Improved Self-management of DataCenter Systems Applying Machine Learning

Josep Lluís Berral García

PhD Thesis Defense. UPC-DAC Doctorate Program
22 of November, 2013

Advisors: Prof. Jordi Torres, Prof. Ricard Gavaldà
Improved Self-management of DC Systems Applying ML

...or how we apply modeling and prediction to improve scheduling on datacenters
Introduction

• Multi-DataCenter Scenario

Multi-DataCenter Systems →
[Chapters 8, 9]

Multi-Resource Modeling →
[Chapter 7]

← Mathematical Formalization of the DC
[Chapter 6]

← Ad-hoc Modeling of Server Resources
[Chapters 4, 5]
Introduction

• Problem with DataCenter systems:
  - Complexity of components, elements and usages
  - High consumption waste of energy

• Automation of Management:
  - Not enough “experts” to model every detail
  - … and to manage every element in the system

• Methodologies / research areas:
  - Autonomic Computing (Self-management)
  - Machine Learning (Automatic modeling and prediction)
Introduction – Motivation and Goal

- **Manage complex DC systems automatically:**
  1. To make easier administrators’ and operators’ work
  2. By making decisions automatically
  3. While offering good Quality of Service
  4. Also reducing energy consumption
  5. And obtaining knowledge from the system

- **Goal of this Thesis**
  - Demonstrate that with the use of Machine Learning techniques we help to achieve all these points
Hypothesis (simplified):

An Expert might model better than any automatic learning algorithm.

But an automatic learning algorithm will do better than a generic algorithm in the lack of Experts.
Introduction – Outline

- Background Concepts
- Previous Experiences
- Part I: Ad-hoc Modeling of Server Resources
- Part II: Mathematical Formalization of the DC
- Part III: Multi-Resource Modeling
- Part IV: Multi-DataCenter Systems
- Conclusions and Future Work
Background – Scenario

• Actors in the Multi-DC/Cloud

Pay for resources || QoS

Data-Center resource provider

Service Owner

Clients

Cloud and Data-Center

Physical Infrastructure

Virtual Machines (one per user)

Files and Web Services
Background – Scenario (II)

- Virtualized DataCenter Structure
Background – Green Computing

• Consolidation vs. Load Balancing
  – The most of the load on the least of the resources
  – Turn on/Off Resources
  – Resource Usage vs. Quality of Service

• All in all: Resource scheduling problem
Background – AI and Machine Learning

- ML – Supervised Learning

Training

- Example Labeled DataSet
  - Training DataSet
  - Learning Algorithm

Validation

- Validation DataSet
  - Model
  - Validation DataSet
  - Check Model Prediction
  - When Validated

Application

- New Unlabeled Data
  - Data to predict
  - Validated Model
  - Prediction
  - Predicted Data
Previous Experiences

- **User Modeling** [part of Nicolás Poggi PhD thesis]
  - Learn about web-service user behaviors, apply user/load-balance

- **Self-Protection and Security** [part of my M.Sc. Thesis]
  - Model network traffic behavior to detect malicious patterns

- **Self-Healing and Memory Leaks** [part of Javier Alonso PhD thesis]
  - Predict WS memory behavior to detect anomalous patterns
Outline

• Part I

Multi-DataCenter Systems →

Multi-Resource Modeling →

← Mathematical Formalization of the DC

← Ad-hoc Modeling of Server Resources
Ad-hoc modeling of Server Resources

- **Scenario:**
  - Virtualized Cluster [HPC jobs and Web-Services]
  - Quality of Service: execution deadlines and CPU quota
  - Problem: schedule VMs to hosting machines

- **Contributions**
  - First modeling (human expertise base) “by hand”
  - Then: Introduce Prediction of QoS for tentative schedules
  - Consolidate tasks in a datacenter environment
Ad-hoc modeling of Server Resources – Benefits

- **Ad-Hoc Modeling of Virtualized Clusters**
  - Economic-oriented approach
  - Each element contributes with a revenue or a cost
  - Each revenue or cost depends on each scheduling decision
  ... so each scheduling decision depends on the expected benefit

  \[
  \text{benefit} = \text{revenue (running jobs)} - \sum \text{costs (running jobs)}
  \]

  - **Revenue** depends on the **Quality of Service** of run jobs
  - ... determined by a **Service Level Agreement**
**Ad-hoc modeling of Server Resources – QoS & SLA**

- **Service Level Agreement**
  - Contract between Provider and Job/Service Owner
  - E.g.:
    - Accomplishing job finishing before deadline
    - Accomplishing response times below threshold

- If SLA unfulfilled → Revenue receives a penalization
Ad-hoc modeling of Server Resources – QoS/Energy

• Trade-off between energy and SLA fulfillment
  – Consolidation versus Quality of Service
  – Less resources enabled → less consumption → less QoS
Ad-hoc modeling of Server Resources – Costs/Schedule

• When attempting to place a VM in a Hosting Machine
  – Cost of having resources available
  – Cost of SLA violation (penalty)
  – Cost of energy consumption
  – Cost of operation handicaps

• First schedule approach: heuristic algorithms
  – Check movement by movement until find local minimal

• Second schedule approach: consolidation algorithms
  – Dynamic Backfilling: Fill high-loaded hosts from low-loaded ones
Ad-hoc modeling of Server Resources – Predictions

• Substitute some “expert” criteria for scheduling
  – Prediction of SLA and Power Consumption
  – Use of Linear Regression and Regression Trees

• Applying Machine Learning:
  – For each tentative job allocation <host, vms on host+new vm>:
    • Estimation of resulting SLA fulfillment (Machine Learning)
    • Estimation of resulting power consumption (Machine Learning)
    • If they don’t degrade, allocation is viable

• Experiment Results:
  – On characterized workloads (specific HPC): expert is better
  – On uncertain workloads (WS): Dynamic BF+ML performs better
Ad-hoc modeling of Server Resources – Summary

• Summary:
  - Ad-hoc modeling of a virtualized jobs cluster
    • Human expert definition of profits and costs
    • Heuristic solving of scheduling problem
  - ...vs learned predicting models
    • Predict application SLA and power consumption to decide scheduling

• Contributions to the Thesis
  - First approximation to resource and job modeling (hand-made)
  - Learned models to estimate quality of service and power consumption
Outline

• Part II

Multi-DataCenter Systems →

Multi-Resource Modeling →

← Mathematical Formalization of the DC

← Ad-hoc Modeling of Server Resources
Mathematical Modeling of DataCenters

• **Scenario:**
  - Virtualized DataCenter [Web-Services]
  - Quality of Service: web-service “Response Time”
  - Problem: schedule VMs to hosts

• **Contributions**
  - Mathematical modeling of the scheduling problem
  - Prediction of QoS *a priori* for tentative \(\langle \text{vm}, \text{host}\rangle\) placements
  - Prediction of CPU *a priori* for expected web-service loads

Josep Lluís Berral-García
Math Modeling of DCs – Mixed Program

• Find mapping VMs → (hosts × resources) that maximizes
  \[ \text{benefit} = \sum_j (\text{revenue}_j - \text{penalty}(\text{SLA}_j)) - \sum \text{cost}(\text{power}) \]

• Subject to:
  - \( \forall \text{host } h, \text{vm } j: \) Common sense / coherence constraints
  - \( \forall \text{host } h: \) power(h) = \( \text{function}_{\text{power}}(h, \sum_{j \in \text{vm}(h)} \text{resources}_j) \)
  - \( \forall \text{vm } j: \) reqRes\(_j\) = \( \text{function}_{\text{Load}}(j) \)
  - \( \forall \text{vm } j: \) qos(j,h) = \( \text{function}_{\text{QoS}}(\text{resources}_j, \text{reqRes}_j, \sum_{j \in \text{vm}(h)} \text{resources}_j) \)
  - \( \forall \text{vm } j: \) penalty(j,h) = \( \text{function}_{\text{SLA}}(\text{qos}(j,h)) \)

• Mathematical Model
  - Outputs: Schedule optimizing benefit
  - Parameters & variables: hosts, jobs/VMs and SLAs in DC
  - Some functions predicted \([\text{function}_{\text{Load} \sim \text{CPU}}(), \text{f}_{\text{QoS}}()]\) (M5P, Linreg)

Josep Lluís Berral-García
Math Modeling of DCs – Solvers

• MILP Solver
  – If functions are lineal → MILP problem
  – If solution found: we have an optimal solution
    
  - Exact solving is expensive in time / space!

• Heuristic and approximate algorithms
  – Solving the model using first-fit / best-fit algorithms (or others!)
  – We can adjust solving time and space
  – We can insert non-linear functions
    
  - Suboptimal solution is found
Math Modeling of DCs – Experiments

• Testing the model
  – Simulated DC with 40 VMs x 10 hosts, real workload
  – Different tests: change job revenues, power costs, SLA
  – Results as expected. The model is considered valid
**Math Modeling of DCs – Experiments (II)**

- **Heuristics vs ILP Solver**
  - Time to solve: 4 seconds vs +4 hours of complete search
  - Desc. ordered best-fit is really close to complete search!

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Method</th>
<th>Power KwH</th>
<th>Migs</th>
<th>Profit</th>
<th>Avg QoS</th>
<th>Used CPUs</th>
<th>Used hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.09 €/Kwh</td>
<td>First Fit</td>
<td>290.2</td>
<td>1421</td>
<td>177.290</td>
<td>0.519</td>
<td>3523</td>
<td>921</td>
</tr>
<tr>
<td></td>
<td>λ-RR</td>
<td>358.9</td>
<td>520</td>
<td>241.048</td>
<td>0.716</td>
<td>3917</td>
<td>1237</td>
</tr>
<tr>
<td>0.17 €/job</td>
<td>Desc BF</td>
<td>203.4</td>
<td>1665</td>
<td>321.002</td>
<td>0.866</td>
<td>2230</td>
<td>660</td>
</tr>
<tr>
<td>RT ≤ 2 · RT₀</td>
<td>ILP Solver</td>
<td>169.8</td>
<td>1963</td>
<td>326.935</td>
<td>0.878</td>
<td>1568</td>
<td>571</td>
</tr>
<tr>
<td>0.45 €/Kwh</td>
<td>First Fit</td>
<td>293.2</td>
<td>1491</td>
<td>69.871</td>
<td>0.515</td>
<td>3563</td>
<td>970</td>
</tr>
<tr>
<td></td>
<td>λ-RR</td>
<td>230.7</td>
<td>519</td>
<td>132.610</td>
<td>0.604</td>
<td>3985</td>
<td>1161</td>
</tr>
<tr>
<td>0.17 €/job</td>
<td>Desc BF</td>
<td>139.8</td>
<td>1634</td>
<td>255.980</td>
<td>0.818</td>
<td>1504</td>
<td>456</td>
</tr>
<tr>
<td>RT ≤ 2 · RT₀</td>
<td>ILP Solver</td>
<td>151.3</td>
<td>1998</td>
<td>270.543</td>
<td>0.862</td>
<td>1483</td>
<td>503</td>
</tr>
<tr>
<td>0.09 €/Kwh</td>
<td>First Fit</td>
<td>189.8</td>
<td>1513</td>
<td>310.569</td>
<td>0.859</td>
<td>3529</td>
<td>919</td>
</tr>
<tr>
<td></td>
<td>λ-RR</td>
<td>232.7</td>
<td>527</td>
<td>311.619</td>
<td>0.849</td>
<td>4077</td>
<td>1177</td>
</tr>
<tr>
<td>0.17 €/job</td>
<td>Desc BF</td>
<td>155.7</td>
<td>1404</td>
<td>350.892</td>
<td>0.932</td>
<td>1777</td>
<td>508</td>
</tr>
<tr>
<td>RT ≤ 10 · RT₀</td>
<td>ILP Solver</td>
<td>161.1</td>
<td>2007</td>
<td>354.891</td>
<td>0.852</td>
<td>1697</td>
<td>528</td>
</tr>
</tbody>
</table>

- Expandible model. Ex: add migration penalty
Math Modeling of DCs – Summary

• Summary:
  - Present a mathematical model for the “vm×host” allocation problem
  - Predict web-service CPU demands and SLA fulfillments to decide scheduling
  - Test the model against complete solvers and finding approximate algorithms

• Experimentation Results
  - MILP solver can be expensive in time
  - Approximate algorithms results are close to the MILP results

• Contributions to the Thesis
  - We have an extensible model of the system
  - Learned models to estimate CPU requirements and QoS from allocation
Outline

• Part III

Multi-DataCenter Systems →

Multi-Resource Modeling →

← Mathematical Formalization of the DC

← Ad-hoc Modeling of Server Resources
Modeling Additional Resources

• **Scenario:**
  - Virtualized DataCenter [Web-Services]
  - Quality of Service: web-service “Response Time”
  - Problem: model VMs and hosts behaviors

• **Contributions**
  - Prediction of CPU, MEM and I/O *a priori* for expected WS loads
  - Prediction of QoS *a priori* for tentative \( \langle vm, host \rangle \) placements
  - Solver implemented as expert-tuned vs approximate algorithms
Modeling Add. Resources – Learning & prediction

• CPU required by the VM $E[cpu_{vm}] \leftarrow$ from Load

• MEM required by the VM $E[mem_{vm}] \leftarrow$ Load + $CPU_{vm} + mem_{vm_{t-1}}$

• IO required by the VM $E[{in, out}_{vm}] \leftarrow$ Load + $CPU_{vm} + MEM_{vm}$

• SLA-RT provided by the VM $E[rtpr]$
  - Load information $E[requests], E[timepr], E[bytespr]$
  - Resources required by the VM $E[{cpu, mem, in, out}_{vm}]$
  - Occupied resources in the PM $E[{cpu, mem, in, out}_{pm}]$

• Algorithms used: M5P and Linear Regression
Modeling Add. Resources – Experiments

- **Experiments:**
  - Real workloads and environments for learning
  - An analytic simulator to compare ML-augmented algorithms (20 VM x 20 hosts)
  - Real hosting machines for the model validation

- **ML-augmented scheduling algorithms**
  - Versions with ML perform similar or better than non-ML versions
  - ...and relatively well to the expert-tuned algorithms

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>λ-RoundRobin</td>
<td>33.94</td>
<td>2114</td>
<td>0.6671</td>
<td>33</td>
<td>9.416</td>
</tr>
<tr>
<td>BackFilling</td>
<td>31.32</td>
<td>1032</td>
<td>0.6631</td>
<td>369</td>
<td>6.541</td>
</tr>
<tr>
<td>First-Fit</td>
<td>28.77</td>
<td>1874</td>
<td>0.5966</td>
<td>139</td>
<td>6.542</td>
</tr>
<tr>
<td>First-Fit + ML</td>
<td>29.99</td>
<td>1414</td>
<td>0.6032</td>
<td>153</td>
<td>5.000</td>
</tr>
<tr>
<td>Best-Fit</td>
<td>29.85</td>
<td>778</td>
<td>0.5695</td>
<td>119</td>
<td>2.625</td>
</tr>
<tr>
<td>Best-Fit + ML</td>
<td>31.77</td>
<td>1442</td>
<td>0.6510</td>
<td>218</td>
<td>4.625</td>
</tr>
</tbody>
</table>
Modeling Add. Resources – Summary

• **Summary:**
  - Obtained methodology for modeling datacenter resources
  - Predict web-service resource demands and SLA (RT)
  - Test the model against approx. algorithms with ML

• **Learning and Experimentation Results**
  - Regression Trees (CPU, IO, SLA) and LinReg (MEM) to model WS
  - Without Experts: ML-augmented algorithms give better results

• **Contributions to the Thesis**
  - Studied the learning models to estimate resources and SLA
Outline

• Part IV

Multi-DataCenter Systems →

Multi-Resource Modeling →

← Mathematical Formalization of the DC

← Ad-hoc Modeling of Server Resources
Extending to Multi-DC Systems

• Scenario:
  - Network of Virtualized DataCenters [Web-Services]
  - Quality of Service: web-service “Response Time” (+proximity)
  - Problem: include location variables to scheduling problem

• Contributions
  - Expand the mathematical model to a multi-DC system
  - Include elements of geographical location
Extending to Multi-DCs – Scenario

• Network of DataCenters

- Each location has its own energy prices
- Each client connects to our DC network through the closest DC
- Each VM may have clients from around the world
- Each location clients have different “timetables”
Extending to Multi-DCs – Integer Linear Model

• Expand the Mathematical Model

• Expansion of the QoS
  - \( RT = RT_{\text{process}} + RT_{\text{transport}} \)

• Subject to:
  - All previous constraints still apply
  - \( \forall \) host h: \( \text{power}(h) = \text{function}_{\text{Power}}(h, \sum_{j \in \text{vm}(h)} \text{resources}_j, \text{location}_h) \)
  - \( \forall \) vm j: \( \text{migration}_j = \text{function}_{\text{Migration}}(\text{VMimage}_j, \text{previous}_h(j), \text{target}_h(j)) \)
  - \( \forall \) vm j: \( \text{qos}(j,h) = \text{function}_{\text{QoS}}(\text{resources}_j, \text{reqRes}_j, \sum_{j \in \text{vm}(h)} \text{resources}_j, \text{cl}_\text{sources}_j) \)

Josep Lluís Berral-García
Extending to Multi-DCs – Learning & prediction

- **New Infrastructure**
  - HPC hardware (Xeon) → Low energy consumption HW (Atom)
  - Advantage: only need to re-learn ML models

- **Learning on the new scenario**
  - Apply the previous seen techniques for VM CPU/MEM/IO
  - Improvement: learn PM CPU aggregate
  - Improvement: learn QoS as “RT” or “SLA”

<table>
<thead>
<tr>
<th>Predict</th>
<th>ML Method</th>
<th>Correl.</th>
<th>MAE</th>
<th>Err-StDev</th>
<th>Train/Val</th>
<th>Date Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM CPU</td>
<td>M5P ($M = 4$)</td>
<td>0.854</td>
<td>4.41%_{CPU}</td>
<td>4.03%_{CPU}</td>
<td>959/648</td>
<td>[0, 400] %_{CPU}</td>
</tr>
<tr>
<td>VM MEM</td>
<td>Linear Reg.</td>
<td>0.994</td>
<td>26.85 MB</td>
<td>93.30 MB</td>
<td>959/1324</td>
<td>[256, 1024] MB</td>
</tr>
<tr>
<td>VM IN</td>
<td>M5P ($M = 2$)</td>
<td>0.804</td>
<td>1.77 KB</td>
<td>4.01 KB</td>
<td>319/108</td>
<td>[0, 33] KB</td>
</tr>
<tr>
<td>VM OUT</td>
<td>M5P ($M = 2$)</td>
<td>0.777</td>
<td>25.55 KB</td>
<td>22.06 KB</td>
<td>319/108</td>
<td>[0, 141] KB</td>
</tr>
<tr>
<td>PM CPU</td>
<td>M5P ($M = 4$)</td>
<td>0.909</td>
<td>14.45%_{CPU}</td>
<td>7.70%_{CPU}</td>
<td>477/95</td>
<td>[25, 400] %_{CPU}</td>
</tr>
<tr>
<td>VM RT</td>
<td>M5P ($M = 4$)</td>
<td>0.865</td>
<td>0.234 s</td>
<td>1.279 s</td>
<td>1887/364</td>
<td>[0, 19.35] s</td>
</tr>
<tr>
<td>VM SLA</td>
<td>K-NN ($K = 4$)</td>
<td>0.985</td>
<td>0.0611</td>
<td>0.0815</td>
<td>1887/364</td>
<td>[0.0, 1.0]</td>
</tr>
</tbody>
</table>
Extending to Multi-DCs – Experiments

• Intra-DataCenter comparatives
Extending to Multi-DCs – Experiments (II)

• Inter-DC results

- Migrate when being near clients is worth the energy paid

- De-locate when SLA is not degraded if energy cost improves

- When no load, VMs are sent to cheapest place to stay parked
Extending to Multi-DCs – Trade-offs

- Trade-off between energy consumption and SLA (QoS)
Extending to Multi-DCs – Summary

• **Summary:**
  - Expanded the mathematical model for multi-datacenter systems
  - Test the model on a scenario with different prices and communication penalties

• **Learning and Experimentation Results**
  - It is better to learn the SLA variable than the ones affecting it
  - When having diff. energy prices, de-location becomes a good option

• **Contributions to the Thesis**
  - Introduce localization variables to a DC management model
  - Studied learning models on low-power machines and different views of QoS
Outline

• Part IV (+1)

Green Multi-DataCenter Systems →

Multi-Resource Modeling →

← Mathematical Formalization of the DC

← Ad-hoc Modeling of Server Resources
A Green Approach for Placing DataCenters

• Scenario:
  - Place a Network of Virtualized DataCenters
  - “Green” self-powered DCs (Solar and Wind)
  - Problem: grant energy availability, minimize construction costs

• Contributions
  - Study of costs when building a green DC infrastructure
  - Modeling and solving the problem minimizing DC placement costs

Joint work with Í. Goiri and R. Bianchini (Rutgers)
Green DataCenter Placement – Solar and Wind

- Network of Green DataCenters
  - Each location has its own “green” capacity factor
  - Each location has its own construction and land prices
Green DataCenter Placement – Summary

• Summary:
  – Framework selecting the best locations for green powered DCs
  – Characterization of areas around the world as potential locations
  – Trade-offs by quantifying the minimum cost of achieving
    • different amounts of renewable power
    • at different levels of confidence
    • with and without energy backups
  – A model to migrate VMs across DCs, following green availability

• Contributions to the Thesis
  – Address the design and placement of datacenters, focusing on green energy.
Outline

- Conclusions and Future Work
Conclusions & Future Work – Main Contributions

• Problem:
  - We wanted to schedule a multi-DC system autonomically
  - Reducing wasted energy and keeping Quality of Service

• We did it:
  - Using Machine Learning and Autonomic Computing methods

• Main Contributions
  - Formalization of the problem as an extendable Mathematical Model
  - Turned the problem from a human intensive process to a semi-automatic process
  - Introduction of energy-awareness in the modeling
  - Modeling of job and system behaviors through machine learning
  - Modeling of costs for building green-powered DC infrastructures

Josep Lluís Berral-García
Conclusions & Future Work – Future work

• Energy Efficient Techniques & Non-Economical “Green” Strategies

• Learning new kinds of HPC Jobs, Loads, and Service Behaviors

• Improvements on Model Selection & On-Line Model Learning

• Check the scalability of the management approach

• Managing Multi-DC Networks
Conclusions – Contributions of the thesis

• List of Publications:

  - Articles in Conferences

    • Josep Ll. Berral, Íñigo Goiri, Thu Nguyen, Ricard Gavaldà, Jordi Torres, Ricardo Bianchini. “Building Low-Cost Green DataCenters”. To be submitted to a conference next Fall 2013.


Conclusions – Contributions of the thesis (II)

• (List of Publications)
  - Journals and Book Chapters (directly related to thesis)
  - Technical Reports
Conclusions – About this Thesis

• Financed by:
  - Spanish Ministry of Science (FPI Grant BES-2009-011987)
  - Spanish Ministry of Science (projects MOISES-BAR, BASMATI, SESAAME)
  - Spanish Ministry of Science (project CAP-VI)
  - Generalitat de Catalunya (2009-SGR-1428)
  - Emotive-Cloud BSC project
  - EU COST IC0804, Energy Efficiency in Large Scale Distributed Systems
  - EU Pascal2, Pattern Analysis, Statistical Modeling and Computational Learning
  - EU CoreGRID, Foundations, Software Infrastructures and Applications for Large Scale Distributed, GRID and P2P Technologies
Improved Self-management of DataCenter Systems Applying Machine Learning

Josep Lluís Berral García

PhD Thesis Defense. UPC-DAC Doctorate Program
22 of November, 2013

Advisors: Prof. Jordi Torres, Prof. Ricard Gavalda