Adaptive Scheduling on Power-Aware Managed Data-Centers using Machine Learning

Josep Ll. Berral, Ricard Gavaldà, Jordi Torres

12th IEEE Intl. Conf. on GRID Computing 2011

September 22-23 – Lyon (France)

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Context: Energy, Autonomic Computing and Machine Learning

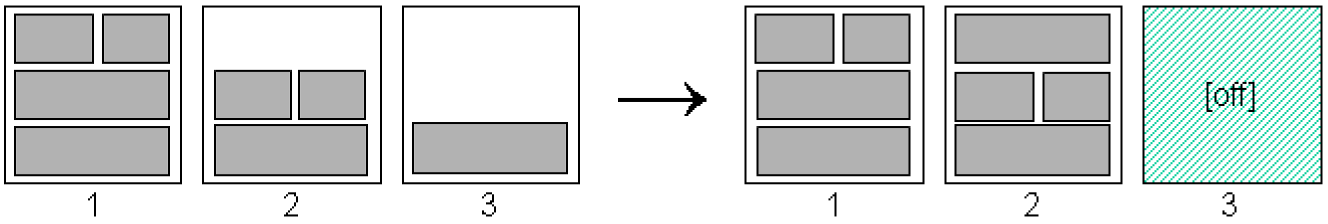
- Keywords:
  - Autonomic Computing (AC): Automation of management
  - Machine Learning (ML): Learning patterns and predict them

- Applying AC and ML 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

- Modeling and prediction:
  - Obtain a model from the system
  - “Let the machine do it for you”

Introduction

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

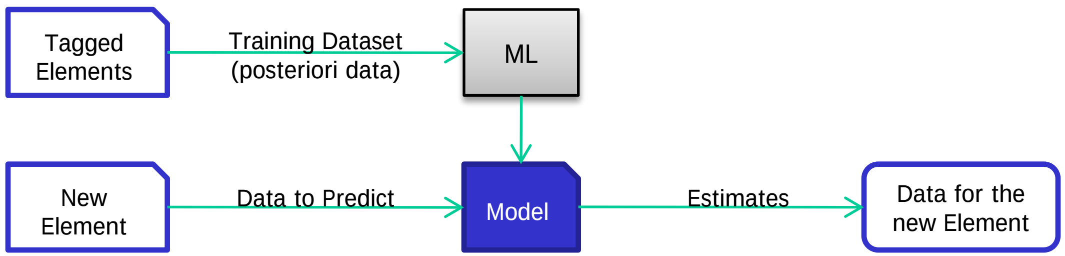


- Challenges:
  - Optimal place for a job/VM?
  - Required resources for the job/VM?
  - Resulting QoS in the new location?

- Contributions:
  - Mathematic model of the cloud/data-center
  - Solve the model as Benefit-Cost optimization
  - Learning and predicting behaviors of jobs and QoS elements

Dividing the problem into parts...

- Model the data-center
  - Create a mathematic model to represent the Data-Center
- Predict relations and goals
  - Relevant variables for decision making only available *a posteriori*
  - ML creates a model from past examples



- Solving the model
  - Complement the model with learned relations
  - Find appropriate algorithm to optimize the model

Modeling the data-center

- Mathematical Model
  - The goal: Profit = Benefits for running jobs/VMs – power costs
  - Outputs: Schedule optimizing profits
  - Constraints: maintaining the consistence of DC and operations
  - Parameters & variables: hosts, jobs/VMs and SLAs in DC

- Find π: jobs → (hosts x resources) that maximizes
  $\sum_j$ (price(j) – penalizations(j, π)) – $\sum_h$ power(h, π)
- Subject to:
  - For all h, jobs that π allocates to h fit in h
  - For all h, power(h, π) = power consumed by host h under jobs j assigned to it by π
  - For all j, QoS(j, π) = QoS of job j given the set of jobs assigned to the h where π places j
  - For all j, penalizations(j, π) = SLAfunction(QoS(j, π))

Modeling the data-center

- Mathematical Model
  - The goal: Profit = Benefits for running jobs/VMs – power costs
  - Outputs: Schedule optimizing profits
  - Constraints: maintaining the consistence of DC and operations
  - Parameters & variables: hosts, jobs/VMs and SLAs in DC

- Find п: jobs → (hosts x resources) that maximizes
  Σ_j (price(j) – penalizations(j, п)) – Σ_h power(h, п)
- Subject to:
  - For all h, jobs that п allocates to h fit in h
  - For all h, power(h, п) = power consumed by host h under jobs j assigned to it by п
  - For all j, QoS(j, п) = QoS of job j given the set of jobs assigned to the h where п places j
  - For all j, penalizations(j, п) = SLAfunction(QoS(j, п))

Machine Learning

Learning relations

- How much CPU will each job (web service in VM) demand?
  - Given the expected or known load (requests and clients)
  - M5P learning algorithm: decision tree + LinReg
    - Load vs Resources: <load~ CPU>
    - Non-linear function approximated by parts
    - Error: ~12% CPU MSE, Stdev 0,1 CPU
  - Use it to fill parameters in the model!

- How good will the job (web service in VM) behave?
  - Given the resources assigned and host characteristics
  - Linear Regression algorithm
    - Context vs Response Time: <given CPU + CPU context ~ RT>
    - Approximation by linear function
    - Error: ~0.00003s MSE, Stdev 0.004s
  - Use it to adjust CPU assignation in solving time!

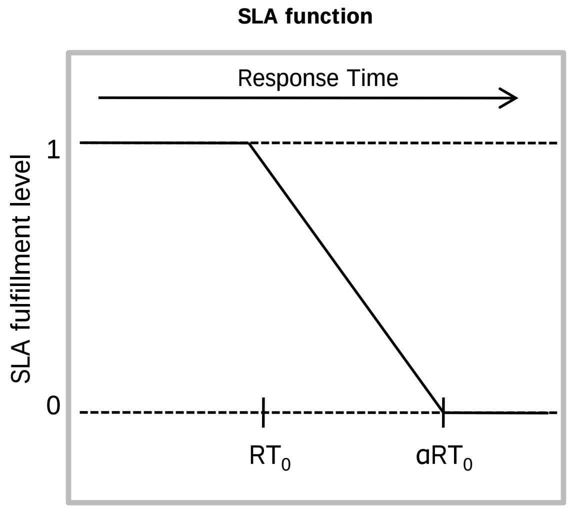
Solving the model

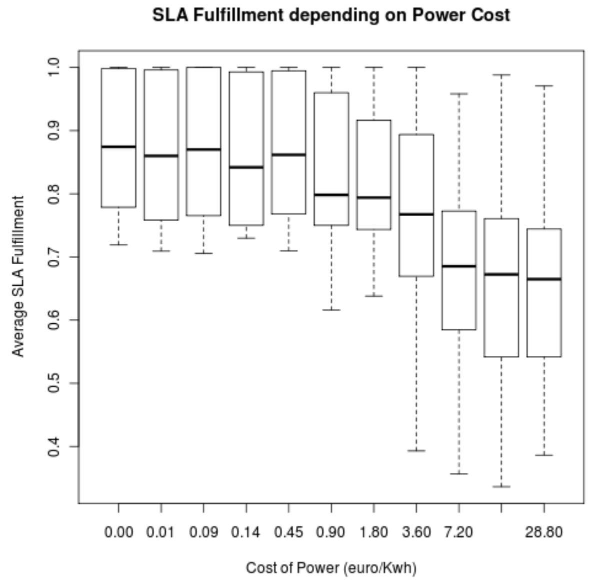
- Linear Functions
  - While all functions involved are written as linear: ILP
  - If solution found: we have an optimal solution
  - ...
  - Solving is expensive in time/space!

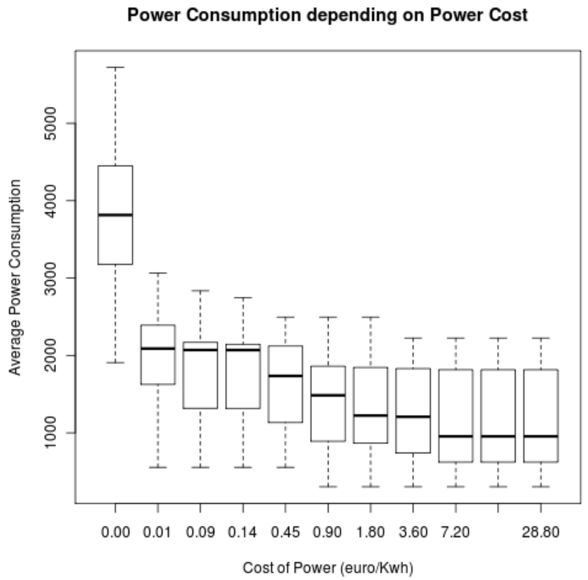
- Heuristic and approximate algorithms
  - Solving the model using first-fit / best-fit algorithms (or others!)
  - Lower bound on optimality assured
  - Solving time and space can be adjusted
  - We can adjust to non-linear functions
  - ...
  - Optimal solution can be missed! (but *how much* missed?)

Experiments

- Testing the model
  - Scenario:
    - Data-center with 40 VMs (web-service) and 10 Xeon 4core machines
    - Workload of real web-applications with variable load
    - ILP solver GUROBI 4.1
  - Different tests: change job revenues, power costs, SLA







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!

Parameters	Method	Power KwH	Migs	Profit	Avg QoS	Used CPUs	Used hosts
	First Fit	290.2	1421	177.290	0.519	3523	921
0.09 €/Kwh	λ-RR	358.9	520	241.048	0.716	3917	1237
0.17 €/job	Desc BF	203.4	1665	321.002	0.866	2230	660
$RT \leq 2 \cdot RT_0$	ILP Solver	169.8	1963	326.935	0.878	1568	571
	First Fit	293.2	1491	69.871	0.515	3563	970
0.45 €/Kwh	λ-RR	230.7	519	132.610	0.604	3985	1161
0.17 €/job	Desc BF	139.8	1634	255.980	0.818	1504	456
$RT \leq 2 \cdot RT_0$	ILP Solver	151.3	1998	270.543	0.862	1483	503
	First Fit	189.8	1513	310.569	0.859	3529	919
0.09 €/Kwh	λ-RR	232.7	527	311.619	0.849	4077	1177
0.17 €/job	Desc BF	155.7	1404	350.892	0.932	1777	508
$RT \leq 10 \cdot RT_0$	ILP Solver	161.1	2007	354.891	0.852	1697	528

Experiments (III)

- Adding a new policy
  - New SLA object: Migrations will penalize the profit
    - We want to reduce migrations, then!
  - Model reduces migrations depending on costs:

Euro/Kwh	Power (KwH)	Migs	Profit (€)	AvgQoS	Used CPUs	Used Hosts
0.00	365.585	195	353.559	0.883	4600	1150
0.01	242.807	225	340.002	0.879	1798	846
0.09	194.974	232	321.319	0.877	1702	662
0.14	170.743	330	312.523	0.878	1619	571
0.45	150.265	445	264.770	0.870	1492	498
0.90	137.242	550	199.360	0.857	1380	455
1.80	128.062	600	81.188	0.840	1327	422
3.60	120.122	776	-136.816	0.815	1285	392
7.20	115.384	889	-559.543	0.776	1274	374
14.40	110.087	837	-134.466	0.737	1248	357
28.80	110.451	913	-293.312	0.738	1251	356

Method	Power (KwH)	Migs	Profit (€)	AvgQoS	UsedCPU	UsedHosts
Desc BestFit	203.175	991	304.885	0.860	2250	658
ILP Solver	194.974	232	321.319	0.877	1702	662

* 0.09 €/Kwh 0.17 €/job RT ≤ 2 · RT0 0.014 €/mig

Conclusions

- Modeled jobs and system behaviors
  - In an automatic manner through Machine Learning
  - Improving decision makers with estimation functions a priori

- Presented a mathematical model for the “job×host” allocation problem

- Tested the model against complete solvers and finding approximate algorithms

- Future work:
  - Extend to multidimensional SLA
  - Extend to multidimensional resource (CPU, memory, disk, database, bandwidth….)
  - Extend to cloud (= many data centers)
  - Extend to more complex economic scenarios (= auctions)

Thank you for your attention

Josep Ll. Berral, Ricard Gavaldà, Jordi Torres

berral@ac.upc.edu

(also jlberral@lsi.upc.edu)