Power-aware Multi-DataCenter Management using Machine Learning

Josep Ll. Berral, Ricard Gavaldà, Jordi Torres

October 1st – Lyon (France)
Context: Energy, Quality of Service and Self-Management

- Scenario: Multi-Datacenter Network
  - Achieve allocation of virtualized web-services
  - ... keeping good Quality of Service
  - ... reducing energy costs
  - ... and doing this “automatically”
Context: Autonomic Computing and Machine Learning

• Keywords:
  – Autonomic Computing (AC): Automation of management
  – Machine Learning (ML): Learning patterns and predict them

• Applying AC to energy control:
  1. Self-management must include energy policies
  2. Optimization mechanisms are becoming more complex
  3. Decision makers can be improved through adaption over time

• Modeling and prediction:
  – Obtain a predictive model from the system from the past
  – ...using minimal expert knowledge
Introduction

• Energy Saving in Cloud Self-management:
  – Apply the well-known consolidation strategy

• Challenges:
  – How do we consolidate? Optimal place for a job/VM
  – How much resources used? Required resources for the job/VM
  – Resulting QoS / Energy cost in the new placement?

• Contributions:
  – Apply ML to learn about resource performance
  – On a mathematical model for a multi-DC (Benefit-Cost optimization)
  – Also include elements of geographical location (and their properties)
Multi-DataCenter Business Model

Pay for resources || QoS

Data-Center resource provider

Pay for use services

Customer

Clients

Cloud and Data-Center

Physical Machines

Virtual Machines (one per user)

Files and Web Services

- Specific Case of Study:
  - Transactional jobs, Quality of Service (i.e. “Response Time”)

- Problem:
  - As a provider: Schedule properly VMs to PMs
Multi-DataCenter 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”
Problem parts...

1. **Model the multi-datacenter**
   - Create a mathematical model to represent the multi-DataCenter

2. **Fit the model to observations:**
   - Relevant variables only available *a posteriori*
   - ML creates a model from past examples

3. **Solving the optimization problem**
Modeling the Multi-DataCenter

- **Mathematical Model**: Find VMs $\rightarrow$ (hosts $\times$ resources)
  - Profit = Benefits for running VMs – QoS penalties – power costs
  - Outputs: Schedule optimizing profit
  - Constraints: maintaining the consistence of M-DC and operations

- **Quality of Service**
  - $RT = RT_{\text{process}} + RT_{\text{transport}}$ ("Latencies")

- **Subject to**:
  - VM requirements, depending on load
  - Power functions, depending on resources and locations
  - Migration penalties, on distances and VM volumes
  - QoS, depending on resource competence and client distance
Modeling the Multi-DataCenter

- **Mathematical Model:** Find VMs $\rightarrow (\text{hosts } \times \text{resources})$
  - Profit = Benefits for running VMs – QoS penalties – power costs
  - Outputs: Schedule optimizing profit
  - Constraints: maintaining the consistence of M-DC and operations

- **Quality of Service**
  - $\text{RT} = \text{RT}_{\text{process}} + \text{RT}_{\text{transport}}$ (“Latencies”)

- **Subject to:**
  - VM requirements, depending on load
  - Power functions, depending on resources and locations
  - Migration penalties, on distances and VM volumes
  - QoS, depending on resource competence and client distance
Learning and Prediction

• Applying modeling and prediction
  – How much CPU/Mem/IO... will each VM demand?
  – How good will each VM behave?

• Learning on the given scenario
  – Apply ML modeling techniques for VM CPU/MEM/IO
  – Also: learn PM CPU aggregate
  – Also: learn QoS as “RT” or “SLA”

• Benefits:
  – When changing machines, we only need to re-learn ML models
  – We discover the bottlenecks of the system

<table>
<thead>
<tr>
<th></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>Predict 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>Predict 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>Predict 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>Predict 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>Predict 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>Predict 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>Predict 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>
Experiments

- Intra-DataCenter comparatives
  - Using approximate algorithms (ordered best-fit):

![Graphs showing average load, SLA factor, energy consumption, and level of consolidation over time for different algorithms.](image-url)
Experiments

- **Inter-DataCenter results**
  - In a dynamic context, energy savings may increase when consolidating spare VMs
  - Average SLA increases when migration costs are smaller than benefit improvements
  - When no load, VMs are sent to cheapest place to stay parked
  - ML models detect QoS violations better than no ML
Energy/QoS/Load Trade-offs

- Trade-off between energy consumption and SLA (QoS)

![Relation SLA vs Energy vs Load](image1)

![Trade-off Energy vs SLA (QoS)](image2)
Summary

• Focus the “VMs × PMs” allocation problem:
  – With mathematical modeling on multi-datacenter systems
  – Focused on energy consumption and quality of service
  – Usage of automatic modeling through machine learning

• Contributions:
  – Introduce localization variables to a DC management model
  – Studied learning models on different kind of machines and views of QoS
  – Trade-off between SLA fulfillment and energy for transactional jobs

• Learning and Experimentation Results
  – When having different energy prices, de-location becomes a good option

• Future work:
  – Study new relevant variables to the multi-DC model, and other kind of jobs and web-services

Josep Ll. Berral, Ricard Gavaldà, Jordi Torres
Thank you for your attention

Josep Ll. Berral, Ricard Gavaldà, Jordi Torres

berral@ac.upc.edu
(also jlberral@lsi.upc.edu)