Power Profiling
& Energy-Aware High Performance Computing

Exa2Green
energy-aware numerics

Reza Heidari
Dr. Manuel F. Dolz

March 9th, 2016

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Power Profiling Motivation

- ✓ Tools for power and energy analysis in Parallel scientific application
- ✓ In combination with performance analysis tool
- ✓ Environment to identify sources of power inefficiency
- ✓ Using **Energy-aware techniques**: P-/C-states, DCT, etc.
- ✓ Energy savings → Reducing energy consumption while maintaining computational performance

Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

Power Profiling Steps:

- ✓ Power Measurement
- ✓ Power Tracing
- ✓ Power/Energy accountingTools

ArduPower: A new low-cost internal wattmeter

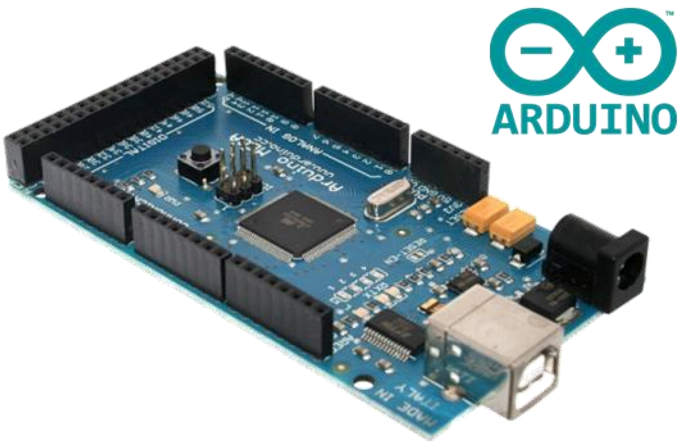
✓**Objective:** Measure internally the power consumption of computers

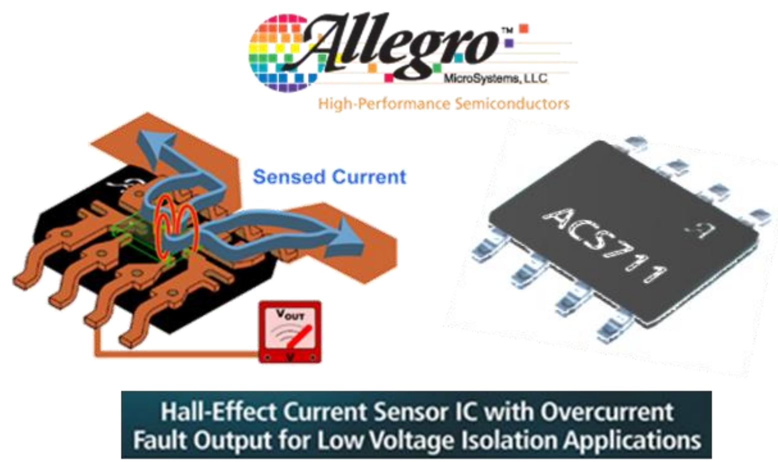
Problem: internal power meters are expensive and difficult to use (e.g., National Instruments DAS)

✓**Requirements**: Build-up a **accurate, small** and **cheap** new wattmeter device

- ✓ *Microcontroller*: Arduino Mega 2560 with 16 analog channels ~50 EUR
- ✓ *Sensors*: Allegro Hall-Effect IC sensor ACS series (accuracy ±5%) ~2 EUR/chip

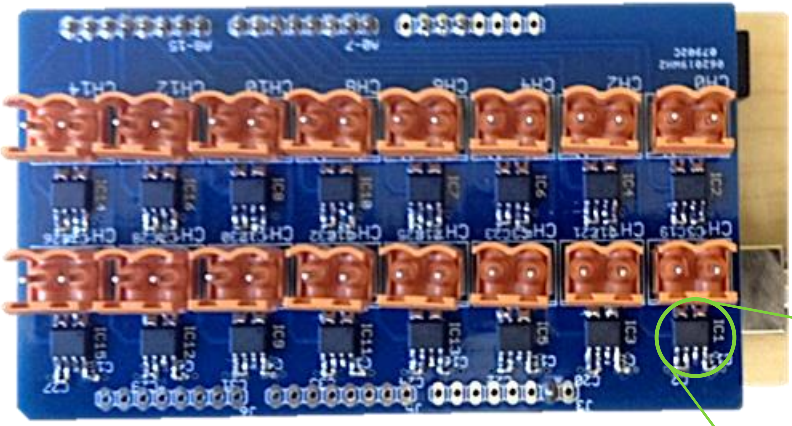
✓**Solution**: A new shield for Arduino Mega with Allegro Hall-Effect sensors!

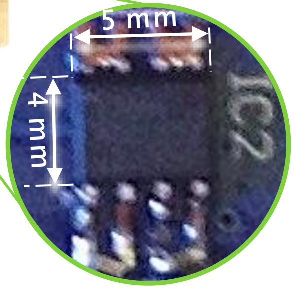


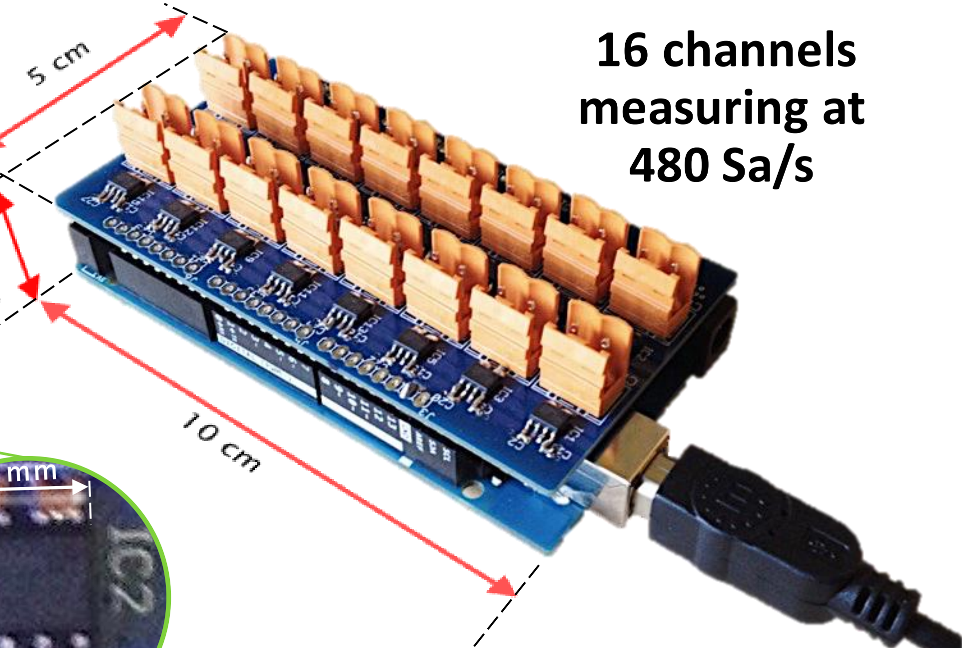


ArduPower: A new low-cost internal wattmeter

The prototype:



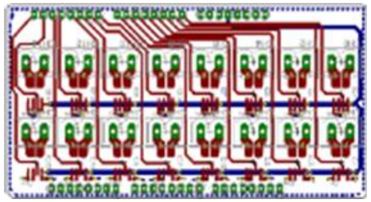




16 channels measuring at 480 Sa/s

- ✓ Final protoype of the shield:
  - ✓ ACS713 up to 20 A (DC) in (±1.5%)
- ✓ Total production cost: 100 EUR
- ✓ PCB circuits available on demand
- ✓ Integration into PMLib!

Design from G. Fabregat (UJI)

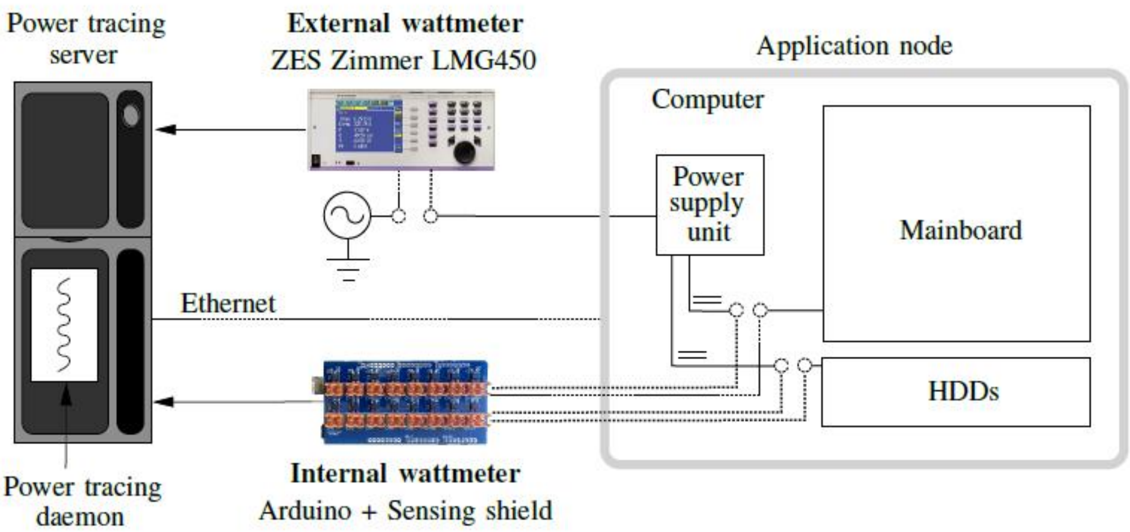


USB Serial port (but it could also work via Ethernet)

Exa2Green energy-aware numerics

ArduPower: A new low-cost internal wattmeter

✓ Installation of ArduPower on a server machine:



2x Intel Xeon X5560 (total of 8 cores) running at 2.80GHz with 12GB RAM

We collect data from the internal ArduPower and the external LMG450 devices using PMLib

Supermicro PSU 720W 80+ Gold (~82% energy efficiency)

Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

ArduPower: A new low-cost internal wattmeter

How many Samples/s?

Experiment to test the sampling rate depending on the number of lines selected from **480 Sa/s** to **5880 Sa/s**

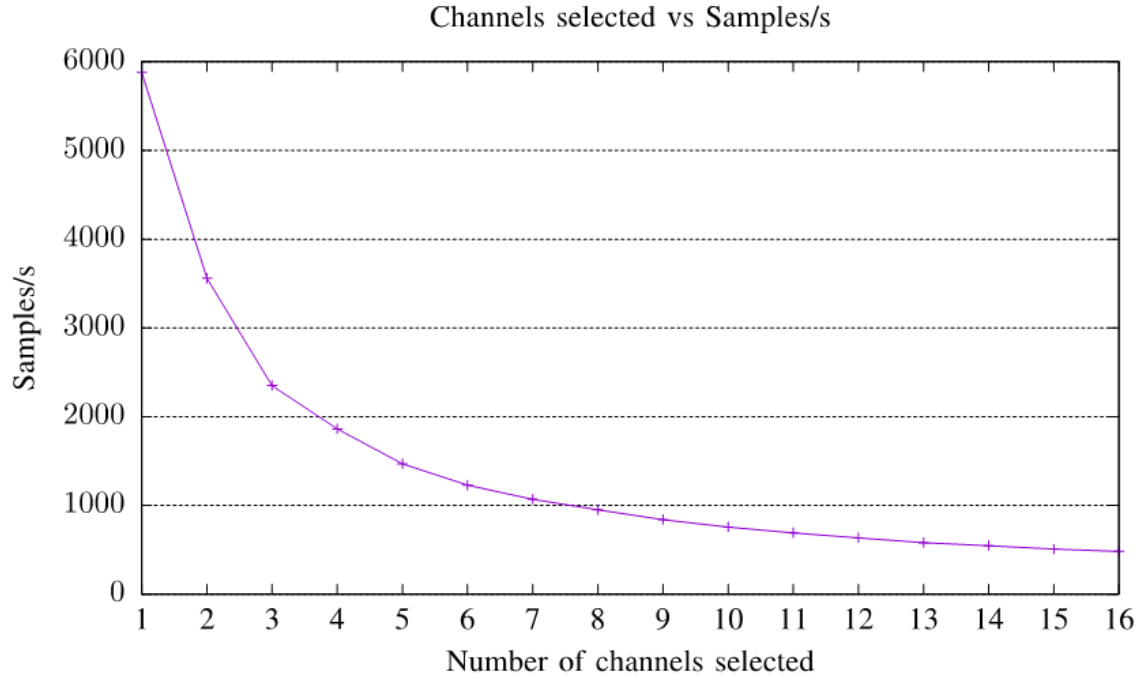


Figure 4. Samples per second with respect to the number of channels selected.

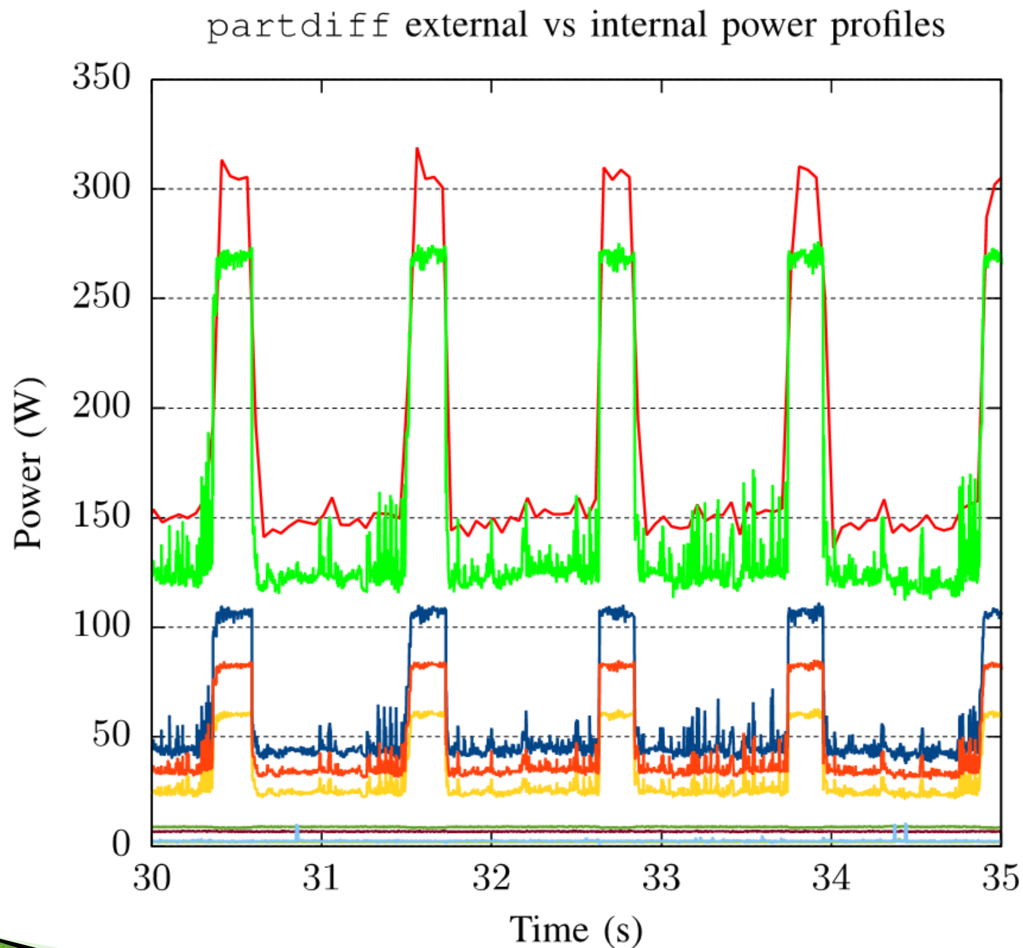
The curve is due to the communication protocol between the power tracing server and the Arduino

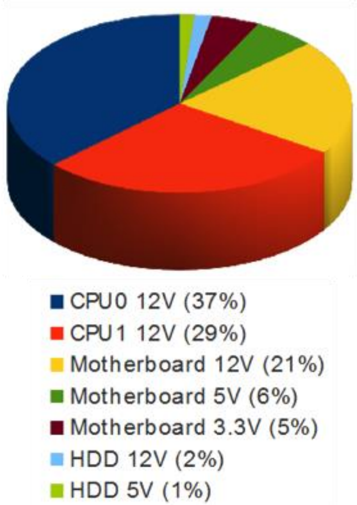
Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

ArduPower: A new low-cost internal wattmeter

✓ An example using a PDE solver: (Partial differential equation solver for Gauss-Seidel and Jacobi Method)





Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

Reduced power models based on HW/SW metrics

Motivation:

- ✓ Models can eventually avoid the use of expensive wattmeters!
- ✓ Perform energy-aware scheduling
- ✓ Increase energy consumption feed-back to the end user

Goal: build power models under the following properties:

- ✓ **Accurate**: they should be precise
- ✓ **Simple**: the prediction should be computed fast, at real time
- ✓ **Inexpensive**: framework/devices should not be expensive
- ✓ **Portable**: should work on many platforms/architectures

Reduced power models based on HW/SW metrics

Lots of works can be found in modelling power, however they always take a set of pre-selected metrics/counters, which might not be a good idea!

We present a methodology in order to select the best HW/SW metrics in order to build reduced and simple linear power models

Linear model:

$$P_T(t) \approx c + \sum_{i=1}^{n} f_i(t) \cdot c_i$$

Reduced model:

$$\tilde{P}_T(t) \approx c + \sum_{i=1}^{r} f_i(t) \cdot c_i'$$

- ✓ *n* : total number of metrics
- ✓ *r* : reduced number of metrics
- ✓ *c* : constant contribution to power (system and static)
- ✓ f_i : value of the i–th metric
- ✓ c_i : value of the i–th weight

Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

Reduced power models based on HW/SW metrics

Metric-filtering algorithm (1st step):

1. We obtain the correlation matrix $n \times n$ for the n metrics and set a threshold $h = h_0$
2. We apply k-means and create r clusters, all the clusters must have a correlation threshold higher than h.
3. When a cluster has a threshold higher than h, we found a relevant group. The metric with highest sum of correlation is the representative and stored separately. We go to step 2. If no more clusters can be found we go to step 4.
4. If $h > 0.5$, we reduce h to $h = h \cdot h_0$ in order to capture the least related features and go to step 2. Not captured metrics are also considered as representatives.

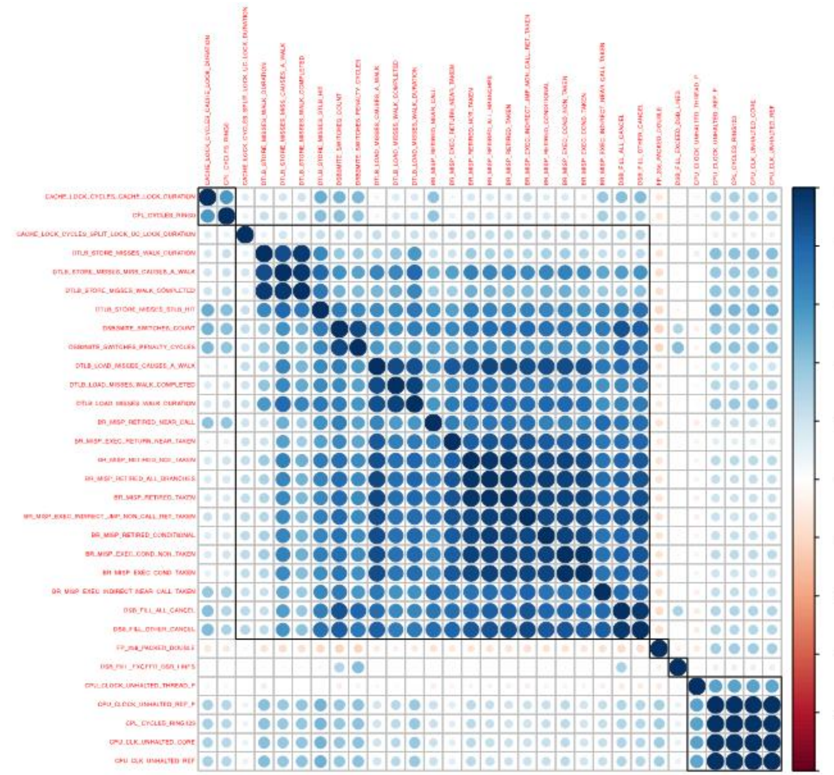


Fig. 1: Partial correlation matrix obtained from the training set. Blue and red circles stand for positive and negative correlations, respectively.

Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

Reduced power models based on HW/SW metrics

Metric-filtering algorithm (2nd step):

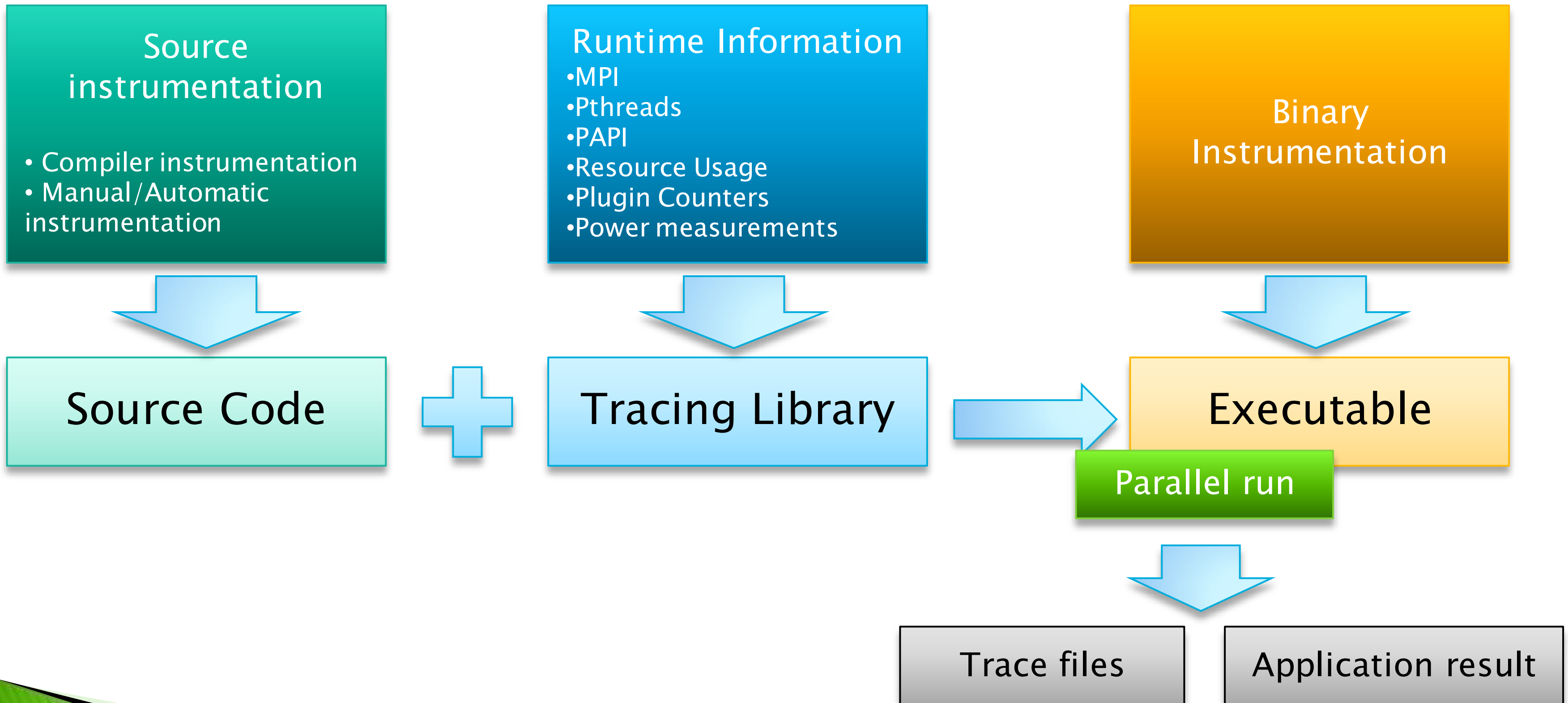
- ✓ The obtained representatives from the 1st step are stored by correlated with respect to the measured power consumption.
- ✓ To reduce the number of combinations and derive reduced power models, we take only the first *f* metrics with highest correlation to power consumption.
- ✓ *r* representative metrics are now combined taking into account metrics that can be read at once:
  - ✓ 4 PMC registers
  - ✓ 3 FIXC registers
  - ✓ RAPL registers
  - ✓ OS stats and temperature

#g	Corr.	Group representant	Other metrics
1	0.970	PWR_PKG_ENERGY (PKG_ENERGY)	SENSORS_PHY_ID_0, PWR_PP0_ENERGY, SENSORS_CPU
2	0.906	CPU_CLK_UNHALTED_CORE (UNHC)	CPL_CYCLES_RING123, CPU_CLK_UNHALTED_CORE, CPU_CLK_UNHALTED_REF, CPU_CLOCK_UNHALTED_REF_P
3	0.881	L1D_BLOCKS_BANK_-CONFLICT_CYCLES (L1CC)	L1D_BLOCKS_BANK_CONFLICT_CYCLES, L2_TRANS_ALL_REQUESTS
4	0.836	L2_RQSTS_MISS (L2RM)	L2_TRANS_DEMAND_DATA_RD
5	0.832	L2_RQSTS_ALL_PF (L2RQ)	L2_TRANS_ALL_PREF
6	0.826	HW_PRE_REQ_DL1_MISS (DL1M)	HW_PRE_REQ_DL1_MISS, L1D_REPLACEMENT
7	0.812	L2_LINES_OUT_DEMAND_CLEAN (L2DC)	OFFCORE_REQUESTS_DEMAND_DATA_RD
8	0.810	UOPS_DISPATCHED_CORE (UOPSD)	MEM_UOP_RETIRED_LOADS, UOPS_DISPATCHED_PORT_PORT_2_LD, UOPS_DISPATCHED_PORT_PORT_3_LD, UOPS_DISPATCHED_THREAD, UOPS_RETIRED_ALL
9	0.801	INSTR_RETIRED_ANY (INRA)	INST_RETIRED_PREC_DIST, INST_RETIRED_ANY_P, MEMLOAD_UOPS_RETIRED_L1_HIT, UOPS_ISSUED_ANY, UOPS_RETIRED_RETIRE_SLOTS
10	0.781	L2_LINES_IN_E (L2LI)	L2_LINES_IN_ALL, L2_RQSTS_PF_MISS, L2_TRANS_L2_FILL, OFFCORE_REQUESTS_ALL_DATA_RD
11	0.773	UOPS_DISPATCHED_PORT_PORT_0 (UOPS0)	UOPS_DISPATCHED_PORT_PORT_1
12	0.762	RESOURCE_STALLS_RS (RSRS)	CPU_UTIL
13	0.753	UOPS_DISPATCHED_PORT_PORT_2 (UOPS2)	UOPS_DISPATCHED_PORT_PORT_3
14	0.744	L2_RQSTS_ALL_DEMAND_DATA_RD (L2DD)	L2_RQSTS_ALL_DEM_AND_DATA_RD_HIT
15	0.675	INT_MISC_STALL_CYCLES (INTS)	RESOURCE_STALLS_ANY

Table 1: List of the 15 most correlated representants in respect to power with $h = 0.95$ for the correlation threshold between clusters.

Why profiles & traces?

- ✓ Details and variability are important (along time, processors, etc.)
- ✓ Extremely useful to analyze performance of applications, also at power level!

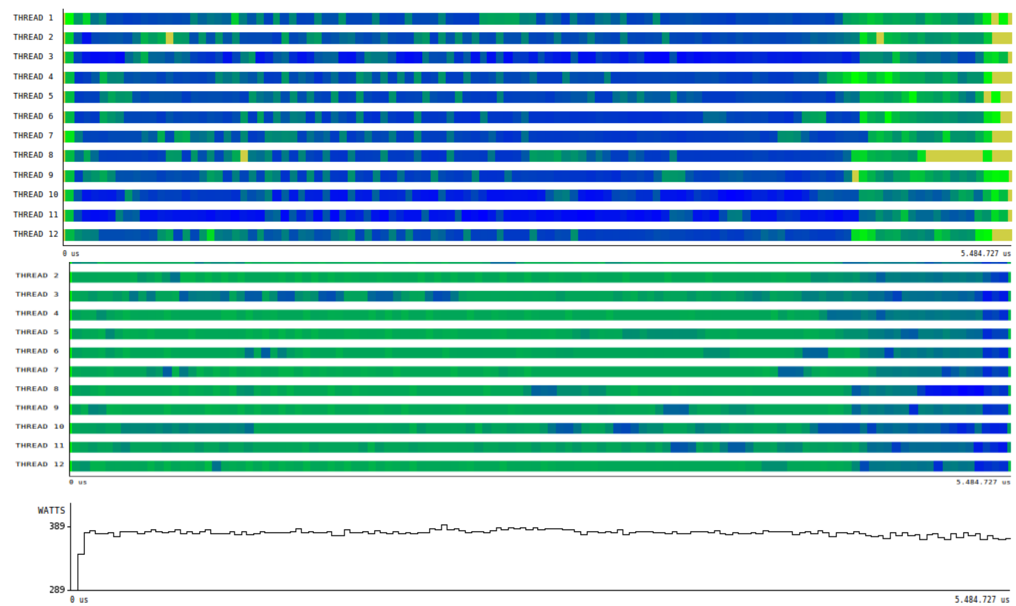


Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

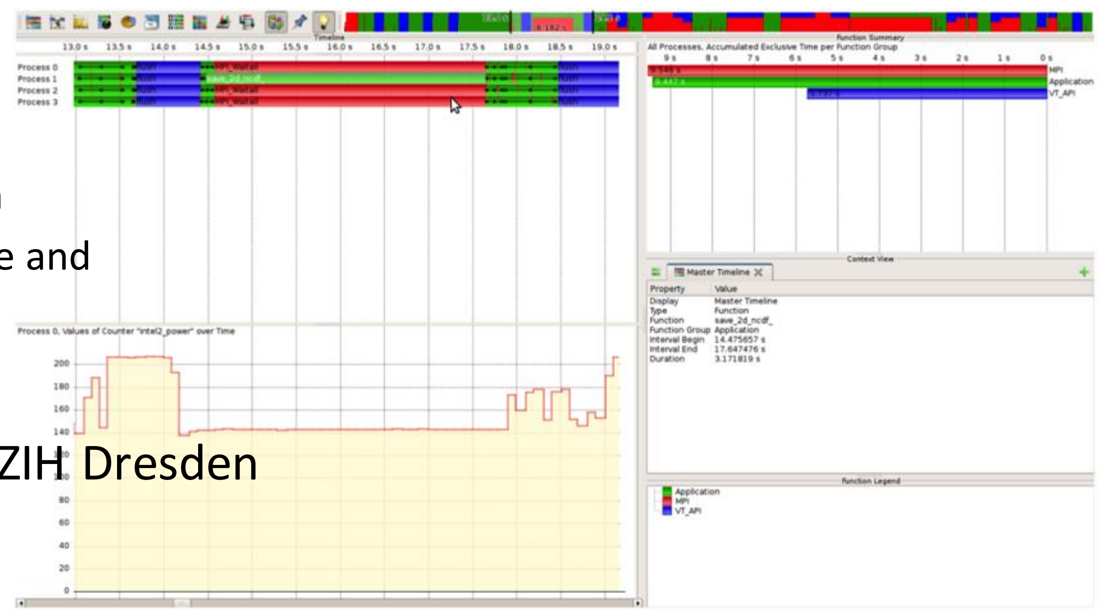
Paraver + Extrae

- ✓ Extrae
  - ✓Interception of MPI, OpenMP, P-thread calls
  - ✓Recording of relevant information: events, hardware counters, etc.
  - ✓Good integration with power profile traces
- ✓ Paraver
  - ✓Visualization tool
  - ✓High number of metrics to characterize the program and Performance application
- ✓ License: GPL at BSC
- ✓ Format .prv



Vampir + VampirTrace

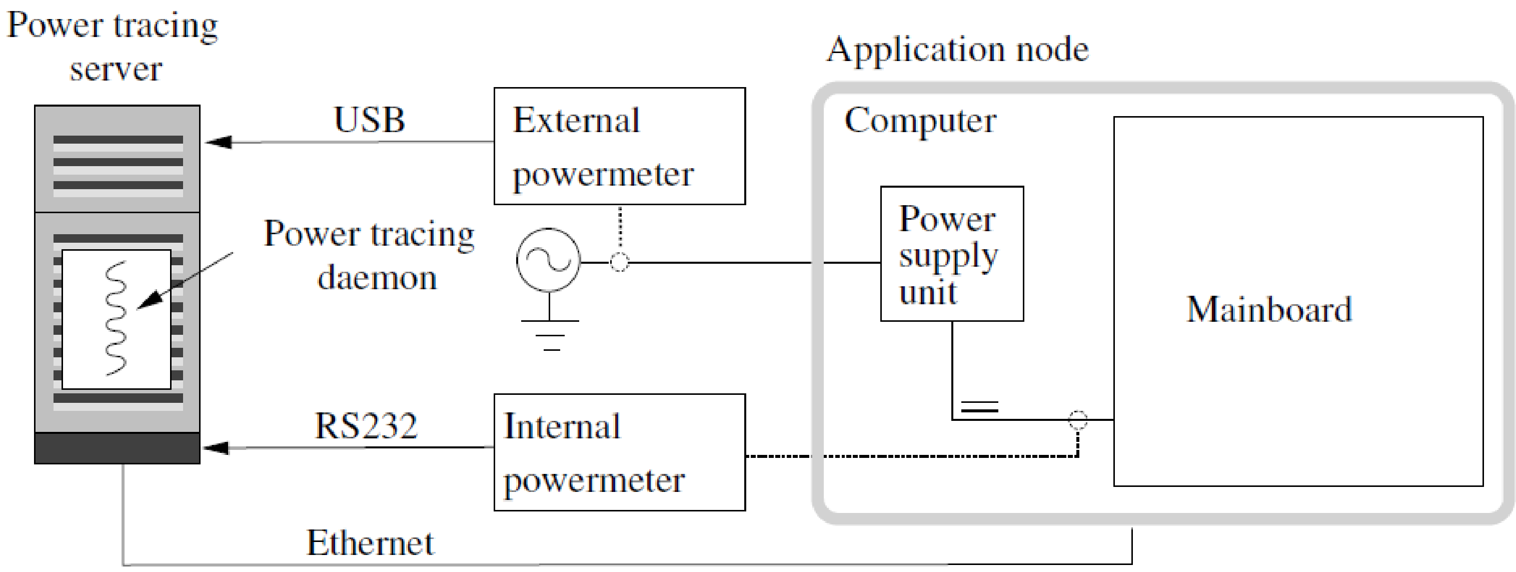
- ✓VampirTrace for instrumentation and event trace recording
- ✓Vampir for event trace visualization
  - ✓OTF
  - ✓Interface analysis
  - ✓Parallel evaluation
  - ✓Distributed trace data evaluation
  - ✓Easy combination of performance and power traces



- ✓ License: BSD Open Source at ZIH Dresden
- ✓ Format: .otf

PMLib: The power measurement library

- ✓Interface to interact and use wattmeters
  - ✓Also power hardware counters: Intel RAPL, P-/C-States, etc.
  - ✓Resource utilization metrics: CPU, Memory, Network, Disk dev. utilization, etc.



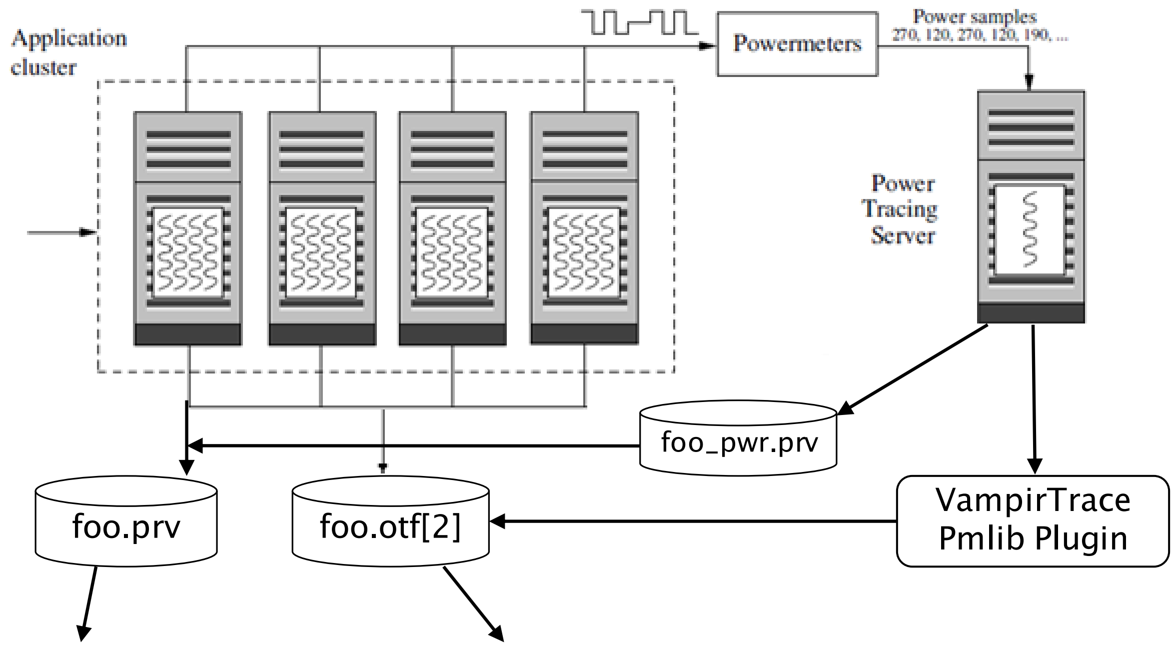
- ✓**Server daemon**: collects data from power meters and send to clients
- ✓**Client library**: enables communication with server and synchronizes with start-stop primitives

Exa2Green energy-aware numerics
SEVENTH FRAMEWORK PROGRAMME

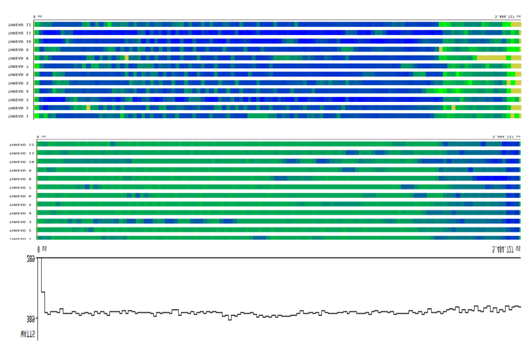
Power-performance Analysis Framework

Instrumented application:

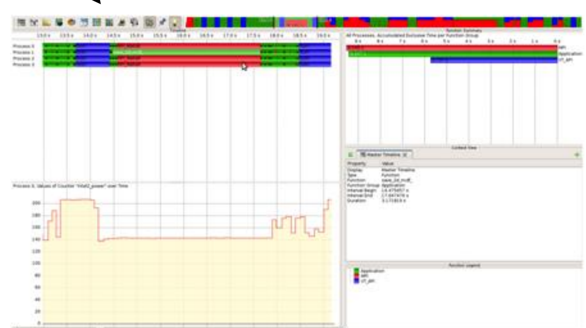
- VampirTrace:
  - ✓ Automatic, Manual, Source and Binary instrumentations
  - ✓ Format: .otf
- Score-P:
  - ✓ Same instrumentation modes as VampirTrace
  - ✓ Format: .otf2 by default
- Extrae:
  - ✓ Automatic/Manual instrumentation
  - ✓ Format: .prv



Two Visualization tools:



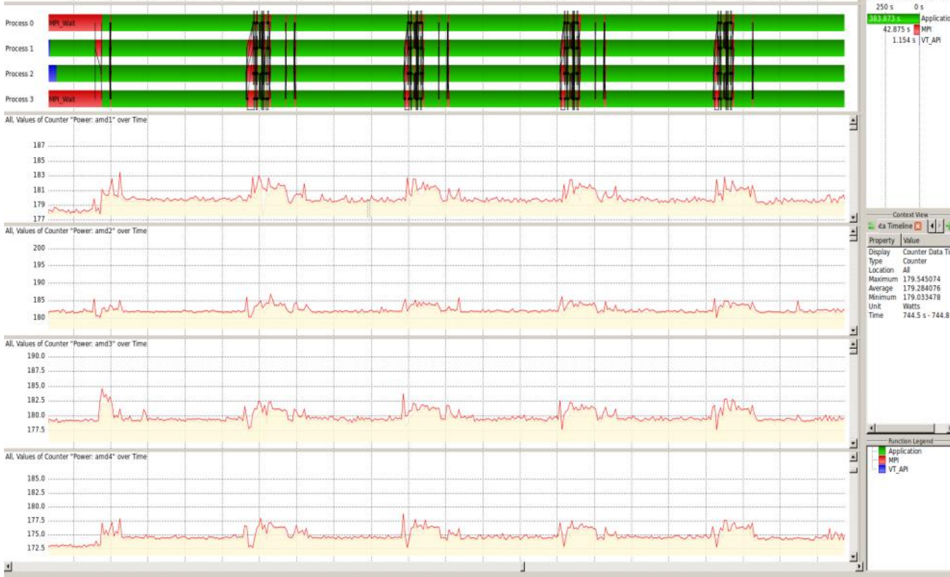
Visualization with Paraver



Visualization with Vampir

Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME



ECMWF OpenIFS

- Vampir + VampirTrace

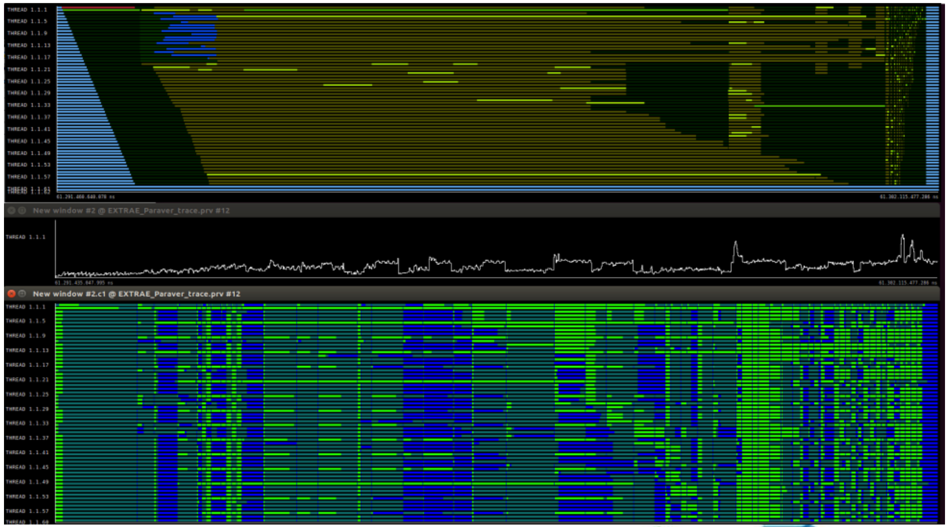
✓OpenIFS forecast model from ECMWF

running on 4 nodes (tasks & power)

- Paraver + Extrae

✓Libflame's Cholesky factorization

running on Xeon Phi (tasks, power, C-states)



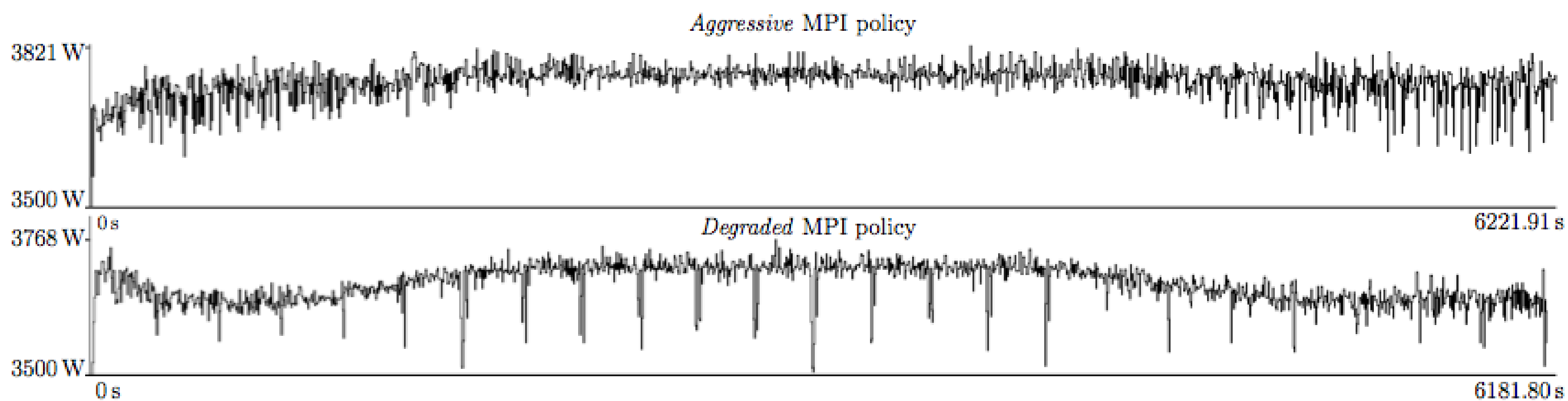
Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

Performance/Energy-efficiency of COSMO-ART

✓ Experimental results:

Power profile of a 24-hour simulation of COSMO-ART with Aggressive and Degraded MPI policies



✓ Systematic power drops each simulated hour with degraded mod

The cores were utilized (load) only ≈50%!

Reduction of in the total power/energy consumption of 2%

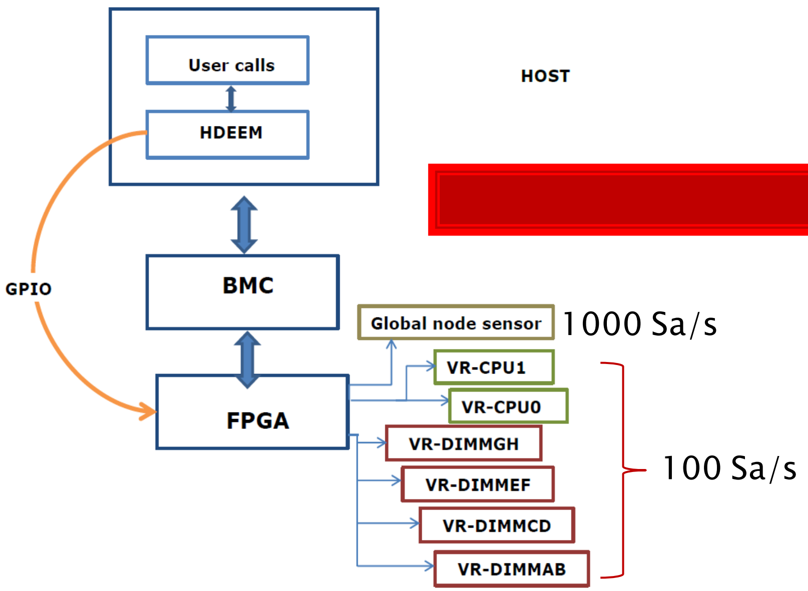
Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

- Add-on to the resource manager to provide energy efficiency feedback

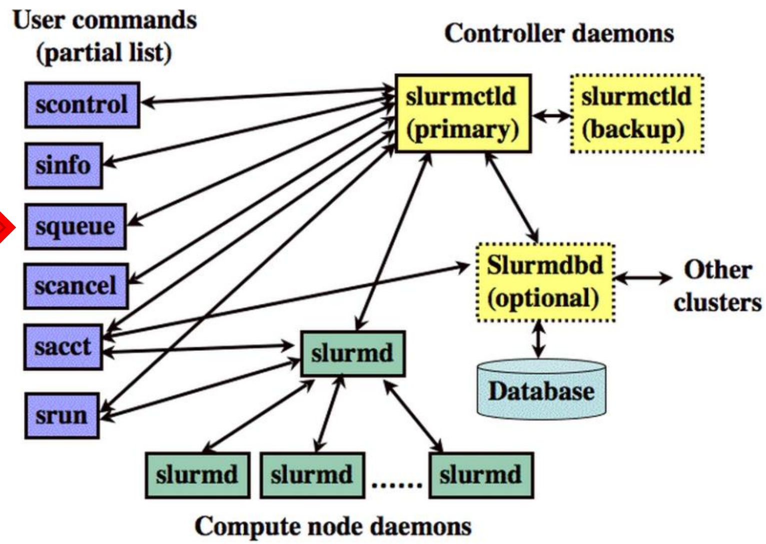
HDEEM Plugin

(High Definition Energy Efficiency Monitoring)



SLURM

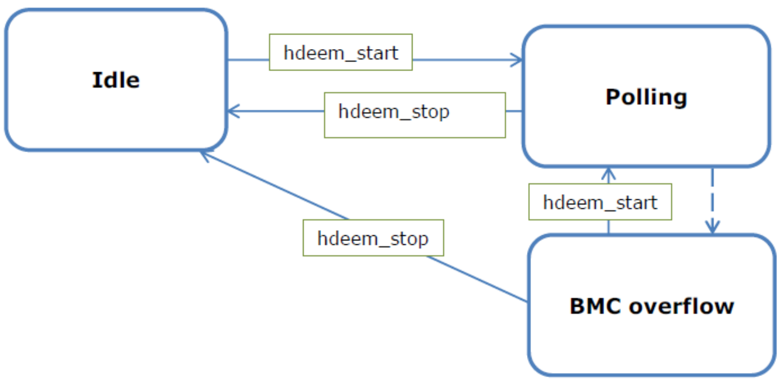
(Simple Linux Utility for Resource Management)

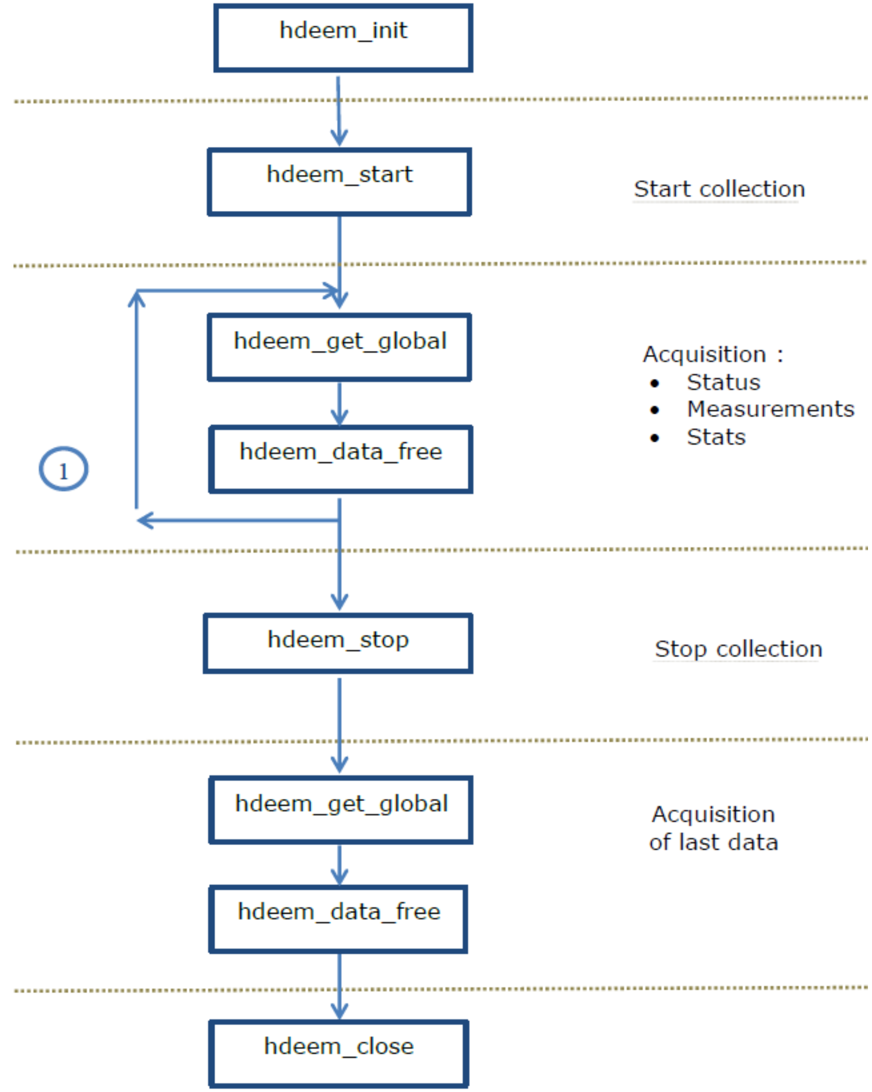


How HDEEM works?

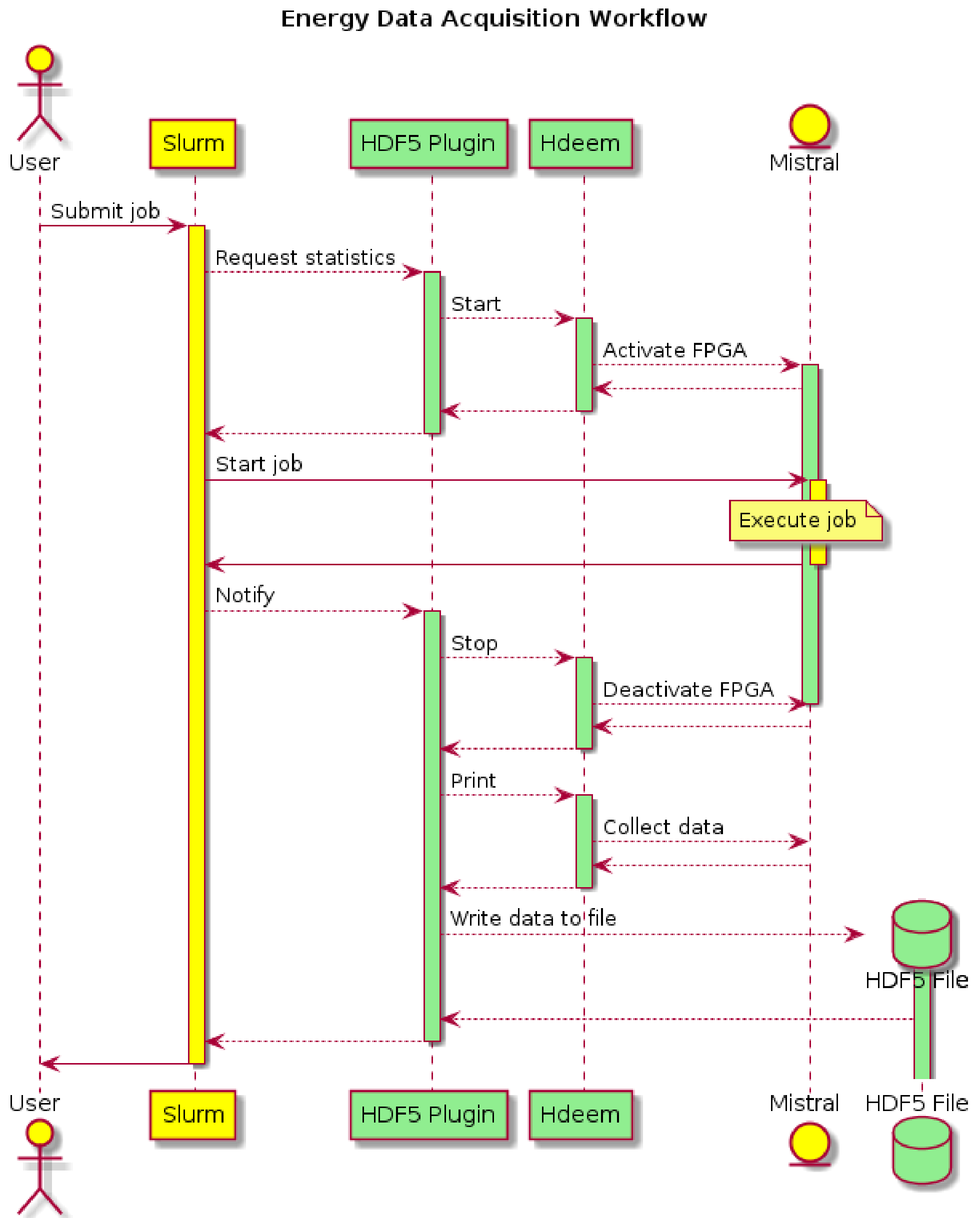
- **HDEEM Command lines**
  1. startHDEEM
  2. stopHDEEM
  3. checkHDEEM

- **HDEEM C API**





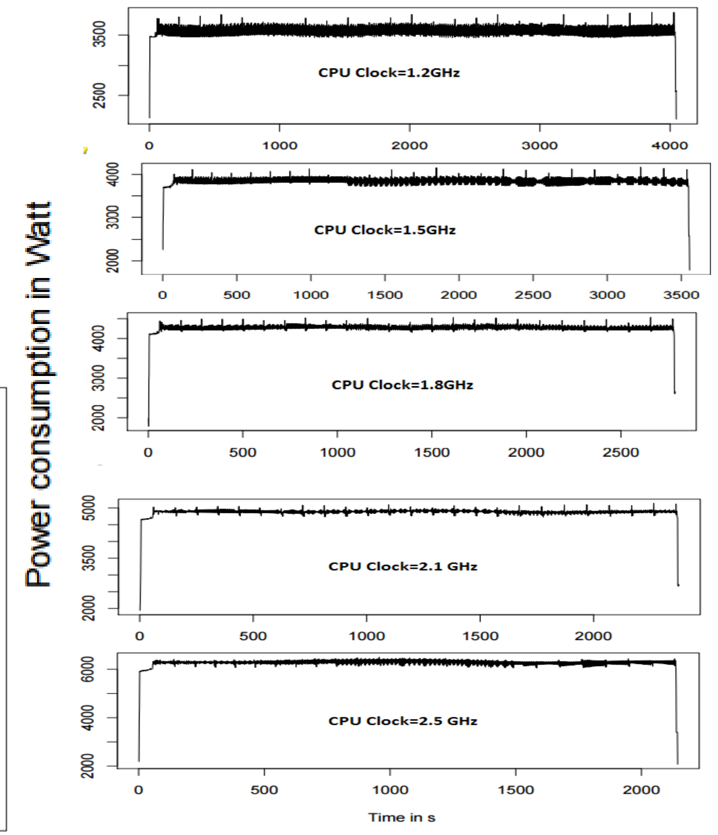
Exa2Green energy-aware numerics


Energy Data Acquisition Workflow
User
Slurm
HDF5 Plugin
Hdeem
Mistral
Submit job
Request statistics
Start
Activate FPGA
Start job
Execute job
Notify
Stop
Deactivate FPGA
Print
Collect data
Write data to file
HDF5 File

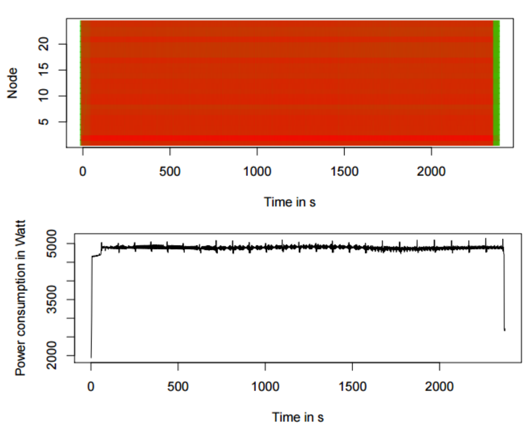
➢ Using HDEEM to investigate COSMO-ART Energy Efficiency

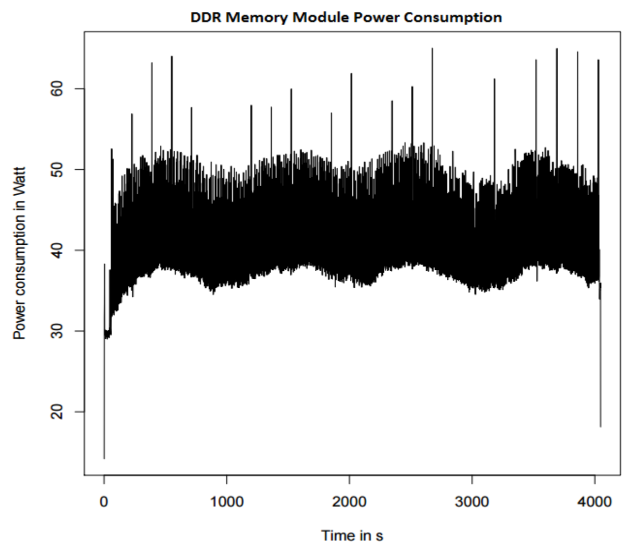
- Running COSMO-ART on different CPU frequency 2.5
- 1.2 GHz: 3988.9 Wh in 4046 s
  1.5 GHz: 3791.1 Wh in 3555 s
  1.8 GHz: 3299.6 Wh in 2786 s
  2.1 GHz: 3212.0 Wh in 2374 s
  2.5 GHz: 3734.4 Wh in 2143 s

Power Consmuption Traces of COSMO-ART
at 5 different frequencies



24 Nodes contribution traces
To power consumption





Exa2Green
energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

- Reads and visualizes Hdf5 & HDEEM power traces
- Interactively explores power readings via a dynamic HTML interface

Exa2Green energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME

Conclusions

Power measurement devices:

- ✓ ArduPower is a promising internal wattmeter by being simple, small, accurate and cheap
- ✓ Reduce Power Models → Less overhead, scalable and cheap

Power-performance analysis framework:

- ✓ Useful to detect power bottlenecks in the code → reduce the energy consumption
- ✓ Performance inefficiency → hot spots in hardware and power bottlenecks in code!

Energy Accounting:

- ✓ A new energy/power profiling based on HDEEM technology
- ✓ High Spatial granularity
- ✓ High Temporal granularity

Thank you for you attention!

Questions?

Exa2Green
energy-aware numerics

SEVENTH FRAMEWORK PROGRAMME