Adaption and Learning on DataCenters and Cloud Systems

Josep Ll. Berral

EEDC Course – Master CANS UPC

Spring 2011

Context: Clouds, Autonomic Computing and Machine Learning

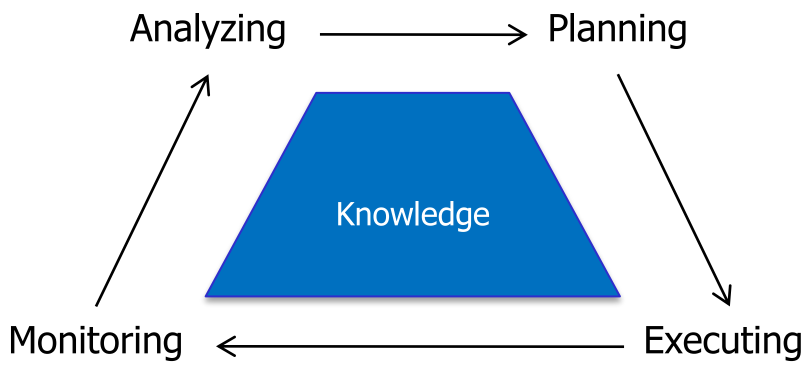
- Keywords:
  - Autonomic Computing (AC): Automation of management
  - Machine Learning (ML): Learning patterns and predict them

- Applying AC and ML to management:
  1. AC allows some degrees of self-management
  2. Optimization mechanisms are becoming more complex
  3. Decision makers can be improved through adaption
  4. ML have methods to extract knowledge from information and data.

- If a system follows any pattern, maybe ML can find an accurate model to help the decision makers and improve policies

Autonomic Computing (I)

- Typical schema on autonomic computing



- AC is based on obtaining and applying knowledge from:
  - Monitoring the system
  - Analyze the monitored data
  - Use this data in order to plan
  - Execute the plan and monitor results

- Until now, this work had to be handled by operators or expert systems

Autonomic Computing (II)

- Dimensions of current Data-Centers *
  - More than 2600 multi-processor nodes running simultaneously
  - More than 400 jobs running, occupying up to 10.000 processors
  - Resource demands of 50.000 processors at the same time
  - Involving other components like shared file-systems, different kinds of networks, …

  - A human operator only can handle basic "eye sight" information
  - … and expert systems can only be based on this information

* Barcelona Supercomputing Center: www.bsc.es (on 2011/May/09)

Autonomic Computing (III)

- Elements change their behavior
  - Jobs can change their behavior and demands
  - Also hardware and environment conditions can change

  - Expert systems can not be built each time
  - Human operators can not predict at detail each change on behaviors
- Remember: Clouds are implemented on (distributed) Data-Centers
- Autonomic Computing is in charge of handle all of this AUTOMATICALLY!

Autonomic Computing (IV)

- The 4 focuses of Self-management:
  1. Optimization of the performance, QoS, consumption…
  2. Configuration of software, networks, …
  3. Protection of execution and data
  4. Healing of degraded systems

- Self-manager must:
  - Know the system at low and high detail, and be adjusted to it
  - Be aware of changes and "understand" these changes
  - Adapt the decision maker towards the constantly changing environment

Autonomic Computing - Summary

This is:

Systems ARE complex, and Clouds even more

Systems are ALWAYS changing ...and Clouds even more!

Human operators and Expert systems can not handle the volume of data/work

Automatic models and adaptive mechanisms are needed

Machine Learning

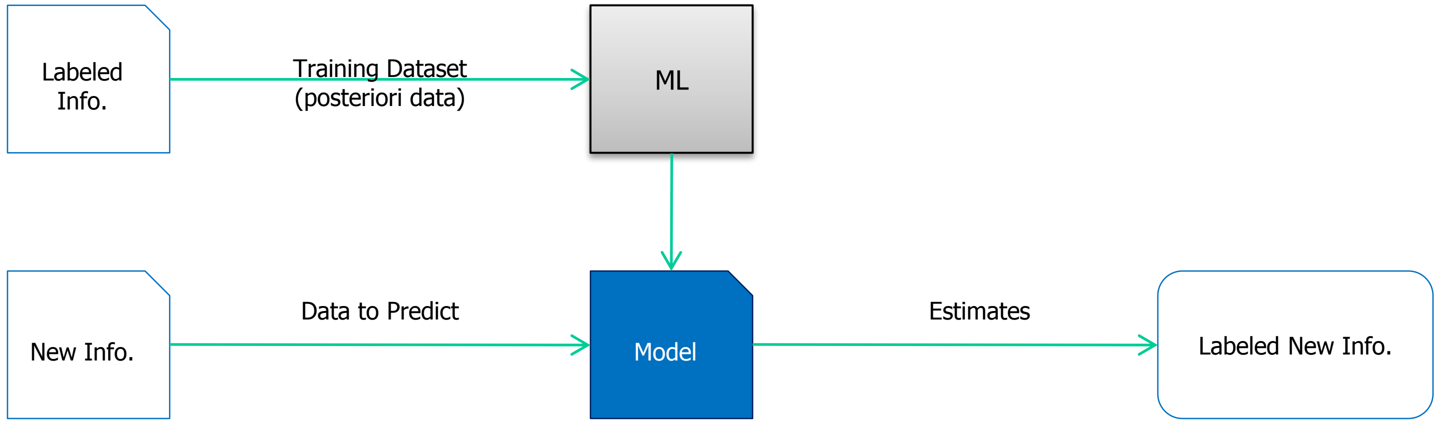
- What is Machine Learning:
  - Machine Learning is part of the Data Mining field
  - … also part of the Knowledge Discovery sciences

  - ML has a set of methods to find models from data/information
  - … so the “machine” learns behaviors from observing the system

- What can Machine Learning do for you?
  1. Modeling: Find a pattern from past or current examples
  2. Prediction: Use patterns to make predictions or estimations
  3. Adaption: Update a pattern with new examples
  4. Discovery: Find the relevant elements that define a system

Machine Learning - Prediction

- Prediction ML schema:



- Typical off-line learner:
  1. Get traces or observed data, and label it with the target information
  2. Train a model from this data, relating the data with the target information
  3. Use the model to predict or estimate the target information for new data

Example of prediction (I)

Learn to predict execution times from given resources:

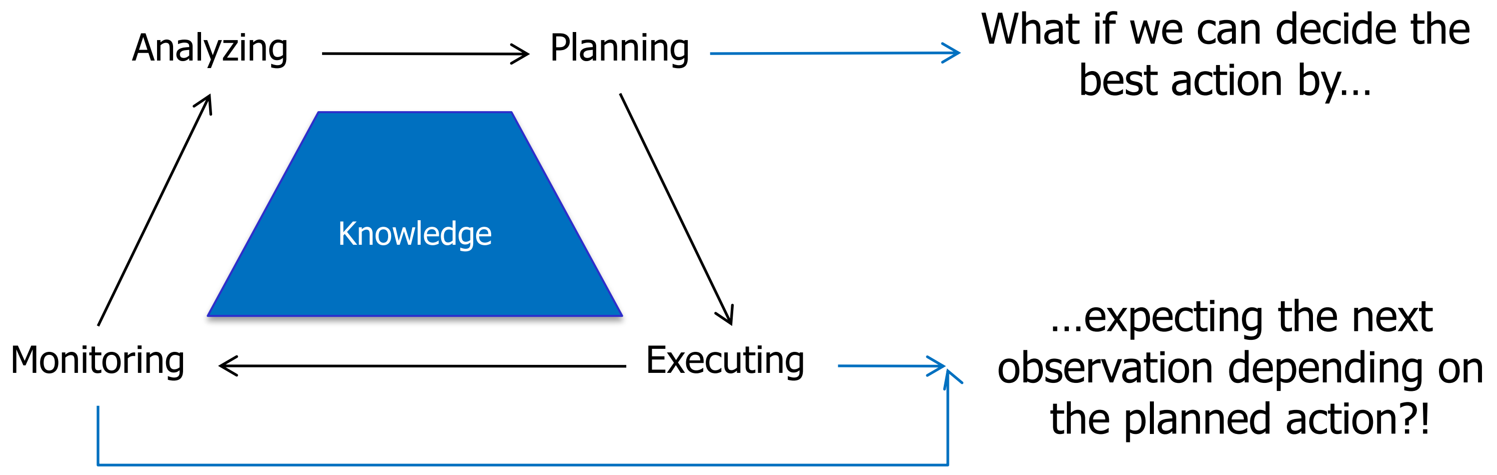
- Learning phase:
  - A job J is given a U amount of resources
  - With these U resources, it needs T time to complete
  - Observing N execution with different U values, we "learn" a function F : U → T

- Prediction phase:
  - Job J is being executed
  - Resources available are V
  - How much time will job J need to complete?

    Just solve learned function F(V) = Estimated T

Example of prediction (II)

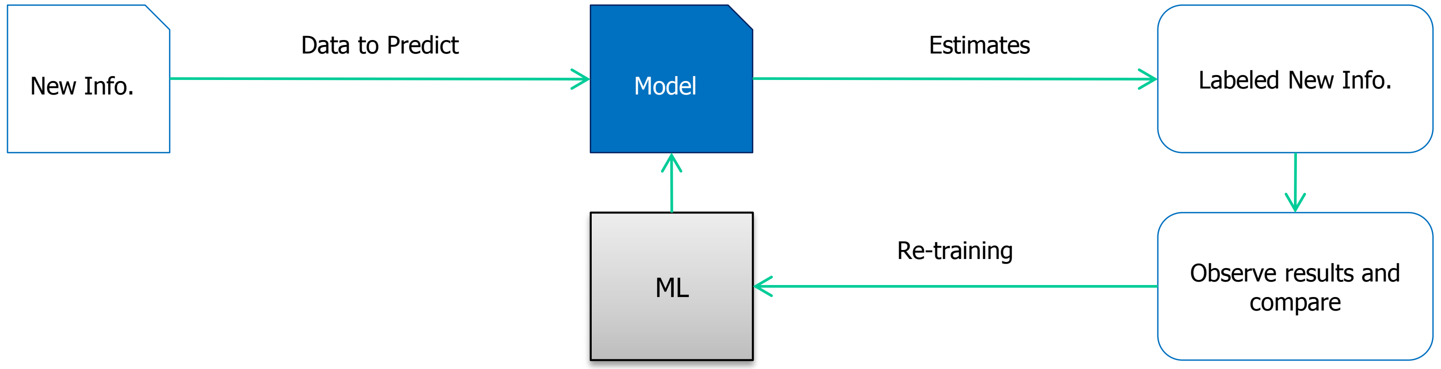
- Predict important information *a priori* :


Analyzing
Planning
Knowledge
Monitoring
Executing
What if we can decide the best action by…
…expecting the next observation depending on the planned action?!

- Basically: predict future monitored variables
  - We get logs from the system, having “a posteriori” information
  - We detect the elements that bring more relevant knowledge
  - ML creates a model able to give that information “a priori”

Machine Learning - Adaption

- Adapting and re-training:



- Typical re-training process:
  1. Estimate target information from new data
  2. Obtain a posteriori the correct result of estimation
  3. Adjust the model using the error and the new information

Example of adaption (I)

Adjust a model predicting visits on a web site:

- Initial model:
  - A web service model learned a relation between "Day Time" versus "#Customers" $F : DT \rightarrow \#C$
  - Also, the model has stored all the samples (the history of logs)

- Change situation:
  - Time ongoing, the web site increases the popularity ($\uparrow$customers)
  - The model starts to become inaccurate

Example of adaption (II)

Adjust a model predicting visits on a web site (cont.)

- Adaption through sliding window:
  - The model discards the oldest examples and rebuilds only using newest samples
  - The number of newest samples to get (window) is determined by how quick the model becomes out of date

- Sliding window
  - A stream or time model like that is based only on the last (useful) samples
  - Kalman filters and other techniques are used to solve the size of window, detection of trends and model corrections

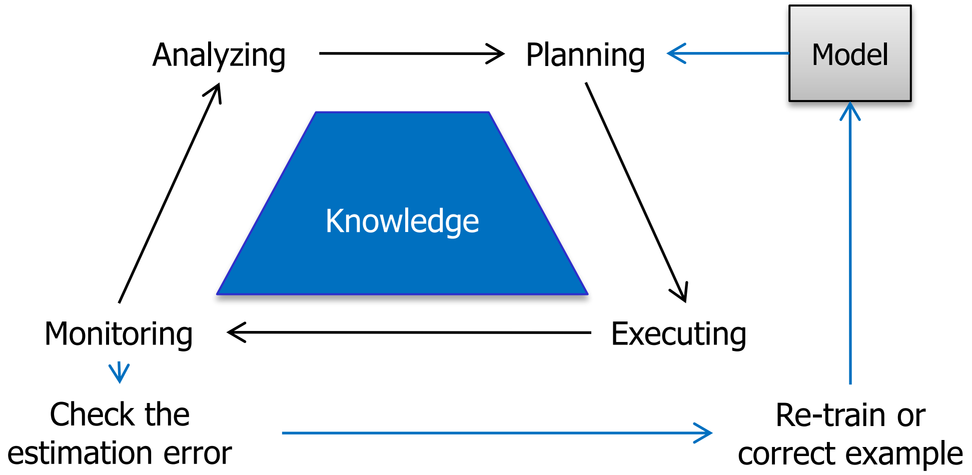
* [1] A.Bifet et al.

Example of adaption (III)

Learning on-line or correcting predictions on a trained system

- Initial model:
  - A model is trained off-line, or is initially “empty”.
  - New input comes from the monitors

- On-Line training / Updating
  - At each loop and inaccurate prediction, the model is re-trained



Example of adaption (IV)

Learning on-line or correcting predictions on a trained system (cont.)

- Previous example on Job timing
  - The job J changes its behavior (runtime environment changes)
  - Now, the learned function $F : U \rightarrow T$ starts being inaccurate

- Adaption process
  - Each time a job finishes, execution time is reintroduced into the model
  - Previous instance may be discarded, or kept
    - Keeping it could be useful if model becomes more complex
    - Adding runtime environment properties to the model, function F could be enhanced!

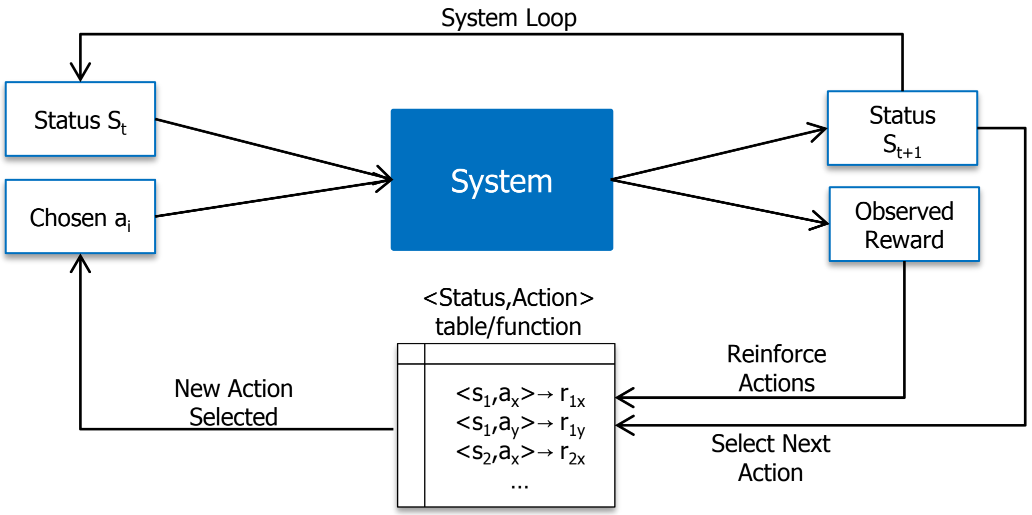
Example of adaption (V)

Adjusting #powered web servers using basic reinforcement learning

- Initial model:
  - A model learned the relation "((start up | shut down) machine | do nothing), web load variation" versus "good action, bad action"

    M : (ΔL , {Start Up, Shut Down, Do Nothing}) $\rightarrow$ [0,1]

- System Loop:
  - At each loop, an action with better rank given ΔL_t is selected and performed, and unused machines or unhandled requests are counted
  - Rank for the pair (selected action, ΔL_t) is promoted/degraded



* [2] G.Tesauro, [6] C.. Watkins et al.

Machine Learning - Clustering

- Clustering / Knowledge Discovery:

  1. Observe the information system offers
  2. Find the relevant data that describes the system
  3. Discover that your system behaves in that way because of ...

  – Different hosts or tasks, apparently different, may behave similar in similar situations
  – A set of similar elements may have very differentiated subsets

- We can group elements by similar behaviors, and observe what make each group different, and what characteristics its elements share

Example of Clustering (I)

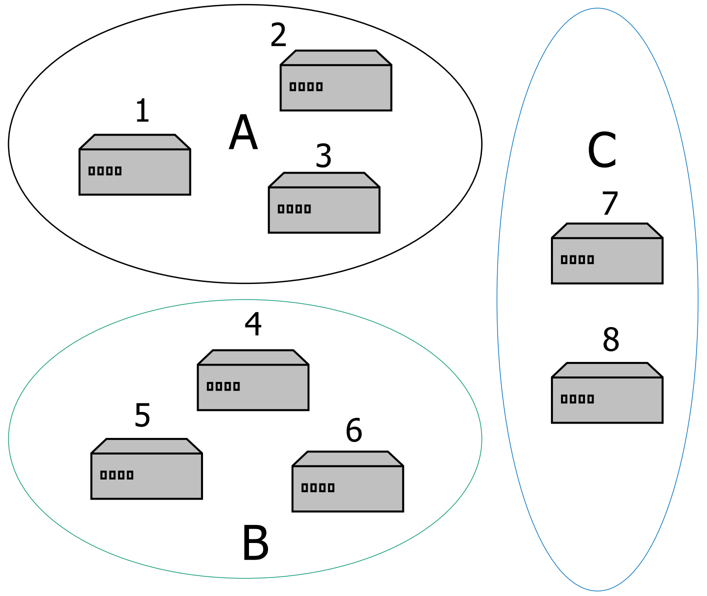
Grouping machines depending on performance

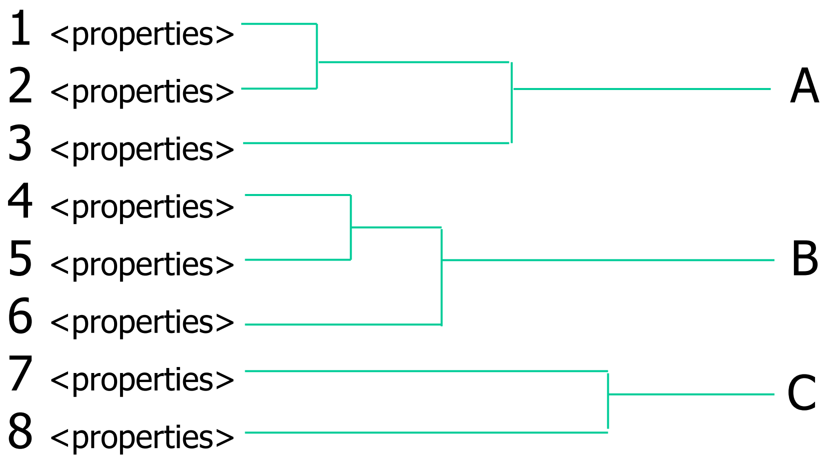
- Context:
  - An heterogeneous set of machines must be grouped to hold specialized kinds of application
  - We have k kinds of applications (“web server”, “data base”, …) and each machine is tested running examples of each kind
- Clustering:
  - Each machine has a vector V of performances with |V| = k
  - Machines are grouped by similarity between their vectors, until we have “N” groups or differences are bigger than “ε”
  - We find that these machines can be grouped in a few big groups

Example of clustering (II)

Grouping machines depending on performance (cont.)

- Discover properties:
  - Each group can contain very different machines, but with similar performances given a set of applications of different kind
  - E.g. group A has good perf. in DBs and Storage, but low perf. with web servers, group B is OK with HPCs and DBs but not others…





Grouping by similarity

Machine Learning - Summary

This becomes:

With ML we can obtaining information, model the system, and predict elements from it

Models can be easily updated or learned from scratch

New knowledge can be retrieved, and the manager can use this knowledge to make better decisions

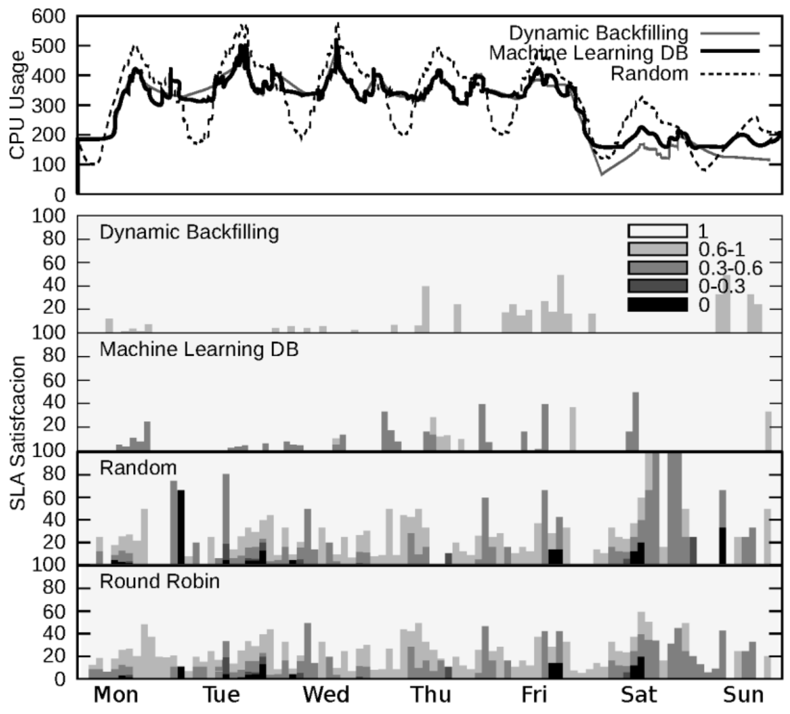
Practical experiments

- Predicting SLA fulfillment levels in a Data-Center
  - Context:
    - We want to reduce number of on-line machines using consolidation
    - A scheduler must schedule jobs with deadline in X hosts
    - Heuristic: predict SLAs for jobs (deadlines) given a candidate host
  - Learning:
    - We train a model using logs from finished jobs, host machines, and execution times
    - Learned function F: (CPU usage, Host Occupation, Deadline) $\rightarrow$ [0,1]
    - 0 ~ Deadline not accomplished, 1 ~ Deadline accomplished
  - Precision of Learning
    - Accuracy around 98.5% on predictions
    - Conservative predictions

* [4] J.Ll.Berral et al.

Practical experiments

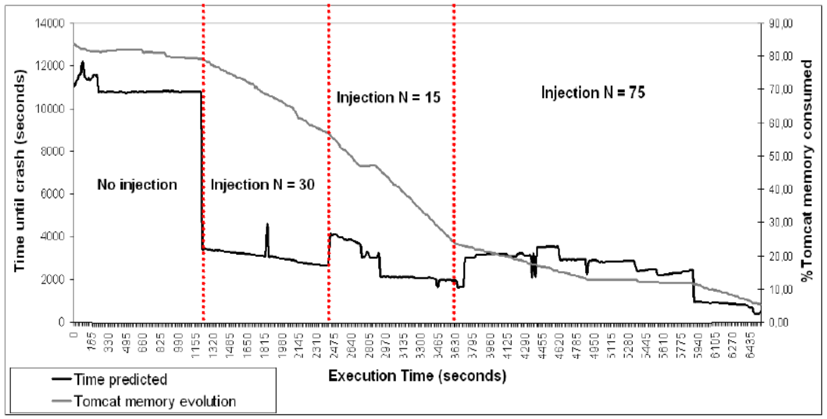
- Predicting SLA fulfillment levels in a Data-Center (Cont)
  - Experimental Environment
    - Simulated datacenter with 400 hosts (4 CPU per host)
    - Use of Linear Regression and M5P for SLA and Power prediction
  - Experimental results
    - SLA fulfillment around 99%
    - Using ML, consolidation techniques are better driven
    - CPU utilization more stable and lower power consumption

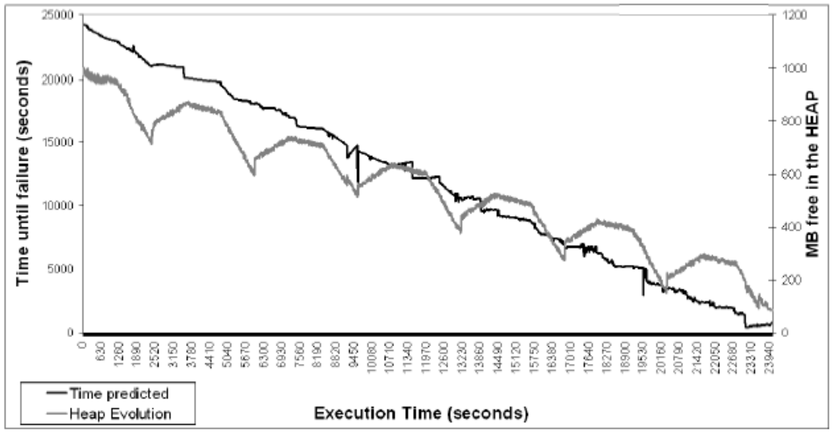


* [4] J.Ll.Berral et al.

Practical experiments

- Predicting Time-to-Crash on memory degraded systems
  - Context
    - A system monitor must schedule machine reboots according to memory degradation
  - Modeling
    - Learn a function to determine the time to crash from memory status
    - Machine learning used algorithm are linear regression and M5P
  - Results
    - The mean average error of prediction is around 16 minutes

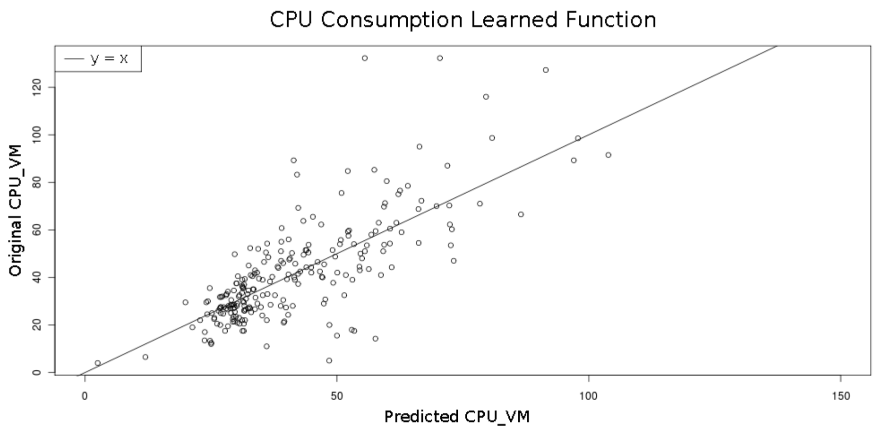


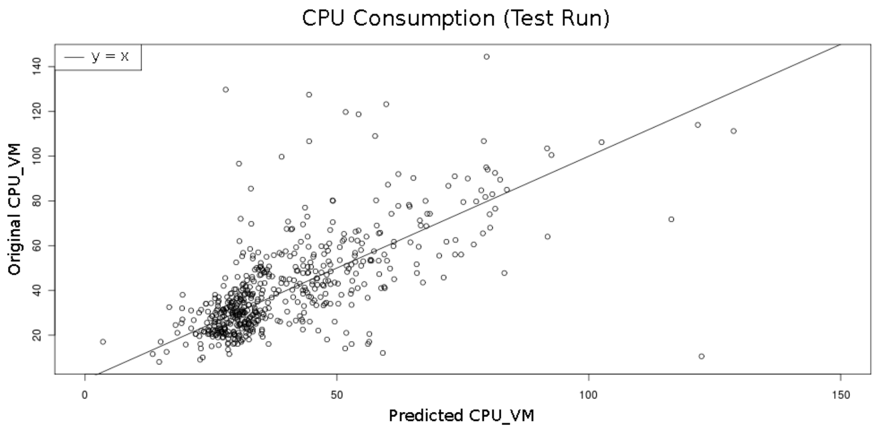


* [3] J.Alonso et al.

Practical experiments

- Predicting CPU load on Xeon Machines:
  - Context:
    - Schedule web services according CPU requirement
  - Modeling:
    - Learn a function to determine the CPU load depending on requests for an Apache WS
    - Machine learning used algorithm is M5P and linear regressions
  - Results:
    - Error on prediction: ~11%





* [5] J.Ll.Berral et al.

Summarizing

- Given the size and volume of resources, data and work on Clouds and Data-centers, manage them with some precision is quite difficult
- Autonomic Computing is in charge of allow systems manage themselves
- Machine Learning brings methods to obtain accurate models to make better managing decisions
- Clouds and Data-Centers can be better managed if they are able to learn from their behaviors and adapt themselves

Bibliography

- Some useful bibliography

[1] Kalman filters and adaptive windows for learning in data streams. Albert Bifet and Ricard Gavaldà. Shorter version in Proc. 9th International Conference on Discovery Science (DS 2006). Springer Lecture Notes in Artificial Intelligence 4265, 29-40.

[2] Reinforcement Learning in Autonomic Computing: A Manifesto and Case Studies. Gerald Tesauro, IEEE Internet Computing, pp. 22-30, January/February, 2007

[3] Adaptive on-line software aging prediction based on Machine Learning. Javier Alonso, Josep Ll. Berral, Ricard Gavaldà, Jordi Torres. 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2010). June 28th - July 1st 2010, Chicago (USA).

[4] Towards energy-aware scheduling in data centers using Machine Learning . Josep Ll. Berral, Iñigo Goiri, Ramon Nou, Ferran Julià, Jordi Guitart, R. Gavaldà, J. Torres. First Intl. Conf. on Energy-Efficient Computing and Networking. Passau, April 13-15, 2010.

[5] Adaptive Scheduling on Power-Aware Managed Data-Centers using Machine Learning. Josep Ll. Berral, Ricard Gavaldà, Jordi Torres. Submitted to Cluster 2011

[6] Technical Note: Q-Learning. Christopher J. C. H. Watkins and Peter Dayan. 1992. Mach. Learn. 8, 3-4 (May 1992), 279-292

[7] Ian H. Witten; Eibe Frank (2005). "Data Mining: Practical machine learning tools and techniques, 2nd Edition". Morgan Kaufmann, San Francisco. http://www.cs.waikato.ac.nz/~ml/weka/book.html

Thank you for your attention

Josep Ll. Berral

berral@ac.upc.edu

(also jlberral@lsi.upc.edu)