Tools for Energy-Efficient HPC

January 20, 2014 | Michael Knobloch

Energy-Efficiency Tools Projects at JSC

The Past

The Future

Score-E

The Present

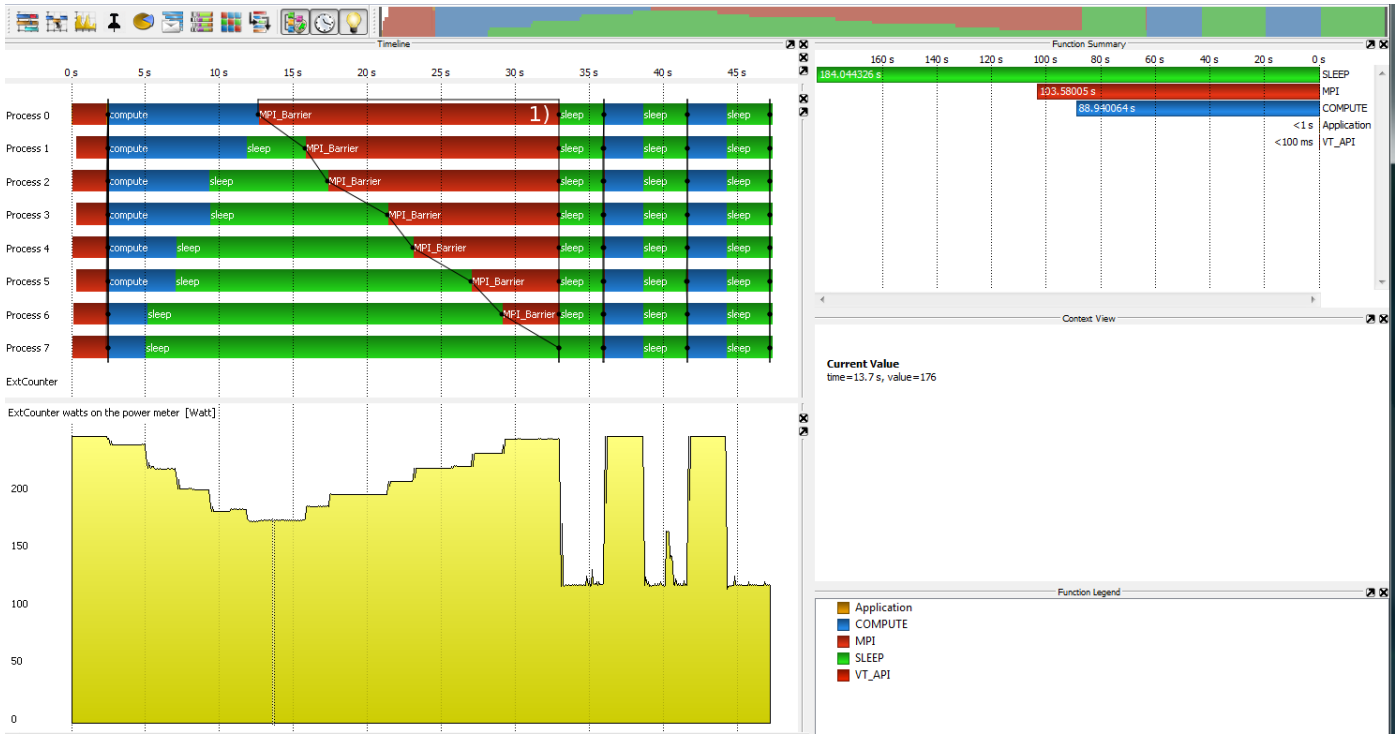
Exa Scale
Innovation Center

The Past – 2009 - 2012

eeClust - Energy-Efficient Cluster Computing

- Project partners: Uni Hamburg, TU Dresden, JSC, ParTec
- www.eeclust.de
- Goals: Identify phases of low resource utilization and turn hardware to lower power-states in such phases
- Integral point: Extension of performance analysis tools to analyse application power consumption and hardware utilization

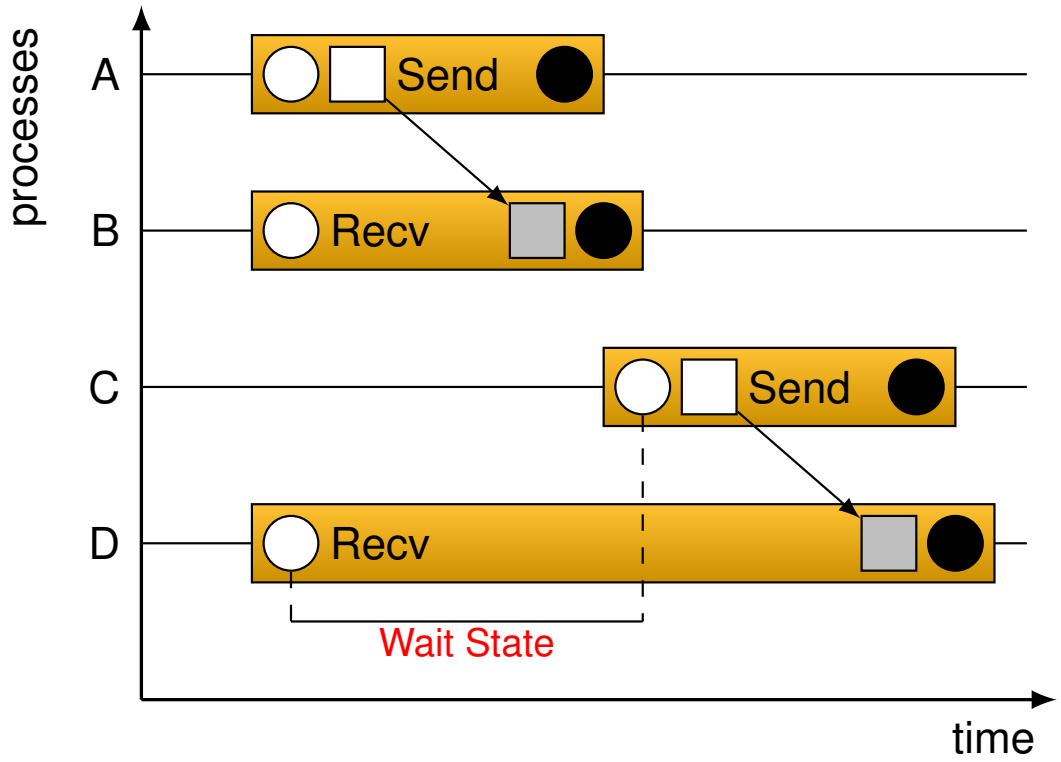
MPI Busy-Waiting



MPI Busy-Waiting

Power consumption in phases of busy-waiting is very high due to constant CPU activity.

Scalasca Wait-State Detection



Calculating Energy-Saving Potential

Idle-Waiting

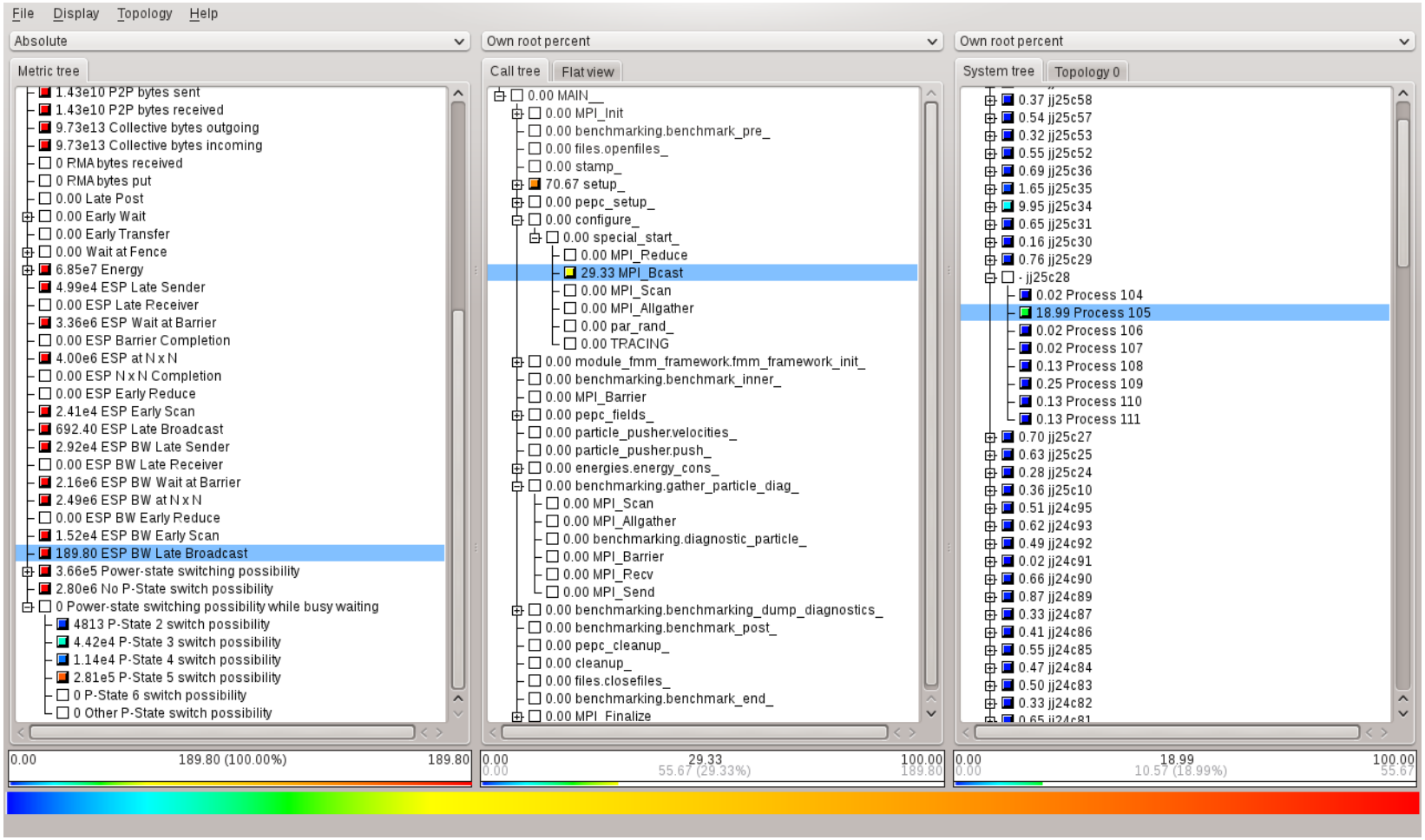
$$ESP = \max_{p \in PS}((t_w * A_{p_1}) - (t_w - t_{T_{p,p_1}}) * I_p + E_{T_{p,p_1}})$$

Busy-Waiting

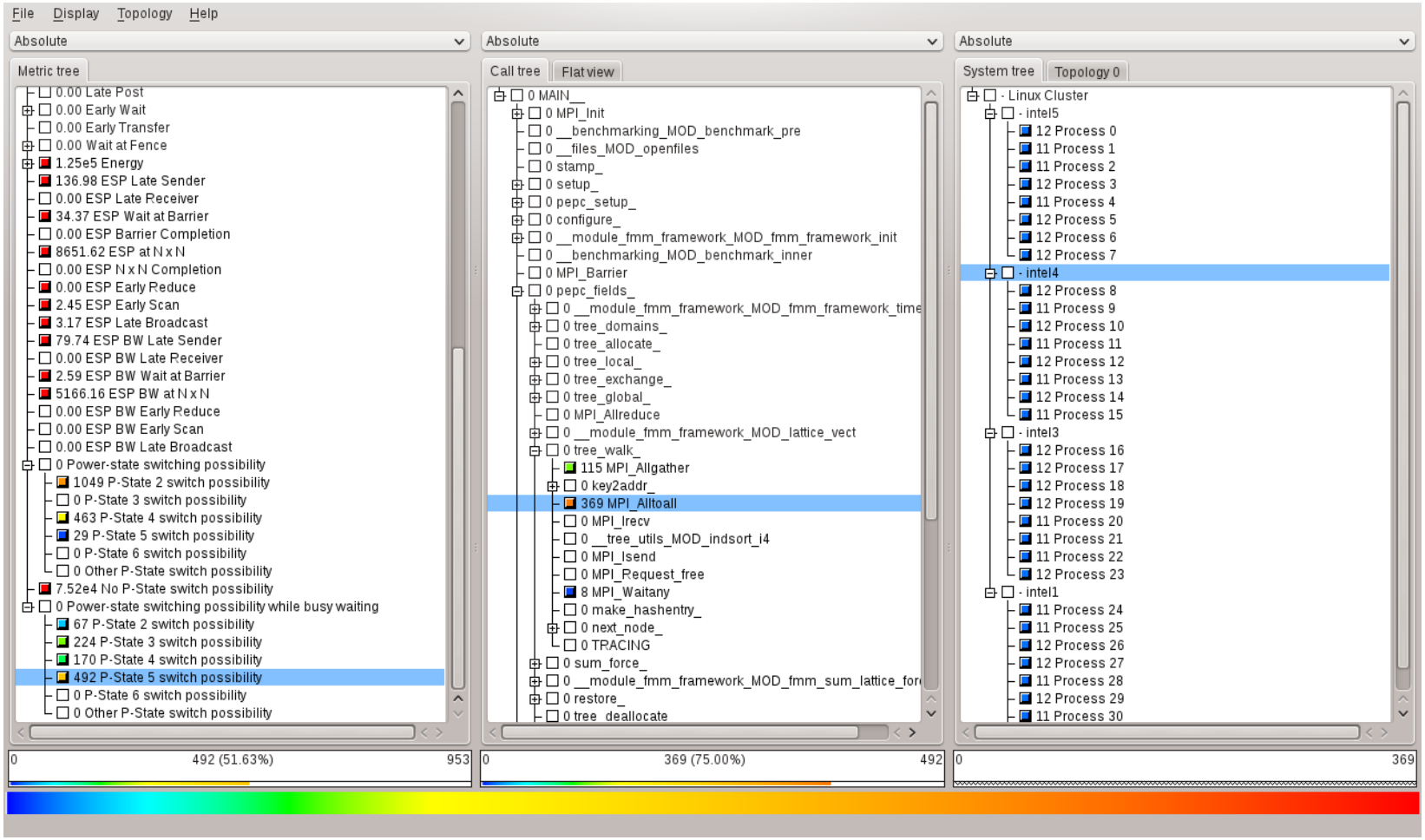
$$ESP\_BW = \max_{p \in PS}((t_w * A_{p_1}) - (t_w - t_{T_{p,p_1}}) * A_p + E_{T_{p,p_1}})$$

PS – Set of power states
A_p – Active energy in P-State p
I_p – Idle energy in P-State p
t_w – Waiting time
$t_{T_{p_1,p_2}}$ – Transition time
$E_{T_{p,p_1}}$ – Transition energy

Example: Energy Saving Potential



Example: Optimal P-State Detection



The Present – 2010 - . . .

EIC - Exascale Innovation Center

- Project partners: IBM Germany R&D and JSC
- Goal: Co-Design for next-gen of Supercomputers
- One work-package on energy-efficiency
- Investigation of power consumption on Blue Gene
- Fine-grained power measurements on POWER7

Power Consumption Analysis on POWER7

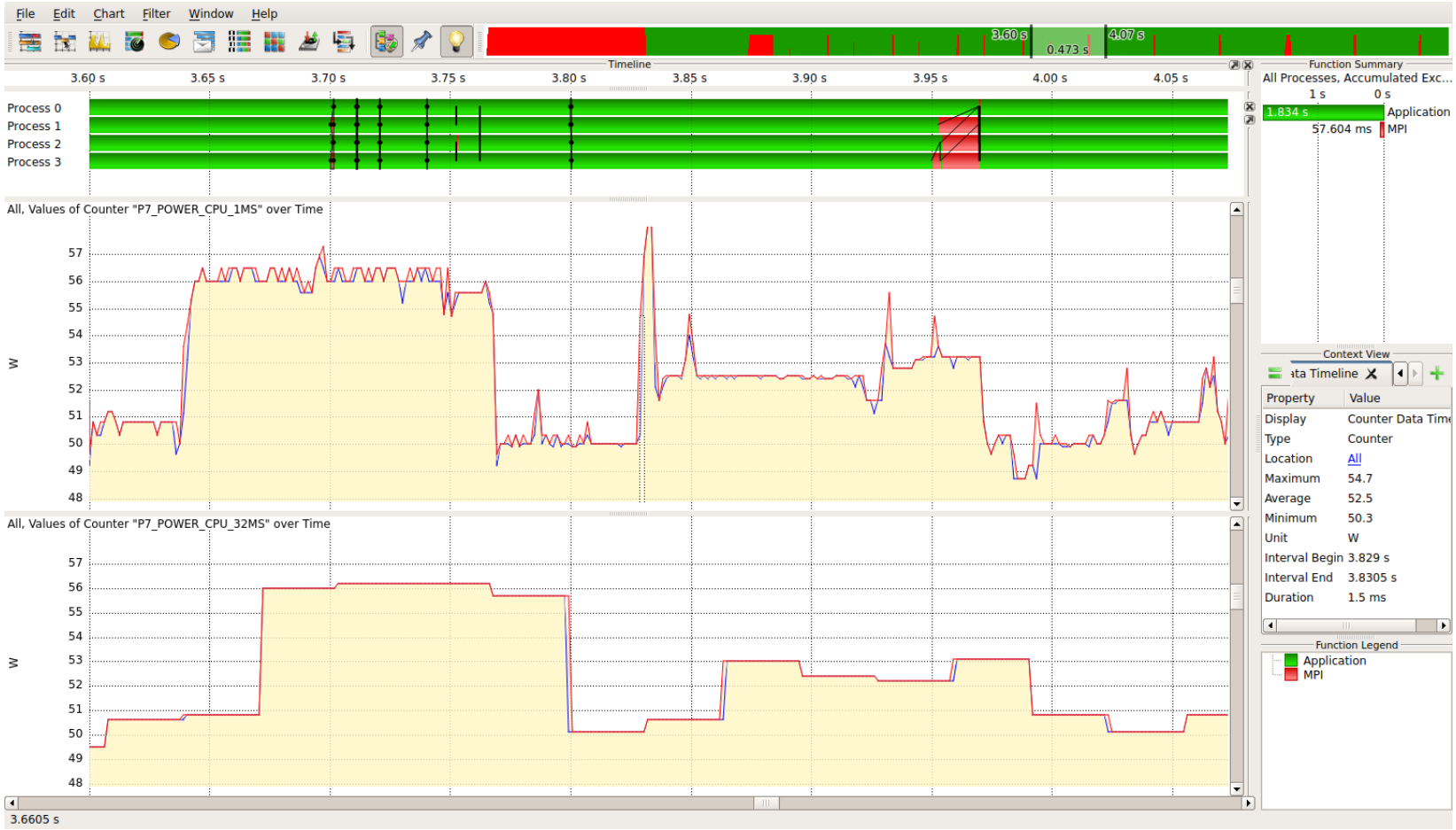
Amester

IBM Automated Measurement of Systems for Temperature and Energy Reporting software.
Results were published at EnA-HPC 2013.

Sensor name	Units	Time scale	Description
PWR1MS	W	Instantaneous	Node power consumption
PWR1MSP0	W	Instantaneous	Processor power consumption
PWR1MSMEM0	W	Instantaneous	Memory power consumption
PWR32MS	W	avg. over last 32 ms	Node power consumption
PWR32MSP0	W	avg. over last 32 ms	Pocessor power consumption
PWR32MSMEM0	W	avg. over last 32 ms	Memory power consumption
IPS32MS	Mips	Every 32 ms	Instructions per second rate

Example: Component Level Power Measurement

Example: Counter Resolution Comparison



The Future – 2013 - 2016

Score-E

- Main Tools Partners: JSC, TU Dresden, TU Munich
- Successor of SILC and LMAC
- Extension of Score-P measurement system (www.score-p.org)
  - Common measurement system for Scalasca, Vampir, and Periscope
- Power and Energy measurements from different sources, e.g. RAPL, Xeon Phi power sensors, etc.
- Energy modelling from power consumption data
- Enable auto-tuning for energy efficiency
- New visualization
- Test on real-world applications

Profiling for Energy

Profiling

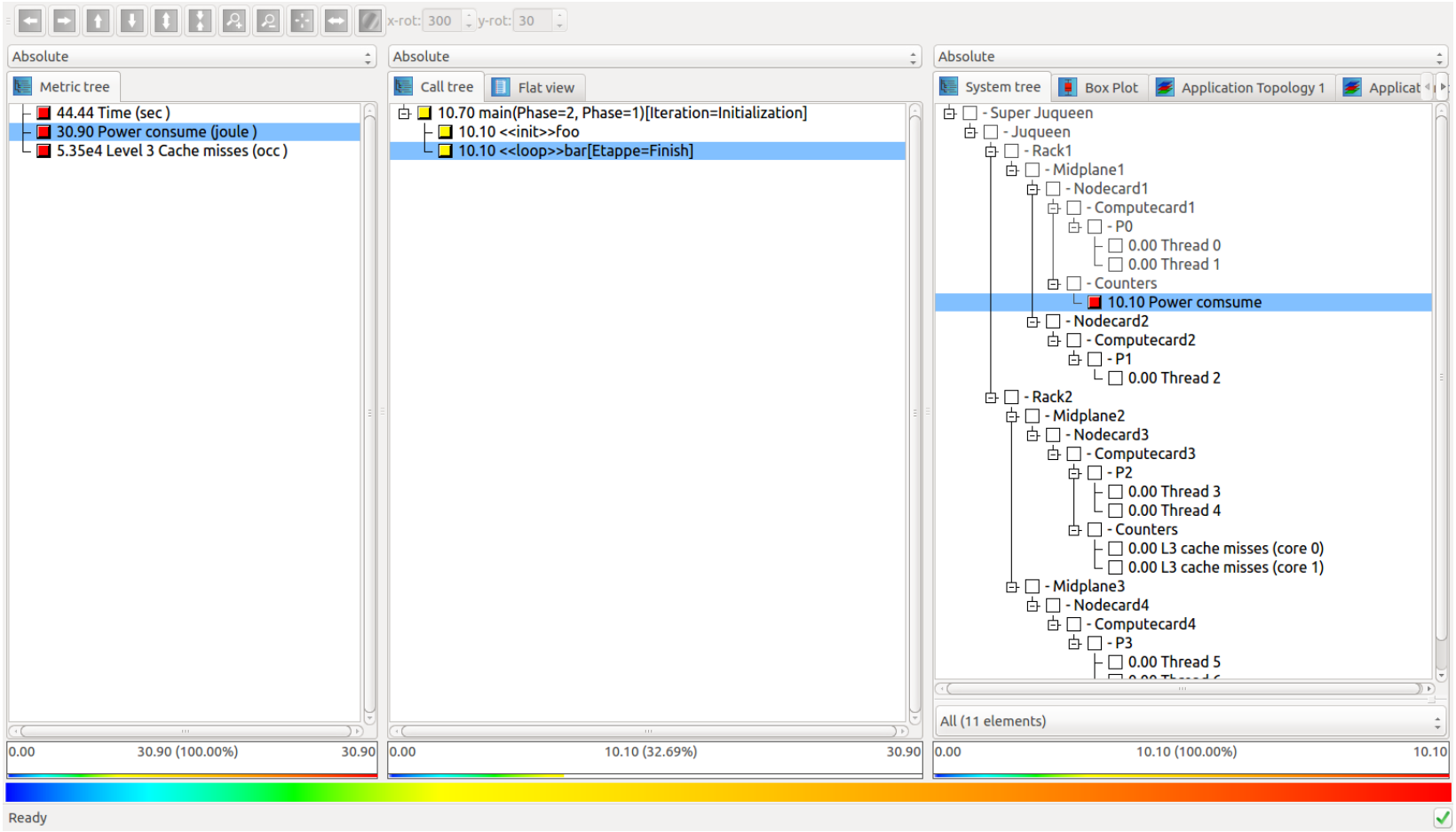
- Instrumentation-based or sampling
- Aggregation of event data at runtime
- Instrumentation-based profilers: Requires energy data instead of power data
- Ideal situation: Power integration done in hardware $\Rightarrow$ Energy-counters
- Possible alternative: Do counter sampling between two measurement points, power integration done in software
  - Overhead problem?
  - Sampling frequency?

Node-Level Counters

Shared Counters

- Can be shared by multiple processes at different scopes:
  - Last Level Cache: Shared across all processes on a socket
  - Network counters: Shared across all processes on a node
  - BG/Q Power consumption: on a node-board level
- Some can be queried by all processes, some only by one
- Doesn’t make sense in current metric scheme
- How to display them in Cube?
  - Flexible system tree
  - ~~Separate cube file for each scope~~

Flexible System Tree



To-Do List

What's there

- Infrastructure in Score-P
- Support for different counter types (traditional, sampling, etc.)
- Support for tupels in Cube (min, max, avg, etc.)
- Cube's flexible system tree

What's missing

- Way to handle not available data
- Way to record counters with low overhead
  - Collective user-instrumentation call?

Collective User-Instrumentation Call

Requirements

- Scope should be as small as possible
- Scope should be adjustable
- Scopes are the same as in flexible system tree
- But: Might not work with some applications (Master-Worker)

Examples

- SCOREP_USER_{SCOPE}_COLLECTIVE_REGION_INIT
- SCOREP_USER_COLLECTIVE_{SCOPE}_REGION_INIT
- SCOREP_USER_COLLECTIVE_REGION_INIT(SCOPE)

Lessons learned

Metric discussion

- Metric to define energy-efficiency unclear
- Power vs. Energy
- Might require different analyses

Tools need

- Sensors that provide relevant information
  - Power, energy, temperature, etc.
- At all relevant system levels
- Scalable APIs