.2 Load Redirect problem for Scenario 1 .... 34
      3.4.3 On Solving (SUB_i) problem ........... 39
      3.4.4 Load redirect for Scenario 2 .......... 42
   3.5 Capacity Allocation - Determining Classes Capacity ........ 45

4 Tools ................................ 49
   4.1 AMPL .................................. 49
   4.2 SPECweb2005 .......................... 50
      4.2.1 SPECweb2005 - Banking ............... 54
   4.3 Amazon Tools .......................... 56
   4.4 Amazon’s Scripts ....................... 57
List of Figures

3.1 Cloud System - Scenario 1 ................................. 26
3.2 Large Scale Service Center - Scenario 2 .................... 27
3.3 Cloud System Reference Framework ....................... 28

4.1 SPECweb2005 Test Time Intervals .......................... 53
4.2 Test’s Sequence Diagram .................................. 58

5.1 Our solution, Oracle, Heuristic 1 and 2 - Instantaneous Virtual Machines Cost for low traffic conditions .................. 73
5.2 Heuristic 3 - Instantaneous Virtual Machines Cost for low traffic conditions ................................................... 74
5.3 Our solution, Oracle, Heuristic 1 and 2 - Instantaneous Virtual Machines Cost for high traffic conditions ................ 74
5.4 Heuristic 3 - Instantaneous Virtual Machines Cost for high traffic conditions ................................................... 75
5.5 Our solution, Oracle, Heuristic 1 and 2 - Instantaneous Virtual Machines Cost for high noisy traffic conditions .......... 75
5.6 Heuristic 3 - Instantaneous Virtual Machines Cost for high noisy traffic conditions ............................................... 76
5.7 Our solution, Oracle, Heuristic 1 and 2 - Response Time of site 5 for low traffic conditions ................................. 82
5.8 Our solution, Oracle, Heuristic 1 and 2 - Response Time of site 8 for low traffic conditions ................................. 82
5.9 Heuristic 3 - Response Time of site 5 for low traffic conditions 83
5.10 Heuristic 3 - Response Time of site 8 for low traffic conditions 83
5.11 Our solution, Oracle, Heuristic 1 and 2 - Response Time of site 5 for high traffic conditions ............................. 84
5.12 Our solution, Oracle, Heuristic 1 and 2 - Response Time of site 8 for high traffic conditions ............................. 84
5.13 Heuristic 3 - Response Time of site 5 for high traffic conditions 85
5.14 Heuristic 3 - Response Time of site 8 for high traffic conditions 85
5.15 Our solution, Oracle, Heuristic 1 and 2 - Response Time of site 5 for high noisy traffic conditions ........................................... 86
5.16 Our solution, Oracle, Heuristic 1 and 2 - Response Time of site 8 for high noisy traffic conditions ........................................... 86
5.17 Heuristic 3 - Response Time of site 5 for high noisy traffic conditions .................................................................................. 87
5.18 Heuristic 3 - Response Time of site 8 for high noisy traffic conditions .................................................................................. 87
5.19 Sites Model ......................................................................................... 88
5.20 Parameter Estimation ........................................................................ 90
5.21 First Test - Site 1 (Virginia) - $\Lambda_1$ ........................................ 92
5.22 First Test - Site 2 (North California) - $\Lambda_2$ ............................. 92
5.23 First Test - Site 1 (Virginia) - Response time .............................. 93
5.24 First Test - Site 2 (North California) - Response time ............... 93
5.25 First Test - Site 1 (Virginia) - Requests ....................................... 94
5.26 First Test - Site 2 (North California) - Requests ....................... 94
5.27 First Test - Response Time ............................................................. 95
5.28 Second Test - Site 1 (Virginia) - $\Lambda_1$ ....................................... 98
5.29 Second Test - Site 2 (North California) - $\Lambda_2$ ......................... 98
5.30 Second Test - Site 3 (Ireland) - $\Lambda_3$ ......................................... 99
5.31 Second Test - Site 1 (Virginia) - Response time ....................... 99
5.32 Second Test - Site 2 (North California) - Response time .......... 100
5.33 Second Test - Site 3 (Ireland) - Response time ......................... 100
5.34 Second Test - Site 1 (Virginia) - Requests .................................. 101
5.35 Second Test - Site 2 (North California) - Requests .................. 101
5.36 Second Test - Site 3 (Ireland) - Requests .................................... 102
5.37 Second Test - Response Time ....................................................... 102
# List of Tables

2.1 Amazon EC2 Instances Types ............................................. 10

3.1 Capacity Allocation and Load Redirect Problems Parameters and Decision Variables. ........................................... 29

4.1 Region Parameter ............................................................. 59

4.2 Files uploaded to instances ................................................... 60

5.1 Capacity Allocation Problem Solution Execution Time (sec) .... 68

5.2 Algorithm 1 Performance. ..................................................... 68

5.3 Capacity Allocation (various VM configurations case) Problem Solution Execution Time (sec) ........................................ 69

5.4 Low Traffic case costs ($) .................................................... 76

5.5 Low Traffic case costs - Heuristic 3 ($) ................................ 77

5.6 High Traffic case costs ($) .................................................. 77

5.7 High Traffic case costs - Heuristic 3 ($) ................................ 77

5.8 High Noisy Traffic case costs ($) ......................................... 78

5.9 High Noisy Traffic case costs - Heuristic 3 ($) ..................... 78

5.10 Cost differences percentage with respect to capacity allocation and load redirect solution in low traffic conditions .......... 78

5.11 Cost differences percentage with different thresholds with respect to capacity allocation and load redirect solution in low traffic conditions ......................................................... 79

5.12 Cost differences percentage with respect to capacity allocation and load redirect solution in high traffic conditions .......... 79

5.13 Cost differences percentage with different thresholds with respect to capacity allocation and load redirect solution in high traffic conditions ......................................................... 79

5.14 Cost differences percentage with respect to capacity allocation and load redirect solution in high noisy traffic conditions ... 80
5.15 Cost differences percentage with different thresholds with respect to capacity allocation and load redirect solution in high noisy traffic conditions .................. 80
5.16 Parameter Estimation Test Data .......................... 89
5.17 Parameter Estimation Delay and Service Time ............... 89
5.18 Delays between Sites in [s] .............................. 90
5.19 Second Test - VMs Number per Site/Hour .................. 96
5.20 Second Test Redirection ................................. 97
Abstract

In recent years the evolution and the widespread adoption of virtualization, service-oriented architectures, autonomic and utility computing have converged letting a new paradigm to emerge: The Cloud Computing. Cloud Computing aims at streamlining the on-demand provisioning of software, hardware, and data as services, providing end-user with flexible and scalable services accessible through the Internet.

Due to the large scale nature of the Cloud and the service centers, resource provisioning is one of the most important challenges. Indeed modern cloud infrastructures and service centers are characterized by continuous changes in the environment and in the requirements they have to meet. Therefore, in order to provide infrastructure or software as a service, advanced solutions have to be developed to be able to dynamically adapt the Cloud infrastructure, while providing continuous service and performance guarantees.

This thesis aims to develop capacity allocation techniques able to coordinate multiple distributed resource controllers working in geographically distributed cloud sites. Furthermore, capacity allocation solutions are integrated with a load redirection mechanism which forwards incoming requests among different domains in order to allow a better Quality of Service (QoS) during traffic fluctuations. The overall goal is to minimize the cost of the allocated cloud resources while guaranteeing quality of service constraints. In our work, the capacity allocation and load redirect of multiple class of requests are modeled as non-linear programming problem and solved with decomposition techniques. We performed also evaluations of our solution with multiple heuristics provided in the literature and the effectiveness has been evaluated on a real prototype environment deployed on Amazon EC2.

Results have shown that our solution is always cheaper than other state of the art techniques (up to 35%), especially under noisy workloads, without introducing significant QoS violations. Furthermore, our solutions are very close to the ones found by an oracle with perfect knowledge of the future.
Estratto

Negli ultimi anni l’evoluzione e la diffusa adozione di virtualizzazione, di architetture orientate ai servizi, autonomic and utility computing sono confluiti in un nuovo paradigma emergente: il Cloud Computing. Il Cloud Computing mira a semplificare la fornitura on-demand di software, hardware e dati erogati come servizi, proponendo così all’utente finale servizi flessibili scalabili accessibili tramite internet. Oggi l’offerta Cloud sta diventando sempre più ampia in quanto tutte le principali aziende IT ed i fornitori di servizi, come Microsoft, HP, Google, Amazon, Terremark e VMWare hanno iniziato a fornire soluzioni che sfruttano questo nuovo paradigma tecnologico. Negli ultimi anni si è vista una diffusione a livello mondiale di conglomerati di server chiamati Large Scale Service Center; come nello scenario Cloud anche in questo la gestione delle risorse è un problema critico.

A causa delle dimensioni su larga scala del Cloud e dei Service Center alcune delle maggiori sfide è la fornitura delle risorse. Infatti le infrastrutture delle Cloud moderne e dei Service Center operano in un mondo caratterizzato da cambiamenti continuini nell’ambiente e nei requisiti da soddisfare. Continui cambiamenti avvengono in modo autonomo e imprevedibile ed inoltre sono al di fuori del controllo del fornitore dei servizi Cloud. Pertanto, al fine di fornire infrastrutture o software come servizio, soluzioni avanzate devono essere sviluppate in grado di adattarsi dinamicamente alle infrastrutture Cloud, fornendo un servizio continuo e garantendo le performance. Dal momento che la violazione della qualità del servizio definita nel Service Level Agreement può portare ad una perdita di profitti i fornitori di servizi investono numerose risorse nella ricerca di soluzioni che minimizzino i costi rispettando nel contempo la qualità del servizio.

Questa tesi si propone di sviluppare tecniche di allocazione delle risorse in grado di coordinare variatori controllori di risorse distribuiti operanti in siti Cloud distribuiti geograficamente. Inoltre, le soluzioni di assegnazione delle risorse sono integrate con un meccanismo di reindirizzamento di carico che inoltra le richieste in arrivo tra domini diversi, al fine di consentire una migliore qualità del servizio durante le fluttuazioni del traffico. L’obiettivo
è quello di minimizzare il costo totale delle risorse assegnate garantendo comunque il rispetto dei vincoli sulla qualità del servizio. Nel nostro lavoro, l’assegnazione delle risorse ed il reindirizzamento di svariate classi di richieste vengono modellate come problemi di programmazione non lineare e risolti attraverso tecniche di decomposizione. Abbiamo effettuato inoltre un’ampia valutazione della nostra soluzione confrontandola con diverse euristiche presenti in letteratura. Infine abbiamo valutato l’efficacia dei nostri algoritmi di gestione delle risorse su un vero ambiente di prova il cui prototipo è stato implementato su Amazon EC2.

I risultati hanno dimostrato che la nostra soluzione è sempre più conveniente rispetto alle altre di riferimento (fino ad un 35%), in particolare modo in condizioni di traffico rumoroso senza comportare violazioni significative della qualità del servizio. Inoltre le nostre soluzioni risultano essere molto simili a quelle trovate da un oracolo con perfetta conoscenza del futuro.
Chapter 1

Introduction

In recent years the evolution and the widespread adoption of virtualization, service-oriented architectures, autonomic and utility computing have converged letting a new paradigm to emerge: The Cloud Computing. Cloud Computing aims at streamlining the on-demand provisioning of software, hardware, and data as services, providing end-user with flexible and scalable services accessible through the Internet. Nowadays, the Cloud offer is becoming day by day wider since all the major IT Companies and Service providers, like Microsoft, HP, Google, Amazon, Terremark and VMWare have started providing solutions involving this new technological paradigm.

Resource provisioning is one of the most important challenges for Clouds. Indeed modern Cloud infrastructures live in an open world, characterized by continuous changes in the environment and in the requirements they have to meet. Continuous changes occur autonomously and unpredictably, and they are out of control of the Cloud provider. Therefore, in order to provide infrastructure or software as a service, advanced solutions have to be developed to be able to dynamically adapt the cloud infrastructure, while providing continuous service and performance guarantees.

This thesis aims to develop capacity allocation techniques able to coordinate multiple distributed resource controllers working in geographically distributed Cloud sites. Furthermore, capacity allocation solutions are integrated with a load redirection mechanism which forwards incoming requests among different domains. The overall goal is to minimize the cost of the allocated Cloud resources, while guaranteeing quality of service constraints. In our work, the capacity allocation and load redirect of multiple class of requests are modeled as non-linear programming problems and solved with decomposition techniques. We performed also an extensive evaluation of our solution with multiple heuristics provided in the literature. Finally, the effectiveness of our resource management algorithms has been evaluated on a
real prototype environment deployed on Amazon EC2.

The thesis is organized as follows:

• In Chapter 2, we will introduce the Cloud Computing and its three paradigms: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS); for each of them some of the offerings available on the market will be presented. Then we will review the current state of the art of Centralized and Distributed Solutions for Resource Management and Distributed Techniques for Requests Redirection Mechanisms.

• In Chapter 3 both the Capacity Allocation an Load Redirection problems will be faced. We will start by stating the problem, then we will introduce the various assumptions and present the reference framework and the workload prediction model used in our work.

• In Chapter 4 we will describe all the tools used in our thesis work: At first we will present AMPL, the modeling language used to define the capacity allocation and load redirect optimization problems (in order to be able to solve them using the SNOPT solver) and to implement the heuristics used in our solution. Then we will introduce SPECweb2005, the benchmark software used to test our solution in a real cloud scenario. We will then proceed presenting the tools provided by Amazon to monitor and control the virtual machines running on its cloud; finally the shell scripts we developed to launch and manage the virtual machines on Amazon EC2 and to control SPECweb2005 tests will be described.

• Chapter 5 is dedicated to assess the quality of our solution through simulations and experiments. We will start presenting the results of the scalability analysis (in terms of number of request’s classes and sites/clusters) of both the capacity allocation and load redirect models presented in Chapter 3. Then we compare the simulation’s results of our solution (both in terms of costs and response time) with the results of the current state of the art techniques. In the last part of the Chapter we present the performance results of our solution in a real cloud scenario based on Amazon EC2 realized through SPECweb2005.

• In Chapter 6 are presented the work’s conclusion, underling the achieved results and presenting future research directions.
Chapter 6

Conclusions

In this thesis we proposed capacity allocation techniques able to coordinate multiple distributed resource controllers working in geographically distributed Cloud sites and large scale service centers. Since the Cloud paradigm is getting day by day more popular and that large scale service center are spawning all around the world the optimization of costs and resources is a central topic from both customer’s and provider’s perspective. Indeed, in any time instant resources have to be allocated to handle effectively workload fluctuations, while providing QoS guarantees to the end users. The overall goal we addressed in our thesis is the minimization of the costs associated with the allocated virtual machine instances, while guaranteeing QoS constraints expressed as a threshold on the average response time.

In our work we proposed a formulation of an hourly basis capacity allocation problem suitable for both a distributed cloud system and a large scale service center. Furthermore, we integrated our capacity allocation technique with a load redirect mechanism able to manage workload fluctuations at finer grained time scales (5-10 minutes); like the capacity allocation solution also this one is suitable, in the general approach, for both our scenarios.

We performed an extensive analysis of our proposed solutions considering multiple workloads and system configurations. We simulated the performance of our solution, exploiting the AMPL language and the SNOPT non-linear solver, comparing the achieved results with the ones which can be obtained by the major techniques available in the literature or currently used by service providers. From these comparisons emerged that our solution is always cheaper (up to 35%), especially in very noisy traffic conditions, without introducing significant QoS violations. Furthermore, our solutions are very close to the ones found by an oracle with perfect knowledge of the future. In the final phase of this thesis’s work we tested the effectiveness of our approach by performing experiments in a real prototype environment.
running in Amazon EC2 and the results achieved by simulation have been confirmed also in this case.

Future work will be devoted to a deeper investigation of the time scales which can be adopted to govern the behavior of Cloud systems.
Bibliography


BIBLIOGRAPHY


