Bayllocator: A proactive system to predict server utilization and dynamically allocate memory resources using Bayesian networks and ballooning

Evangelos Tasoulas - University of Oslo
Hårek Haugerud - Oslo And Akershus University College
Kyrre Begnum - Crayon NSA

LISA / USENIX 2012

Date

Motivation

- Consolidation of virtual machines is the norm
- Most systems are idle most of the time, so we can overcommit resources
- However, when they need to do their job, they need their resources
  - *... and could benefit if others don't need theirs*
- Reactive resource allocation may not be fast enough
  - The allocation mechanism is sometimes slow in itself

Problem statement and approach

To get a flexible approach on improving utilization efficiency of hardware resources in a single hypervisor running virtual machines

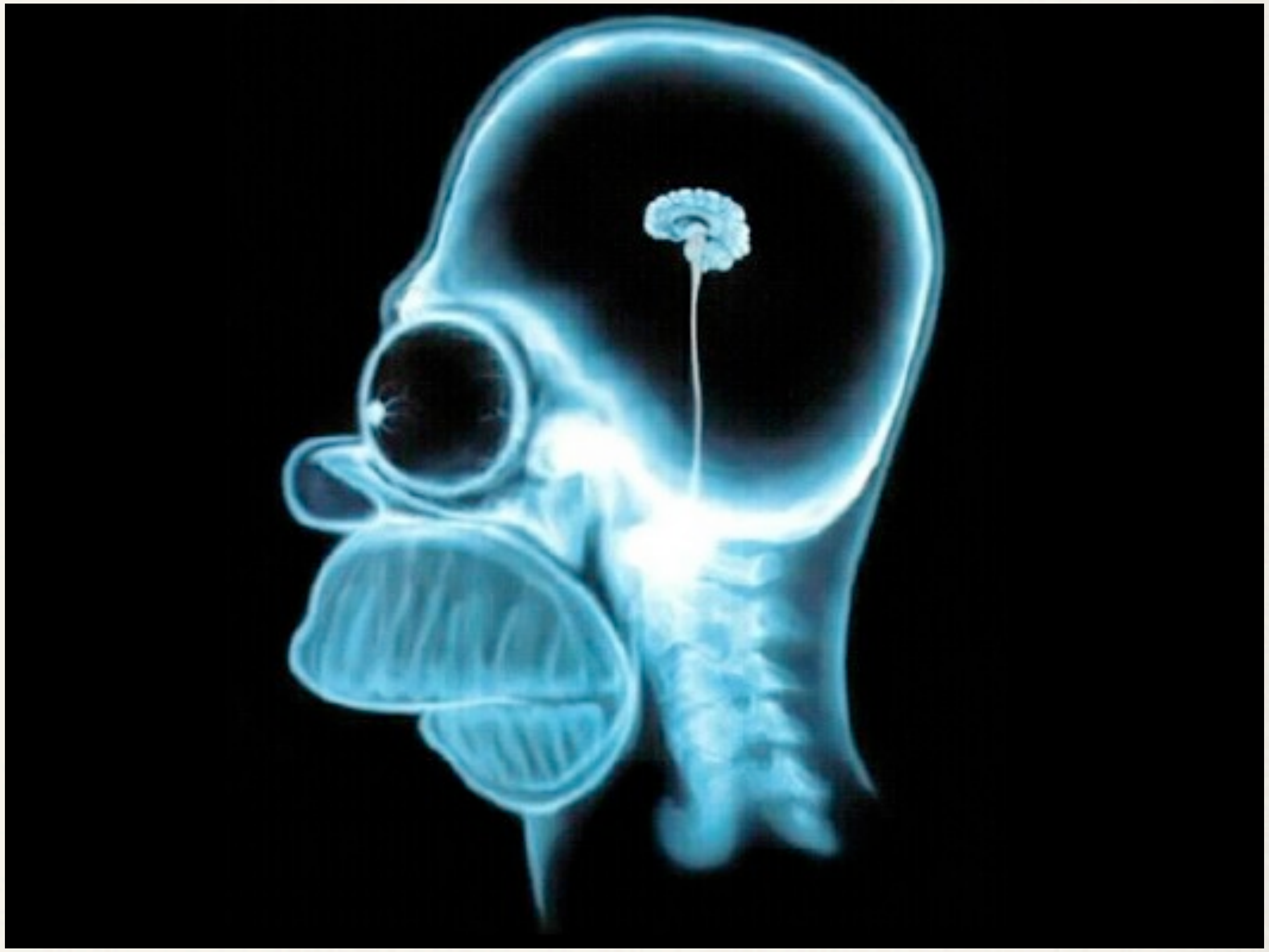
Approach:

- A proactive system to predict server load using Bayesian networks, and dynamically allocate resources on virtual machines running on a single hypervisor to provide more efficient utilization of the physical hardware

- Currently only focused on dynamic memory allocation using ballooning

Ballooning

- A kernel module is loaded into the kernel of the VM
- It can be controlled from the hypervisor to allocate or release memory
- The VM will therefore believe to have the same amount of memory, but have some of it locked



Bayllocator will ...

- Assume an over-provisioned environment
- Make a prediction of how much memory each VM needs + a defined percentage
- Ensure that a VM's min/max values are not violated
- Avoid Hypervisor swapping
- Distribute excessive memory fairly to all VMs
- Claim memory fairly

Prediction Algorithm

- Memory consumption is divided into categories
  - Each category spans 100MB, f.e 400_500MB
- Given the input of: VM, weekday, time interval and current consumption, a probability is calculated for all categories
- Each probability is multiplied with a number representing the category
  - In our case: 400_500MB -> 450MB * P(400_500MB)

Simple example

1. Calculate new probabilities using Bayesian network

```
$ QueryNet .R TestServer1.dat FMU \
> Monday h07 m55_59 mem_100_200
```

mem_100_200, 0.148

mem_200_300, 0.306

mem_300_400, 0.492

mem_400_500, 0.054

2. Use probabilities to calculate memory demand

$150 \cdot 0.148 = 22.2$

$250 \cdot 0.306 = 76.5$

$350 \cdot 0.492 = 172.2$

$450 \cdot 0.054 = 24.3$

$22.2 + 76.5 + 172.2 + 24.3 = 295.2$ MB

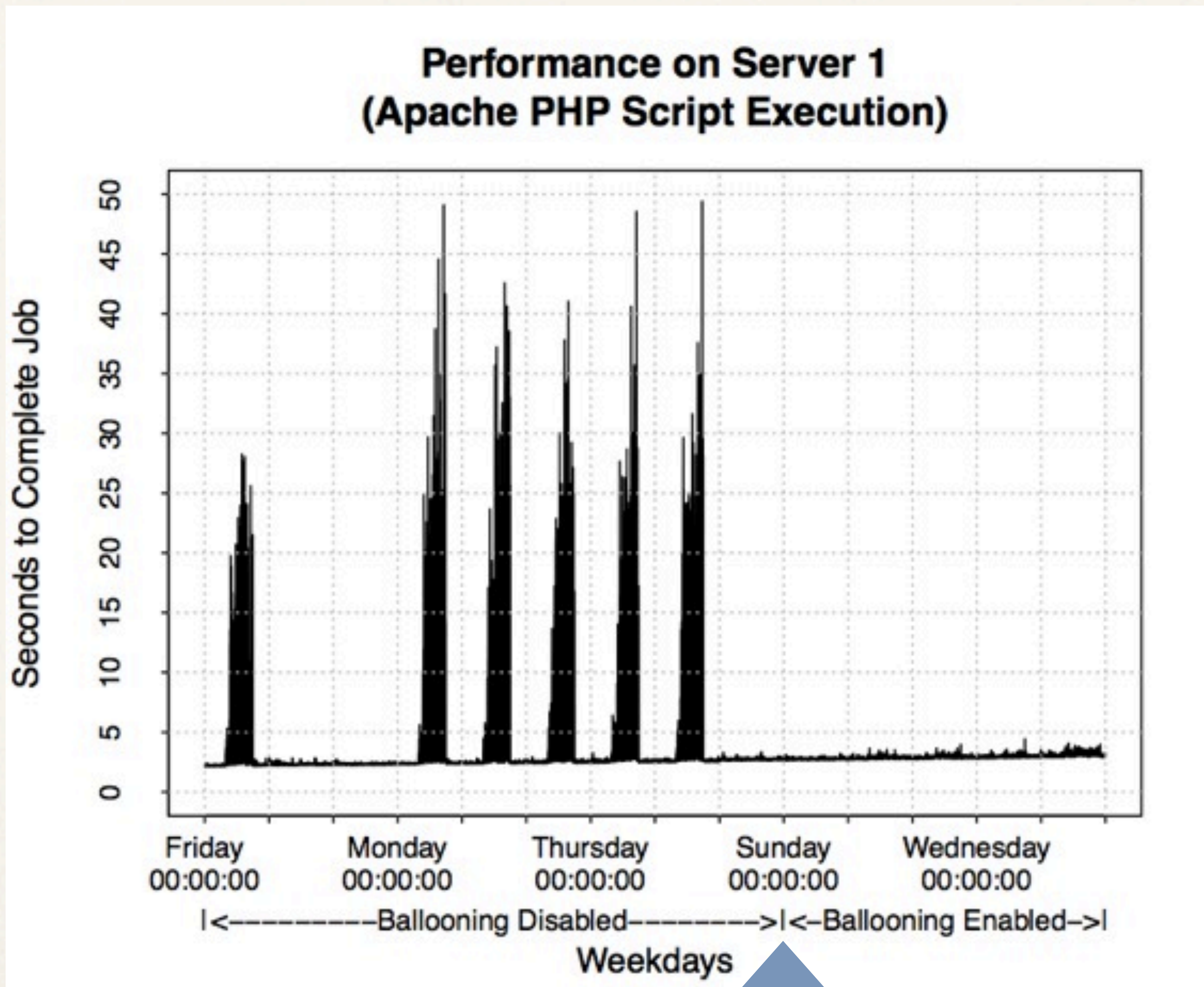
The redistribution of wealth

- There may not be enough memory to meet all the predicted allocations
- *Fair* - Large VM's will need more memory, but should be expected to share more too
- In practice, use a percentage rather than memory values when awarding / claiming memory

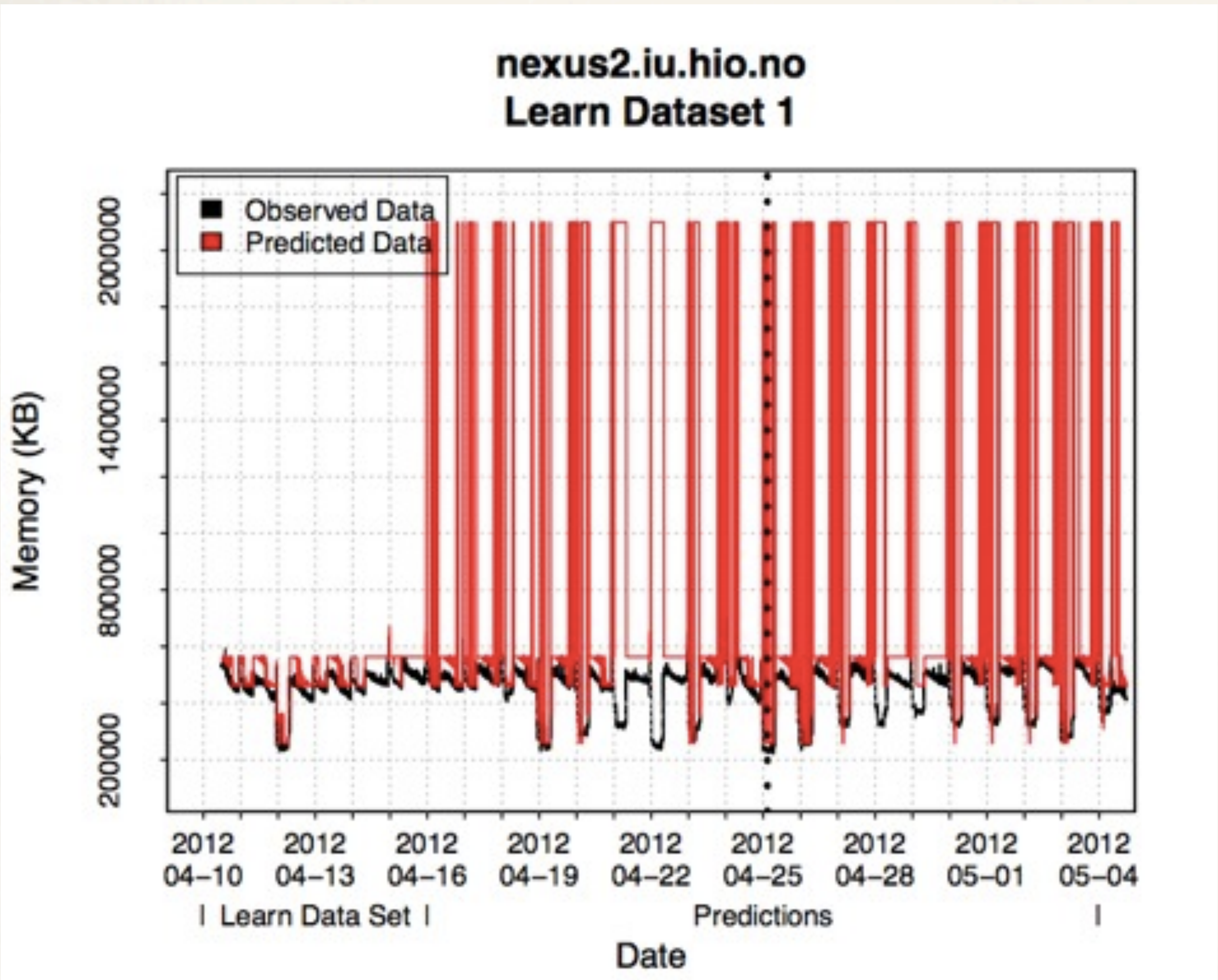
Experimental setup

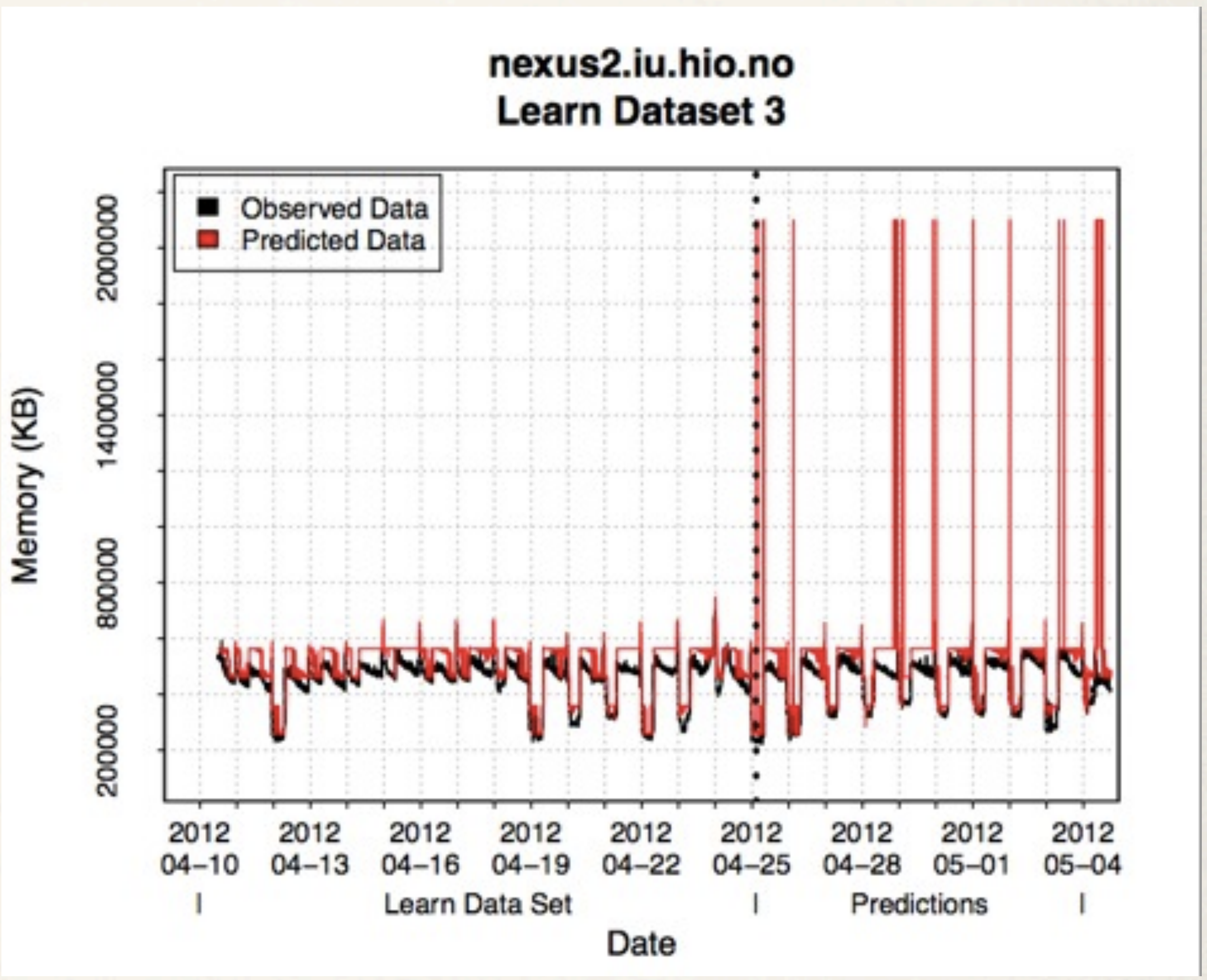
- Both generated data and replays of real data
- Single KVM hypervisor with several VM's
- Special script mimicked workloads inside the VM
- Collected usage data into a DB and wrote a prototype using R and Perl

Results - Simulated data



Results - Real-life replay





Predictions start

Predictions start

Results - Learning periods and accuracy

- The more observed data, the better the accuracy of the predictions
- A fundamental difficulty with using historical data is that you need historical data
  - High-quality historical data is hard to come by
  - Often times underlying trends or gaps pollute the data

Discussion

- A virtual machine will never get less memory than has ever been observed on it*
- "Why not choose the category with the highest probability, instead of calculating the sum?"
- "What is the training time?"
- "Why use only temporal parameters?"
- "Can I run it and expand on the bayesian network myself?"
- Bayllocator is not about reactive, flashmob-mitigation

Future work

- Weight-lifting for servers: If we knew the signs of a flashmob, we could train the model for it.
- Investigate the outer limits of ballooning under extreme circumstances
- Comparison with memory de-duplication
- Combining Bayllocator with reactive behavior

Thank you!

Questions?