SPACE4CLOUD
An approach to System Performance and Cost Evaluation for CLOUD

Advisor: Prof. Elisabetta Di Nitto
Co-Advisor: Prof. Danilo Ardagna

Master Thesis by:
Davide Franceschelli
Student ID 770642

Academic Year 2011-2012
# Contents

1 Introduction 1
   1.1 The Cloud Computing ............................... 3
   1.2 Objectives ....................................... 5
   1.3 Achieved Results .................................. 6
   1.4 Structure of the thesis .......................... 7

2 State of the Art 8
   2.1 Overview of the Cloud ............................ 9
   2.2 Multi-Cloud Libraries ............................. 15
      2.2.1 The Jclouds Library ......................... 16
      2.2.2 The δ-cloud Library ....................... 19
      2.2.3 The Fog Library ............................. 21
      2.2.4 The Apache Libcloud Library ............... 21
   2.3 The 4CaaS Project ................................. 25
   2.4 The OCCI Interface ............................... 30
   2.5 The mOSAIC Project ............................... 34
   2.6 The Optimis Project ............................. 41
   2.7 The Cloud4SOA Project ......................... 46
   2.8 Feedback provisioning tools .................... 51
   2.9 Final considerations ............................. 52

3 The Palladio Framework 54
   3.1 Introduction .................................... 54
   3.2 The development process ......................... 55
   3.3 Simulation and system review .................... 67
3.4 Palladio and LQNs .............................................. 68

4 Pricing and Scaling .................................................. 74
  4.1 Google AppEngine ................................................. 75
    4.1.1 Free Quotas ................................................ 76
    4.1.2 Billing Enabled Quotas .................................... 76
    4.1.3 Scaling Policies ........................................... 78
  4.2 Amazon Web Services .............................................. 81
    4.2.1 Free Trial .................................................. 83
    4.2.2 Instance Pricing ........................................... 83
    4.2.3 Storage Pricing ............................................ 92
    4.2.4 Scaling Policies ........................................... 94
  4.3 Windows Azure .................................................... 95
    4.3.1 Free Trial .................................................. 95
    4.3.2 Instance Pricing ........................................... 96
    4.3.3 Storage Pricing ............................................ 96
    4.3.4 Scaling Policies ........................................... 98
  4.4 Flexiscale ........................................................ 102
    4.4.1 Instance Pricing ........................................... 103
    4.4.2 Storage Pricing ............................................ 104
  4.5 Comparisons ...................................................... 105

5 Meta-Models for Cloud Systems Performance and Cost Evaluation .............. 107
  5.1 Objectives ...................................................... 108
  5.2 The CPIM ......................................................... 111
  5.3 CPSMs Examples ................................................ 120
    5.3.1 Amazon CPSM ............................................... 120
    5.3.2 Windows Azure CPSM ...................................... 131
    5.3.3 Google AppEngine CPSM .................................. 141
  5.4 How to derive CPSMs ........................................... 154

6 Extending Palladio .................................................. 155
  6.1 Mapping the CPIM to the PCM ................................... 155
CONTENTS

6.2 Performance and Cost Metrics ........................................... 161
6.3 The Media Store Example .................................................. 163

7 The SPACE4CLOUD Tool ...................................................... 172
  7.1 Overview ........................................................................... 172
  7.2 Design .............................................................................. 177
  7.3 Cost derivation ................................................................. 181
  7.4 Performance evaluation ..................................................... 184
  7.5 Generated Outcomes ......................................................... 184
  7.6 Execution example ............................................................ 186

8 Testing SPACE4CLOUD .......................................................... 196
  8.1 SPECweb2005 ................................................................. 196
  8.2 Basic Test Settings ............................................................ 199
     8.2.1 PCM SPECweb Model ............................................... 200
  8.3 PCM Validation ................................................................. 207
  8.4 Cloud Providers Comparison ............................................. 215

9 Conclusions ......................................................................... 222

A Cloud Libraries ................................................................. 224
  A.1 Jclouds ............................................................................. 224
  A.2 d-cloud ............................................................................ 229
  A.3 Fog .................................................................................. 249
  A.4 Libcloud .......................................................................... 252
  A.5 OCCI ............................................................................... 255

Bibliography ........................................................................... 260

III
## List of Figures

1.0.1 MODAClouds vision ........................................ 2

2.1.1 State of the art ........................................... 13
2.2.1 δ-cloud - Available Compute Services .................... 20
2.2.2 δ-cloud - Available Storage Services .................... 20
2.2.3 δ-cloud - Overview ....................................... 20
2.3.1 4CaaS - Overall Architecture ............................ 26
2.3.2 4CaaS - Service Lifecycle Manager ....................... 27
2.3.3 4CaaS - Example of REC .................................. 28
2.3.4 4CaaS - Application Lifecycle ............................ 30
2.4.1 OCCI - Overview .......................................... 32
2.4.2 OCCI - Core Model ........................................ 32
2.5.1 mOSAIC - Cloud Ontology ................................. 36
2.5.2 mOSAIC - Cloud Agency .................................. 38
2.5.3 mOSAIC - API Structure .................................. 39
2.6.1 Optimis - Cloud Ecosystem ............................... 42
2.6.2 Optimis - IDE ............................................. 45
2.6.3 Optimis - Programming Model ............................ 46
2.7.1 Cloud4SOA - Methodology ............................... 49
2.7.2 Cloud4SOA - PaaS architecture based on the State of the Art 50
2.7.3 Cloud4SOA - Cloud Semantic Interoperability Framework . 51

3.1.1 Palladio - Developer Roles in the Process Model ........ 55
3.2.1 Palladio - Media Store Example, Repository Diagram .... 57
3.2.2 Palladio - Media Store example, HTTPDownload SEFF .... 58
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.3</td>
<td>Palladio - Media Store Example, HTTPUpload SEFF</td>
<td>59</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Palladio - Example of IntPMF specification</td>
<td>61</td>
</tr>
<tr>
<td>3.2.5</td>
<td>Palladio - Example of DoublePDF specification.</td>
<td>61</td>
</tr>
<tr>
<td>3.2.6</td>
<td>Palladio - Media Store Example, System Diagram</td>
<td>63</td>
</tr>
<tr>
<td>3.2.7</td>
<td>Palladio - Media Store Example, Resource Environment</td>
<td>64</td>
</tr>
<tr>
<td>3.2.8</td>
<td>Palladio - Media Store Example, Allocation Diagram</td>
<td>65</td>
</tr>
<tr>
<td>3.2.9</td>
<td>Palladio - Media Store Example, Usage Model</td>
<td>66</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Palladio - PCM to LQN mapping alternatives</td>
<td>69</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Palladio - PCM to LQN mapping process</td>
<td>70</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Palladio - PCM2LQN mapping overview</td>
<td>73</td>
</tr>
<tr>
<td>4.2.1</td>
<td>AWS - Regions and Availability Zones</td>
<td>82</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Azure - Multiple Constraint Rules Definition</td>
<td>100</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Azure - Constraint and Reactive Rules Combination</td>
<td>101</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Cloud Meta-Model - General Concepts</td>
<td>112</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Cloud Meta-Model - IaaS, PaaS and SaaS Elements</td>
<td>115</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Cloud Meta-Model - IaaS Cloud Resource</td>
<td>119</td>
</tr>
<tr>
<td>5.3.1</td>
<td>AWS CPSM - General overview</td>
<td>122</td>
</tr>
<tr>
<td>5.3.2</td>
<td>AWS CPSM - DynamoDB and RDS Overview</td>
<td>123</td>
</tr>
<tr>
<td>5.3.3</td>
<td>AWS CPSM - RDS Instance</td>
<td>124</td>
</tr>
<tr>
<td>5.3.4</td>
<td>AWS CPSM - EBS and S3 Overview</td>
<td>125</td>
</tr>
<tr>
<td>5.3.5</td>
<td>AWS CPSM - S3 Costs Representation</td>
<td>126</td>
</tr>
<tr>
<td>5.3.6</td>
<td>AWS CPSM - EC2 Overview</td>
<td>128</td>
</tr>
<tr>
<td>5.3.7</td>
<td>AWS CPSM - Spot Instance Costs Representation</td>
<td>129</td>
</tr>
<tr>
<td>5.3.8</td>
<td>AWS CPSM - EC2 Details</td>
<td>130</td>
</tr>
<tr>
<td>5.3.9</td>
<td>AWS CPSM - EC2 Instance Details</td>
<td>132</td>
</tr>
<tr>
<td>5.3.10</td>
<td>AWS CPSM - EC2 Micro Instance Details</td>
<td>133</td>
</tr>
<tr>
<td>5.3.11</td>
<td>AWS CPSM - EC2 Micro Instance Example</td>
<td>134</td>
</tr>
<tr>
<td>5.3.12</td>
<td>Azure CPSM - General overview</td>
<td>136</td>
</tr>
<tr>
<td>5.3.13</td>
<td>Azure CPSM - Blob and Drive Storage overview</td>
<td>137</td>
</tr>
<tr>
<td>5.3.14</td>
<td>Azure CPSM - SQL Database and Table overview</td>
<td>138</td>
</tr>
<tr>
<td>5.3.15</td>
<td>Azure CPSM - Web and Worker Roles overview</td>
<td>139</td>
</tr>
<tr>
<td>5.3.16</td>
<td>Azure CPSM - Virtual Machine overview</td>
<td>140</td>
</tr>
</tbody>
</table>
5.3.17 Azure CPSM - Virtual Machine details .......................... 142
5.3.18 Azure CPSM - Virtual Machine Instance details .......... 143
5.3.19 Azure CPSM - Virtual Machine Medium Instance example 144
5.3.20 AppEngine CPSM - Overview ..................................... 146
5.3.21 AppEngine CPSM - Compute and Storage Services ....... 147
5.3.22 AppEngine CPSM - Datastore and CloudSQL Services .... 148
5.3.23 AppEngine CPSM - Runtime Environments Overview ....... 149
5.3.24 AppEngine CPSM - Runtime Environments Details ........ 151
5.3.25 AppEngine CPSM - Frontend Instances Details .......... 152
5.3.26 AppEngine CPSM - Java Runtime Environment Overview . 153

6.1.1 Palladio - Relations between Software and Hardware Components ........................................... 156
6.1.2 Possible mapping between Palladio and Cloud IaaS ......... 160
6.3.1 Palladio Media Store Example - Resource Environment UML Class Diagram (CPIM) .................. 165
6.3.2 Palladio Media Store Example - Cloud Resource Environment UML Class Diagram (CPIM) ............. 166
6.3.3 Palladio Media Store Example - Detailed Resource Environment UML Class Diagram (CPIM) ............ 167
6.3.4 Palladio Media Store Example - Detailed Amazon Web Services Resource Environment UML Class Diagram (CPSM) 168
6.3.5 Palladio Media Store Example - Detailed Windows Azure Resource Environment UML Class Diagram (CPSM) 170
6.3.6 Palladio Media Store Example - Detailed Google AppEngine Resource Environment UML Class Diagram (CPSM) 171

7.1.1 SPACE4CLOUD - Use Case Diagram ............................ 174
7.1.2 SPACE4CLOUD - Workflow ....................................... 176
7.2.1 SPACE4CLOUD - MVC ........................................... 178
7.2.2 SPACE4CLOUD - Database Schema ............................ 180
7.6.1 SPACE4CLOUD - LoadModel.java, Resource Model ......... 186
7.6.2 SPACE4CLOUD - ResourceContainerSelection.java (1) .... 187
7.6.3 SPACE4CLOUD - CloudResourceSelection.java (1) .......... 188
7.6.4 SPACE4CLOUD - AllocationProfileSpecification.java . . . . . 189
7.6.5 SPACE4CLOUD - CloudResourceSelection.java (2) . . . . . . 189
7.6.6 SPACE4CLOUD - ProcessingResourceSpecification.java . . . 190
7.6.7 SPACE4CLOUD - ResourceContainerSelection.java (2) . . . 191
7.6.8 SPACE4CLOUD - EfficiencyProfileSpecification.java . . . . 191
7.6.9 SPACE4CLOUD - ResourceContainerSelection.java (3) . . . 192
7.6.10 SPACE4CLOUD - ResourceContainerSelection.java (4) . . . 193
7.6.11 SPACE4CLOUD - LoadModel.java, Allocation Model . . . . 193
7.6.12 SPACE4CLOUD - LoadModel.java, Usage Model . . . . . . 194
7.6.13 SPACE4CLOUD - UsageProfileSpecification.java, Closed Work-
load . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
7.6.14 SPACE4CLOUD - Choose.java, Solver Specification . . . . 194
7.6.15 SPACE4CLOUD - LoadModel.java, Repository Model . . . . 195
7.6.16 SPACE4CLOUD - Choose.java, Automatic Analysis . . . . 195

8.1.1 SPECweb2005 - General Architecture . . . . . . . . . . . . . 197
8.1.2 SPECweb2005 - Banking Suite Markov Chain . . . . . . . . 199
8.2.1 Test Settings - Modified Banking Suite Markov Chain . . . 200
8.2.2 Test Settings - Banking Suite, Palladio Repository Model . . 202
8.2.3 Test Settings - Banking Suite, Palladio Resource Model . . 203
8.2.4 Test Settings - Banking Suite, Palladio System Model . . . 203
8.2.5 Test Settings - Banking Suite, Palladio Allocation Model . . 204
8.2.6 SPECweb2005 - Banking Suite, Palladio Usage Model . . . 206
8.3.1 SPECweb2005 VS Palladio - login response times (1) . . . 208
8.3.2 SPECweb2005 VS Palladio - login response times (2) . . . 209
8.3.3 SPECweb2005 - Run with 5100 users, login response times . 210
8.3.4 SPECweb2005 - Run with 5100 users, account_summary re-
sponse times . . . . . . . . . . . . . . . . . . . . . . . . . . . 210
8.3.5 SPECweb2005 - Run with 5100 users, check_details response
times . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 211
8.3.6 SPECweb2005 - Run with 5100 users, logout response times . 211
8.3.7 SPECweb2005 VS Palladio - account_summary response times
(1) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 212
8.3.8 SPECweb2005 VS Palladio - account_summary response times
   (2) ................................................................. 213
8.3.9 SPECweb2005 VS Palladio - check_details response times (1) 213
8.3.10 SPECweb2005 VS Palladio - check_details response times (2) 214
8.3.11 SPECweb2005 VS Palladio - logout response times (1) .... 214
8.3.12 SPECweb2005 VS Palladio - logout response times (2) .... 215
8.4.1 Amazon VS Flexiscale - Closed Workload Population .... 216
8.4.2 Amazon VS Flexiscale - Allocation Profile (partial) .... 217
8.4.3 Amazon VS Flexiscale - login response time ............... 218
8.4.4 Amazon VS Flexiscale - account_summary response times . 219
8.4.5 Amazon VS Flexiscale - check_details response times .... 219
8.4.6 Amazon VS Flexiscale - logout response times ............ 220
8.4.7 Amazon VS Flexiscale - Cost Comparison ................. 221
## List of Tables

2.1   jclouds - List of supported cloud providers .......................... 17
2.2   fog - Cloud Providers and Supported Services ....................... 22
2.3   libcloud - Supported Compute Services and Cloud Providers ........ 24

4.1   AppEngine - Frontend Instance Types ............................... 75
4.2   AppEngine - Backend Instance Types ............................... 76
4.3   AppEngine - Free Quotas ........................................... 77
4.4   AppEngine - Billing Enabled Quotas ............................... 79
4.5   AppEngine - Resource Billings ..................................... 80
4.6   AppEngine - Datastore Billings .................................... 81
4.7   AWS - EC2 Instance Types ........................................... 84
4.8   AWS - EC2 On-Demand Instance Pricing ............................... 86
4.9   AWS - EC2 Light Utilization Reserved Instance Pricing ............ 87
4.10  AWS - EC2 Medium Utilization Reserved Instance Pricing .......... 88
4.11  AWS - EC2 Heavy Utilization Reserved Instance Pricing .......... 89
4.12  AWS - EC2 3-Year RI Percentage Savings Over On-Demand
      Comparison .......................................................... 90
4.13  AWS - EC2 Lowest Spot Price ....................................... 91
4.14  AWS - S3 Storage Pricing ........................................... 93
4.15  AWS - S3 Request Pricing ........................................... 93
4.16  AWS - S3 Data Transfer OUT Pricing ................................ 93
4.17  Azure - Instance Types ............................................. 97
4.18  Azure - Instance Pricing ............................................ 97
4.19  Azure - 6-month storage plan pricing ................................ 98
4.20  Flexiscale - Units Cost ............................................... 103
<table>
<thead>
<tr>
<th>Table Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.21</td>
<td>Flexiscale - Instance Pricing</td>
<td>104</td>
</tr>
<tr>
<td>4.22</td>
<td>Flexiscale - Instance Pricing (worst case)</td>
<td>105</td>
</tr>
<tr>
<td>5.1</td>
<td>Cloud Meta-Model - Cost Ranges</td>
<td>113</td>
</tr>
<tr>
<td>5.2</td>
<td>AWS CPSM - S3 Standard Storage Price Ranges Representation</td>
<td>127</td>
</tr>
<tr>
<td>8.1</td>
<td>PCM SPECweb Model - Resource Demands</td>
<td>207</td>
</tr>
<tr>
<td>8.2</td>
<td>Amazon VS Flexiscale - Response Time Comparison</td>
<td>221</td>
</tr>
</tbody>
</table>
List of Algorithms

A.1 Example of use of the \textit{jclouds} Compute API ............ 225
A.2 Main methods of the class \texttt{ComputeService} of \textit{jclouds} Compute API ......................................... 225
A.3 Context creation with the \textit{jclouds} Blobstore API ........ 226
A.4 Managing the blobstore in \textit{jclouds} with the class \texttt{Map} . . . . 227
A.5 Managing the blobstore in \textit{jclouds} with the class \texttt{BlobMap} . . 227
A.6 Managing the blobstore in \textit{jclouds} with the class \texttt{BlobStore} . . 227
A.7 Managing the blobstore in \textit{jclouds} with the class \texttt{AsyncBlobStore} 228
A.8 Retrieving the unified vCloud API in jclouds .................... 228
A.9 Starting a \texttt{d-cloud} server ........................................ 229
A.10 Using \texttt{GET /api/realms/:id} in \texttt{d-cloud} .................. 230
A.11 Using \texttt{GET /api/hardware_profiles/:id} in \texttt{d-cloud} ........ 231
A.12 Using \texttt{GET /api/images/:id} in \texttt{d-cloud} ................... 232
A.13 Using \texttt{POST /api/images/} in \texttt{d-cloud} .................. 233
A.14 Using \texttt{DELETE /api/images/:id} in \texttt{d-cloud} ............... 233
A.15 Using \texttt{GET /api/instances/:id} in \texttt{d-cloud} .............. 235
A.16 Using \texttt{POST /api/instances/} in \texttt{d-cloud} ................. 236
A.17 Using \texttt{POST /api/storage_volumes in \texttt{d-cloud}} ........... 239
A.18 Using \texttt{POST /api/storage_volumes/:id/attach} in \texttt{d-cloud} . 240
A.19 Using \texttt{GET /api/storage_snapshots/:id} in \texttt{d-cloud} ........ 241
A.20 Using \texttt{POST /api/storage_snapshots in \texttt{d-cloud}} ........ 242
A.21 Using \texttt{GET /api/buckets/:id} in \texttt{d-cloud} ................ 244
A.22 Using \texttt{POST /api/buckets in \texttt{d-cloud}} ................... 245
A.23 Using \texttt{GET /api/buckets/:bucket_id/:blob_id} in \texttt{d-cloud} ... 246
A.24 Using \texttt{GET /api/buckets/:bucket_id/:blob_id/content} in \texttt{d-cloud} 247
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.25</td>
<td>Using PUT /api/buckets/:bucket_id/:blob_id in $\delta$-cloud</td>
<td>248</td>
</tr>
<tr>
<td>A.26</td>
<td>Using the CDN service within $fog$</td>
<td>249</td>
</tr>
<tr>
<td>A.27</td>
<td>Using the Compute service within $fog$</td>
<td>249</td>
</tr>
<tr>
<td>A.28</td>
<td>Using the DNS service within $fog$</td>
<td>250</td>
</tr>
<tr>
<td>A.29</td>
<td>Using the Storage service within $fog$</td>
<td>251</td>
</tr>
<tr>
<td>A.30</td>
<td>Connecting a Driver within $libcloud$</td>
<td>252</td>
</tr>
<tr>
<td>A.31</td>
<td>Creating a node within $libcloud$</td>
<td>252</td>
</tr>
<tr>
<td>A.32</td>
<td>Retrieving the list of nodes belonging to different clouds within $libcloud$</td>
<td>253</td>
</tr>
<tr>
<td>A.33</td>
<td>Executing a script on a node within $libcloud$</td>
<td>254</td>
</tr>
<tr>
<td>A.34</td>
<td>Connection with a cloud provider within OCCI</td>
<td>255</td>
</tr>
<tr>
<td>A.35</td>
<td>Bootstrapping within OCCI</td>
<td>256</td>
</tr>
<tr>
<td>A.36</td>
<td>Create a custom Compute resource within OCCI</td>
<td>256</td>
</tr>
<tr>
<td>A.37</td>
<td>Retrieving the URIs of Compute resources within OCCI</td>
<td>257</td>
</tr>
<tr>
<td>A.38</td>
<td>Deleting a Compute resource within OCCI</td>
<td>257</td>
</tr>
</tbody>
</table>
Sommario


Esistono però anche delle problematiche rilevanti legate all’utilizzo dei sistemi cloud, principalmente derivanti dalla mancanza di standard tecnologici e dalle caratteristiche intrinseche di tali sistemi geograficamente distribuiti. Ad esempio possiamo citare il problema del lock-in che riguarda la portabilità delle applicazioni cloud, il problema della collocazione geografica e della sicurezza dei dati, la mancanza di interoperabilità tra diversi sistemi cloud, il problema della stima dei costi e delle performance.

Questa tesi è focalizzata sul problema della stima dei costi e delle performance dei sistemi cloud a livello IaaS (Infrastructure-as-a-Service) e PaaS (Platform-as-a-Service), di fondamentale importanza per i fornitori di servizi. Questi ultimi necessitano di metriche di confronto valide per scegliere se utilizzare o meno tecnologie cloud e, soprattutto, a quale Cloud Provider affidarsi. La derivazione e l’analisi di queste metriche sono tutt’altro che banali, dato che i sistemi cloud sono geograficamente distribuiti, dinamici e dunque soggetti ad elevata variabilità.

In questo contesto, l’obiettivo della tesi è di fornire un approccio model-driven per la stima dei costi e delle performance dei sistemi cloud. Come si vedrà in seguito, la modellazione di tali sistemi ha coinvolto diversi livelli di astrazione, partendo dalla rappresentazione delle applicazioni cloud per finire con la modellazione delle infrastrutture di alcuni specifici Cloud Provider.
Abstract

Cloud Computing is assuming a relevant role in the world of web applications and web services. On the one hand, cloud technologies allow to realize dynamic system which are able to adapt their performance to workload fluctuations. On the other hand, these technologies allow to eliminate the burden related to the purchase of the infrastructure, allowing more flexible pricing models based on the actual resource utilization. Last, but not least, is the possibility to completely delegate to the Cloud Provider intensive tasks as the management and the maintenance of the cloud infrastructure.

Moreover, the usage of cloud systems can lead to relevant issues, which mainly derive from the lack of technology standards and from the intrinsic characteristics of such geographically distributed systems. For example, we can mention the lock-in effect related to the portability of cloud applications, the problem of data location and data security, the lack of interoperability between different cloud systems, the problem of performance and cost estimation.

This thesis is focused on the problem of performance and cost estimation of cloud system at IaaS (Infrastructure-as-a-Service) and PaaS (Platform-as-a-Service) level, which is crucial for service providers and cloud end users. These latter need valid comparison metrics, in order to choose whether or not to use cloud technologies and, above all, on which Cloud Provider they can rely. The derivation and the analysis of these metrics are not straightforward tasks, since cloud systems are geographically distributed, dynamic and therefore subject to high variability.

In this context, the goal of the thesis is to provide a model-driven approach to performance and cost estimation of cloud systems. As we will see later in this thesis, the modelling of such systems has involved different abstraction levels, starting from the representation of cloud applications and ending with the modelling of cloud infrastructures belonging to specific Cloud Providers.
Chapter 1

Introduction

In a world of fast changes, dynamic systems are required to provide cheap, scalable and responsive services and applications. The Cloud Computing is a possible solution, a possible answer to these requests. Cloud systems are assuming more and more importance for service providers due to their cheapness and dynamicity with respect to the classical systems. Nowadays there are many applications and services which require high scalability, so that for service providers the in-house management of the needed resources is not convenient. In this scenario, cloud systems can provide the required resources with an on-demand, self-service mechanism, applying the pay-per-use paradigm.

The spread of cloud systems has unearthed the other side of the medal: if we use these systems, we have to take into account problems in terms of quality of service, service level agreements, security, compatibility, interoperability, cost and performance estimation and so on. For these reasons, many cloud related projects have been developed around the concept of Multi-Cloud, which is intended to solve most of the aforementioned issues, especially compatibility and interoperability. We will discuss about Multi-Cloud and related projects in the next chapter.

This thesis is focused on the issue related to performance and cost evaluation of cloud systems and is intended to support the MODAClouds\footnote{http://www.modaclouds.eu/} European
Chapter 1. Introduction

Figure 1.0.1: MODAClouds vision

project, which proposes a Model-Driven Approach for the design and execution of applications on multiple clouds [81]. MODAClouds aims at solving the most relevant issues related to the cloud and, in order to achieve this goal, it uses meta-models to represent and generalize cloud systems at different levels of abstraction. According to the MODAClouds vision (Figure 1.0.1), we can distinguish three types of meta-models: the Computation Independent Model (CIM), the Cloud Provider Independent Model (CPIM) and the Cloud Provider Specific Model (CPSM).

The approach we have used is different from the ones which are generally adopted by other several cloud related projects. In the next chapter we will discuss about such projects and approaches, which are generally based on multi-cloud libraries, so they try to face cloud issues like interoperability, compatibility and portability working at API level.

In Section 1.1 we introduce the context providing a definition of Cloud Computing and describing its general features. Then we will discuss about the objectives of this thesis in Section 1.2 and we will anticipate the achieved
1.1 The Cloud Computing

Nowadays there are several definitions of Cloud Computing, but the one given by the National Institute of Standard Technology (NIST) looks the most accurate [1]:

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

This definition highlights the basic properties of Cloud Computing:

- Ubiquity: the user can totally ignore the location of the hardware infrastructure hosting the required service and can use the service everywhere and every time through his client application.

- Convenience: the consumer can use a service exploiting remote physical resources, without necessity of buying/acquiring those resources. He just uses the resources provided by the provider and pays for them with a pay-per-use mechanism.

- On-demand activation: a service consumes resources only when is explicitly activated by the user, otherwise it is considered inactive and the resources needed for its execution can be used for other purposes.

The NIST definition also specifies five essential characteristics of Cloud Computing:

- On-demand self-service: a consumer can use the resources without any interaction with the service provider, with an on-demand policy.
• Broad network access: resources are available through the Internet and can be accessed through mechanisms that promote their use by simple (thin) or complex (thick) clients.

• Resource pooling: physical and virtual resources are pooled to serve many users and are dynamically assigned with respect to the users’ needs and requirements.

• Rapid elasticity: resources can be rapidly and elastically provisioned; the consumer often perceives “unlimited” resources that can be purchased in a very short time.

• Measured service: resources use is always automatically controlled and optimized by monitoring mechanisms at a level of abstraction appropriate to the type of service (e.g., CPU activity time for processing services and so on).

We can distinguish three main service models in the world of Cloud Computing:

• Software-as-a-Service (SaaS): the consumer uses a provider’s application running on a cloud infrastructure. The user can manage only limited user-specific application settings and cannot control the underlying infrastructure.

• Platform-as-a-Service (Paas): the consumer can deploy on the cloud infrastructure owned or acquired applications created using programming languages and tools supported by the provider. The user can control the application and the deployment settings, but cannot manage the underlying infrastructure, or the allocated resources.

• Infrastructure-as-a-Service (IaaS): the consumer can deploy and execute any kind of software on the cloud infrastructure. The user cannot control the underlying infrastructure, but has control over the deployed applications, Operating System, storage, and some network components (e.g., firewalls) and it is responsible of the management of the resources.
Finally, we can distinguish different deployment models:

- Public Cloud: the cloud infrastructure is made available to the general public and is owned by a private organization selling cloud services.

- Community Cloud: the cloud infrastructure is shared among several organizations and supports a specific community with shared concerns. It can be managed by the organizations or a third party and may exist on-premise or off-premise.

- Private Cloud: the cloud infrastructure can be accessed only within the organization and can be managed by the organization itself or a third party and may exist on-premise or off-premise.

- Hybrid Cloud: the cloud infrastructure is composed by different autonomous clouds connected together with a standard or proprietary technology that enables data and application portability.

\section*{1.2 Objectives}

\textit{The objective of this thesis is to exploit the model-driven approach proposed by MODAClouds in order to define suitable meta-models at different abstraction levels, oriented to cloud systems modelling and their performance and costs evaluation.}

As we have seen, the Cloud Computing can offer many interesting features, but at the same time it may introduce some non-negligible issues. Two relevant issues are about performance and cost of systems and applications deployed on the cloud. Nowadays there are a lot of cloud providers and each of them offers a proprietary cloud infrastructures with certain configurations and cost profiles. Cloud users should be able to compare different providers against costs and performance, in order to select the best solution(s). So, in the most general case, cloud users should take into account several different architectures and should be able to evaluate costs and performance for each of them. This task can be very complex and unfeasable if it is carried on
1.3 Achieved Results

Several results have been achieved while trying to reach the goal of this thesis:

- We have analyzed cloud services, cost profiles and scaling policies of Google, Microsoft, Amazon and Flexiscale. The obtained results have been useful to derive the general cloud meta-model (CPIM) and to derive a suitable cost representation.

- We have defined the CPIM and the CPSMs related to Amazon, Google and Microsoft. The CPSMs have been derived using the general concepts represented in the CPIM. Also, a guideline to derive CPSMs starting from the CPIM is provided, taking as example the derivation of the CPSM related to Flexiscale.

- In order to use Palladio for performance and cost analyses, we have provided a mapping between the concepts defined within the CPIM/CPSMs and the ones defined within the Palladio Resource Model.
Finally, a Java tool has been implemented in order to extend Palladio, allowing to run semi-automatic 24-hours analyses and to include cost estimations within them. We have also tested the Java tool in order to make a comparison between Amazon and Flexiscale.

1.4 Structure of the thesis

This thesis is structured as follows.

The state of the art is described in Chapter 2, in which the main approaches to ensure interoperability and compatibility among heterogeneous cloud systems are presented. In this chapter we will also describe several cloud-related projects and multi-cloud libraries which follow these approaches and we will present some existing tools used for performance and cost evaluation.

Chapter 3 is about the Palladio Framework we have extended and used for performance and cost analyses.

In Chapter 4 we report the analysis about cloud services, cost profiles and scaling policies of the cloud systems owned by Amazon, Google and Microsoft.

The general cloud meta-model (CPIM) and its specific derivations (CPSMs) for amazon, Google and Microsoft are presented in Chapter 5.

In Chapter 6 we present the mapping which allow to represent cloud resources within Palladio. Also an example for the Amazon, Google, Microsoft cloud providers is presented.

Chapter 7 describes the Java tool we have implemented to extend Palladio in order to consider 24-hours analyses and cost estimations. The tool has been tested using a Palladio representation of the SpecWeb 2005 benchmark and the results are presented in Chapter 8.
Chapter 9

Conclusions

Cloud Technologies are promising but performance and cost evaluation are challenging for service providers that decide to deploy their applications on cloud systems.

This thesis provides a model-driven approach to address this problem, following the MODAClouds vision. We have proposed a Cloud Provider Independent Model (CPIM) to represent general cloud systems focusing on those aspects which affect performance and costs. From this general representation we have derived examples of Cloud Provider Specific Models (CPSMs) related to Amazon Web Services, Microsoft Windows Azure, and Flexiscale.

We have demonstrated that it is possible to integrate these representations with the existing performance and cost evaluation tools extending their performance and cost analysis capabilities to cloud systems. In particular, we have extended the Palladio Framework, using parts of the Palladio Component Model as a Cloud Independent Model (CIM), and integrating within the tool suite our CPIM and CPSM models.

Furthermore, we have implemented a Java tool that exploits the features offered by Palladio to run 24 hours analyses on cloud systems.

Finally, we have performed some tests and validations using the SPECweb-2005 benchmark in order to evaluate the accuracy of our approach.

Results have shown that the performance estimates are accurate for light workloads, while are very conservative for heavy loads. As part of our future
work, we plan to develop ad hoc techniques for estimating Palladio Component Model parameters with a higher accuracy. Furthermore, we plan to introduce some improvements in the SPACE4CLOUD tool, in order to take into account more complex cost metrics (e.g., data transfer and I/O operations). For the moment, the tool supports few cloud providers and cloud services, so we plan to increase the number of supported cloud providers and cloud services. Finally, the auto/semi-automatical definition of the PCM models from the functional description of the application is another promising line of research.
Bibliography


[33] http://www.optimis-project.eu/


Bibliography


[57] https://cloud.google.com/products/


    windows-azure-geo-location-live.aspx
    50).aspx
    50).aspx
[73] Ralf Reussner, Steffen Becker, Erik Burger, Jens Happe, Michael
    Hauck, Anne Koziolok, Heiko Koziolok, Klaus Krogmann, Michael Kuperberg, The Palladio Component Model. Chair for Software Design & Quality (SDQ) Karlsruhe Institute of Technology, Germany. 22 March 2011.


