Estimating system-wide power consumption through artificial neural networks

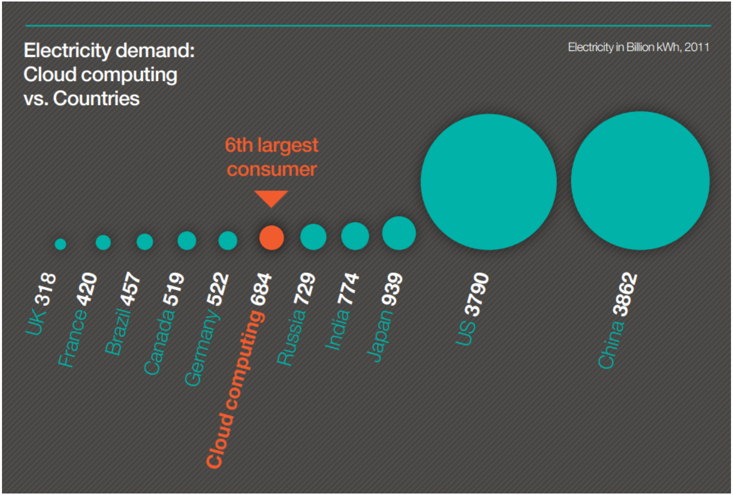
L. F. Cupertino, G. Da Costa and J. M. Pierson

IRIT - Toulouse Institute of Computer Science Research
UPS - University of Toulouse (Paul Sabatier)

GreenDays@Rennes – 02/07/2014

Outline

Introduction

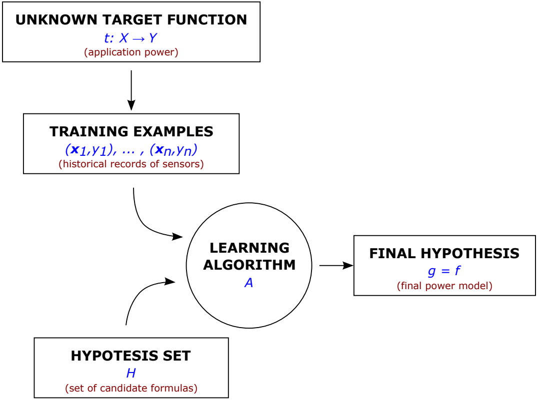


"Clicking Clean: How Companies are Creating the Green Internet," Tech. Report, Greenpeace, April 2014.

Motivation

- Power consumption varies according to servers' usage
  - Energy efficient scheduling
  - Power estimation of applications
- Curent models limitations
  - Application specific or lack accuracy
  - Hardware dependent
- External power meters
  - Easy to measure
  - Energy providers charge power in AC mode

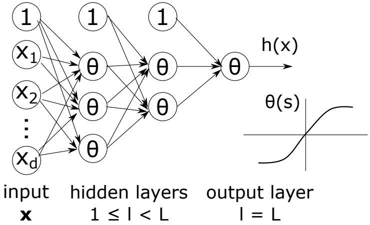
Estimating power with ANN



- Definition of the Training Set
- Performance indicators
- Accurate power measurements

Artificial Neural Networks

- Feedforward Multilayer Perceptron Network topology



$$x_j^{(l)} = \theta\left(\sum_{i=0}^{d^{l-1}} w_{ij}^{(l)} x_i^{(l-1)}\right) \tag{1}$$

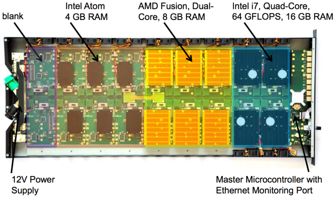
where $\theta(s) = \tanh(s)$

- Backpropagation *

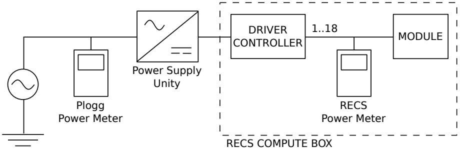
$$\Delta w_{ij}^{(l)} = -\eta x_i^{(l-1)} \delta_j^{(l)} \tag{2}$$

$$\delta_i^{(l-1)} = \left(1 - \left(x_i^{(l-1)}\right)^2\right) \sum_{j=1}^{d^{(l)}} w_{ij}^{(l)} \delta_j^{(l)} \tag{3}$$

Environment setup



- RECS compute box
  - Processor: Intel Core i7-3615QE
  - Memory: 16GB of RAM
  - NIC: Intel 82579LM Gigabit Ethernet
  - Disk: diskless environment
- OS: Scientific Linux release 6.4 (kernel v2.6.32)
- Power meter: Plogg

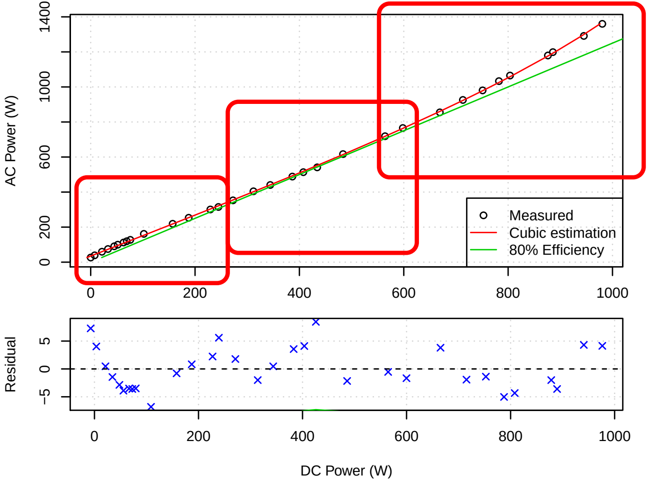


- Modules management and power monitoring is done by an external server to not impact the measurements.

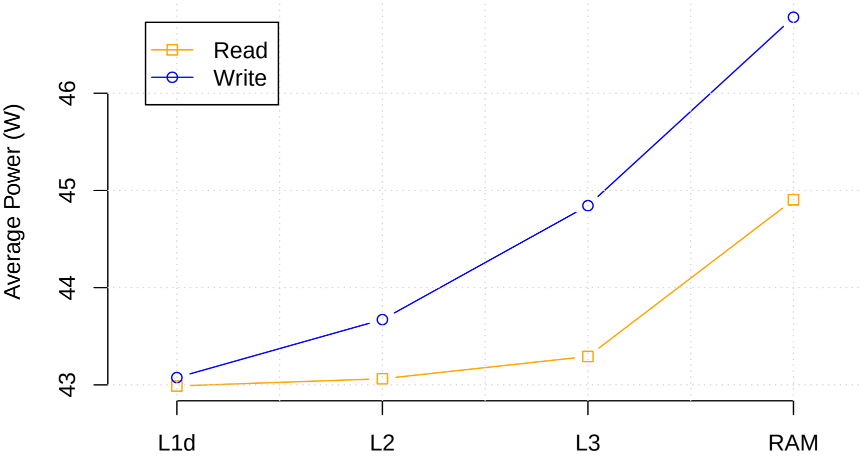
L. Cupertino et al. “Towards a Generic Power Estimator,” Computer Science – Research and Development, 2014 (in press).

Modeling PSU's conversion losses

- PSU Input/Output power data provided by PSU's vendors
- Cubic model: $DC \sim w_0 * AC + w_1 * AC^3$

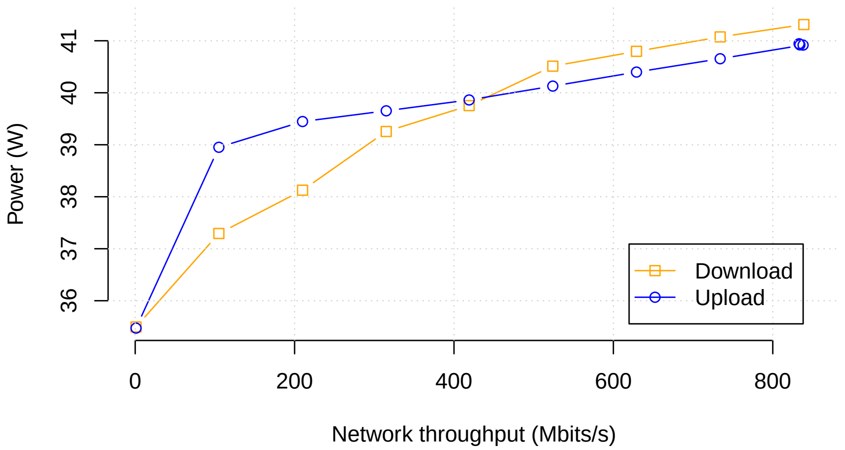


Memory power profile

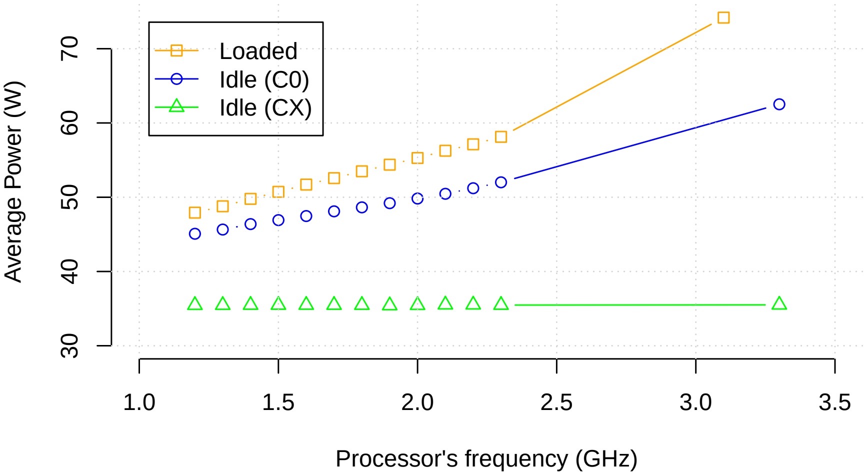


Network usage

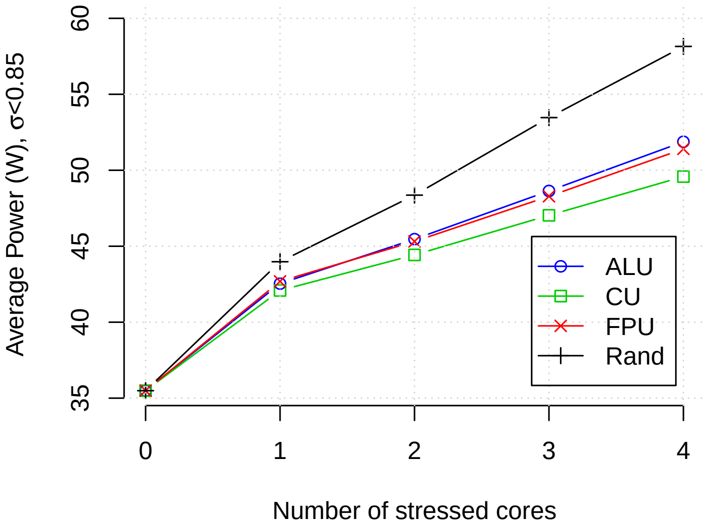
- Benchmark: iperf3
- CPU usage $< 5\%$



Dynamic Voltage and Frequency Scaling

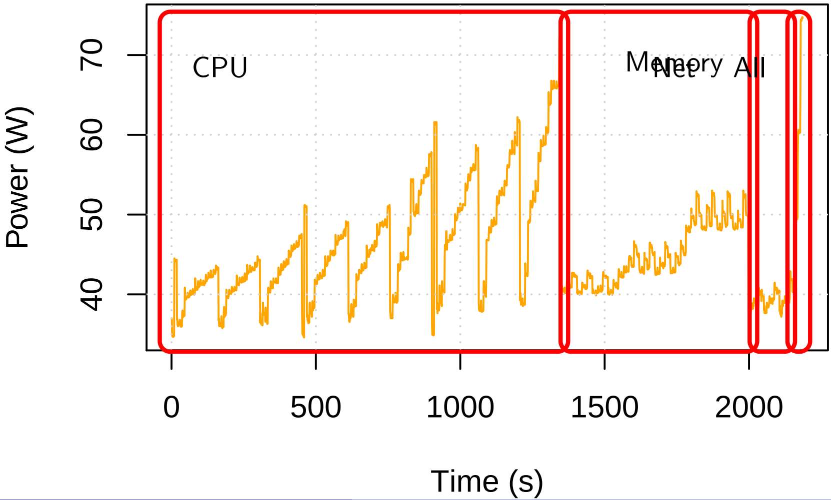


Power profile of computation intensive applications



Training set's power profile

- Training set: CPU, memory and network stressed
- 3 frequencies: min (1.2GHz), max (2.3GHz) and boost (3.1GHz)



Calibrated capacitive model

- Capacitive model:

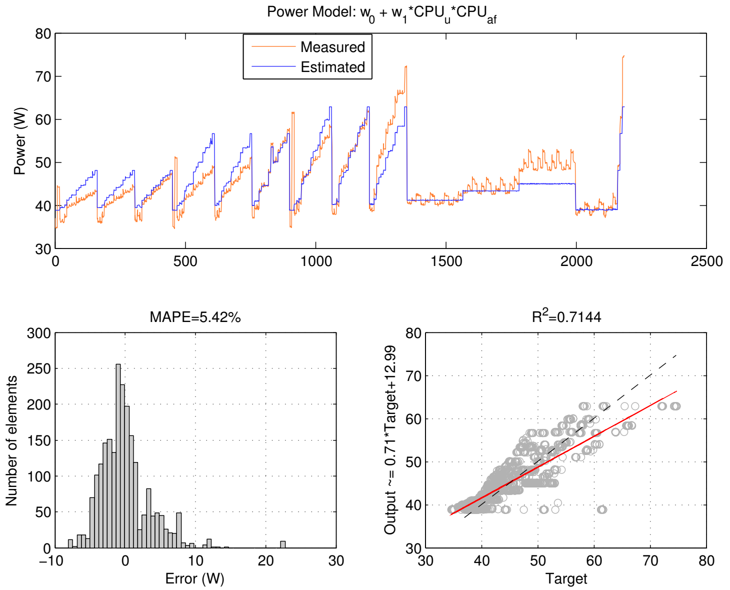
$$P \sim \%cpu * (cap * volt^2 * freq) \tag{4}$$

- Simplified version:

$$P \sim w_0 + w_1 * \%cpu * freq \tag{5}$$

- Calibration: Linear Regression

Calibrated capacitive model



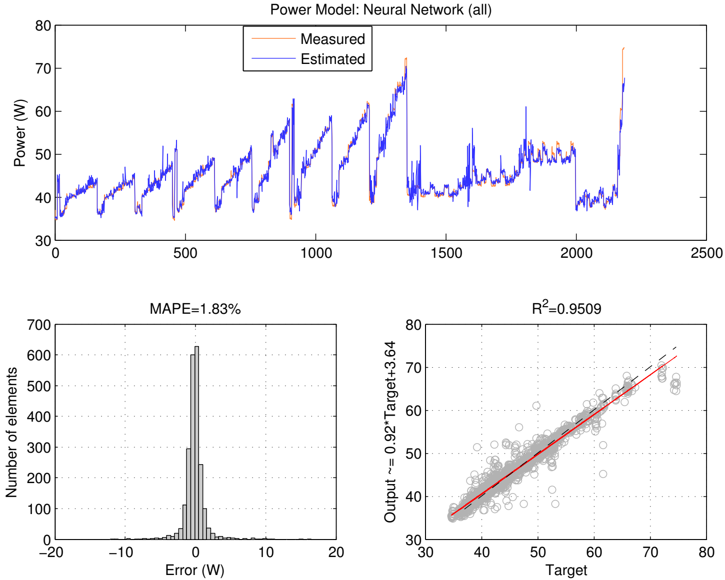
Artificial neural network model

Type	Name	Name
PC	cycles	instructions
	cache references	cache misses
	branch instructions	branch misses
	bus cycles	idle cycles frontend
	idle cycles frontend	cpu clock
	task clock	page faults
	context switches	cpu migrations
	minor faults	major faults
	alignment faults	emulation faults
	L1d loads	L1d load misses
	L1d stores	L1d store misses
	L1d prefetch misses	L1i load misses
	LLC loads	LLC load misses
	LLC stores	LLC stores misses
	L1d prefetches	LLC prefetch misses
	dTLB loads	dTLB load misses
	dTLB stores	dTLB store misses
	iTLB loads	iTLB load misses
	branch loads	branch load misses
	node loads	node load misses
	node stores	node store misses
	node prefetches	node prefetch misses
SYS	cpu usage	resident set size
	received bytes	sent bytes
MSR	cpu frequency	cpu time in C0
	cpu temperature	

ANN setup

- Topology
  Feedforward Multilayer Perceptron
  2 hidden layers: 20 5
- Training set: 70, 15, 15%
- Training algorithm: Levenberg-Marquardt

Artificial neural network model



Conclusions and perspectives

Conclusions:

- When using external power meter, energy conversion lossses need to be modeled
- CPU proportional models: calibration and accuracy
- Artificial Neural Networks seems promissing for generic power estimators

Future work:

- Reduce the number of variables
- Validation with real usecases
- Validation on distributed applications

Thank you.

fontoura@irit.fr