

Building an energy dashboard

Energy measurement and visualization in current HPC systems

Thomas Geenen



SURFsara

The Dutch national HPC center

- 2H 2014 > 1PFlop
- GPGPU accelerators
- Grid
- HPC Cloud
- Hadoop
- Data Services

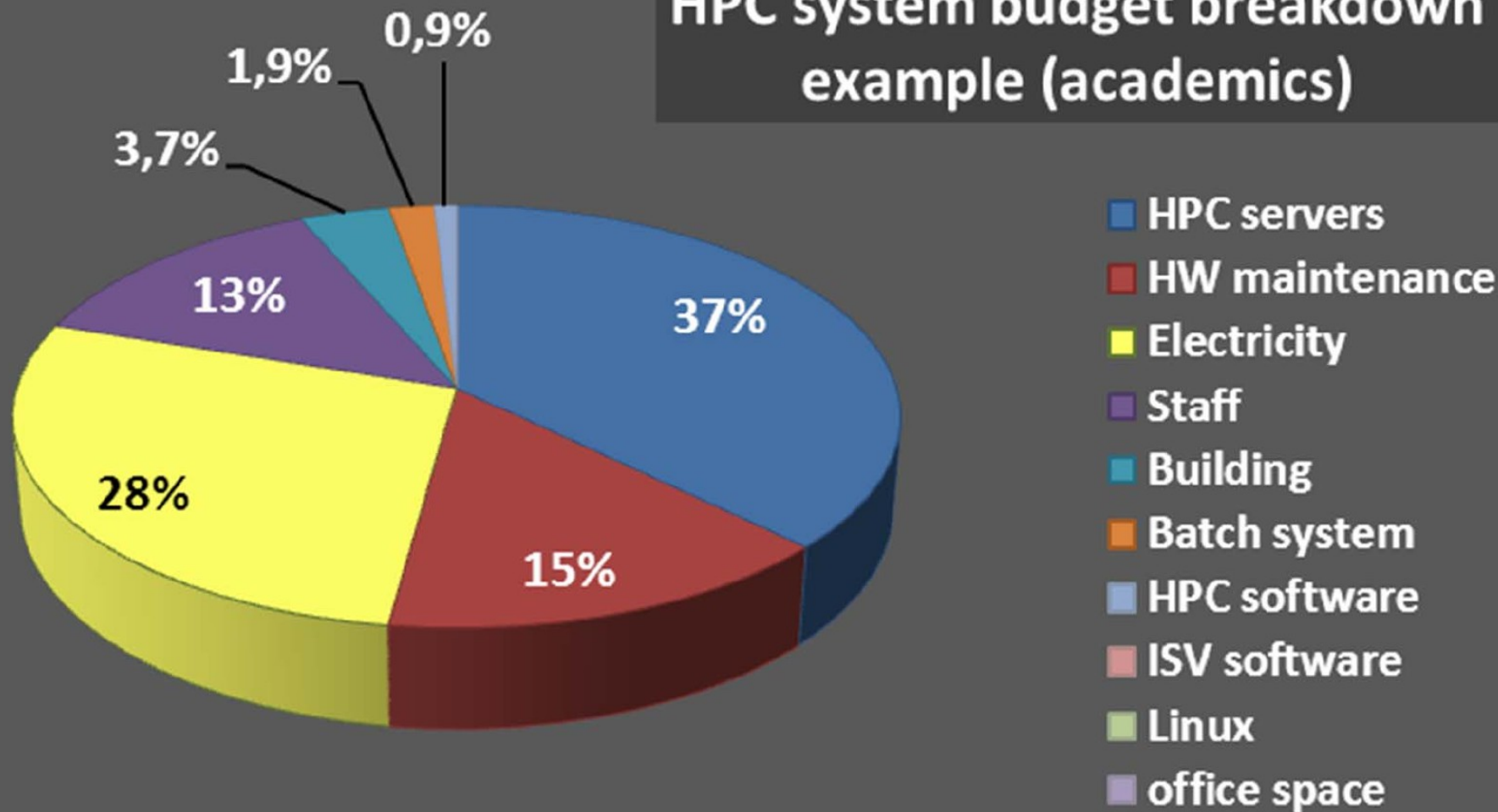


Dashboards and Profilers for greener software RVO KNGS

2/58 thomas.geenen@surfsara.nl



HPC system budget breakdown example (academics)



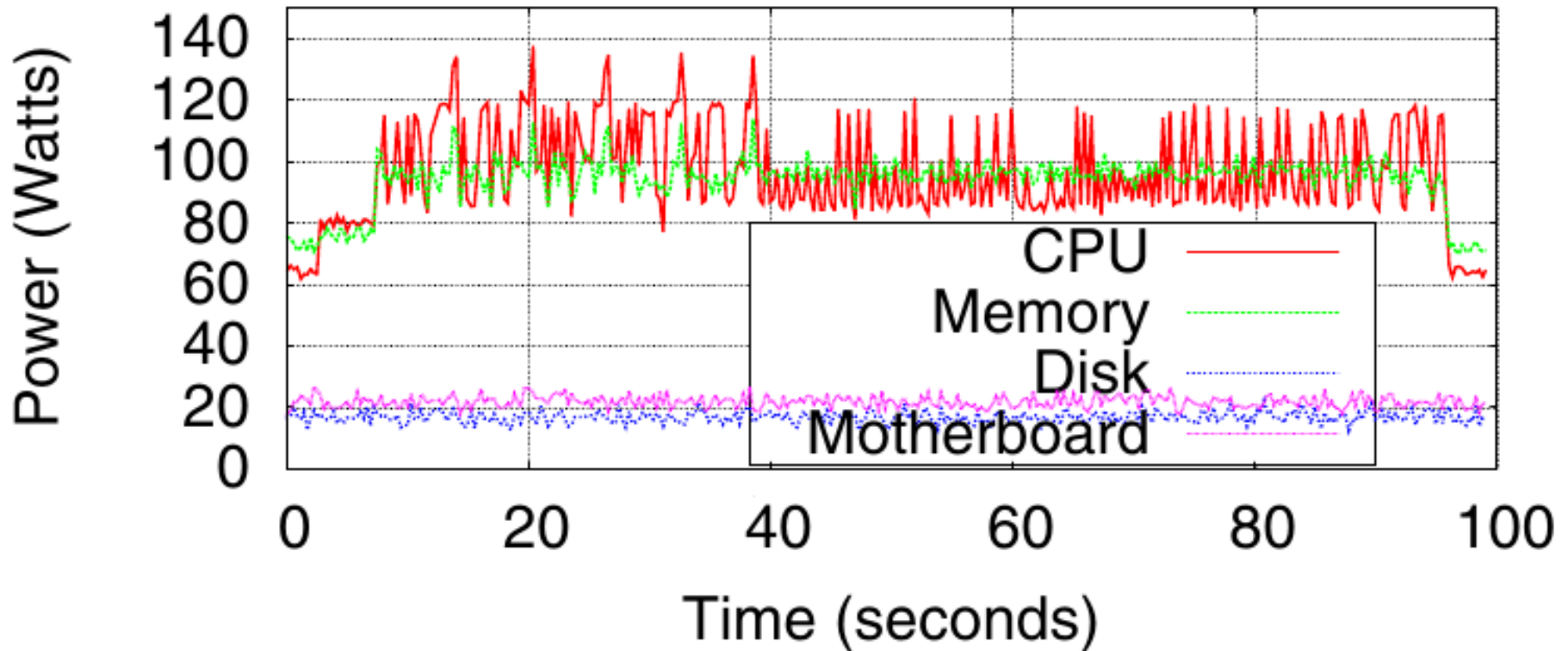
SURFsara

Power primary design constraint for future HPC systems

- Demand for more powerful hardware
- 20 MW power cap
- Today 10 PF 12 MW
- Exaflop
- 1000 PF 20 MW
- 60X improvement

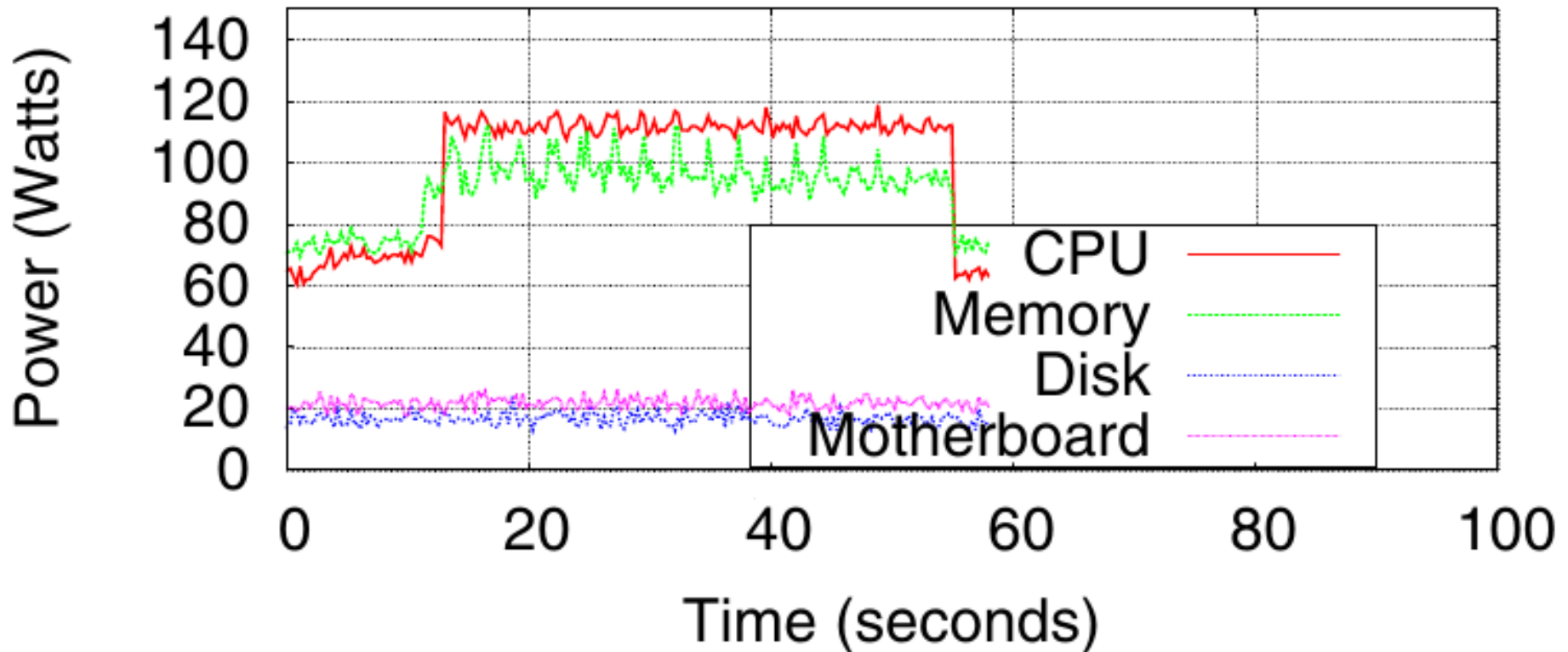


Energy consumption



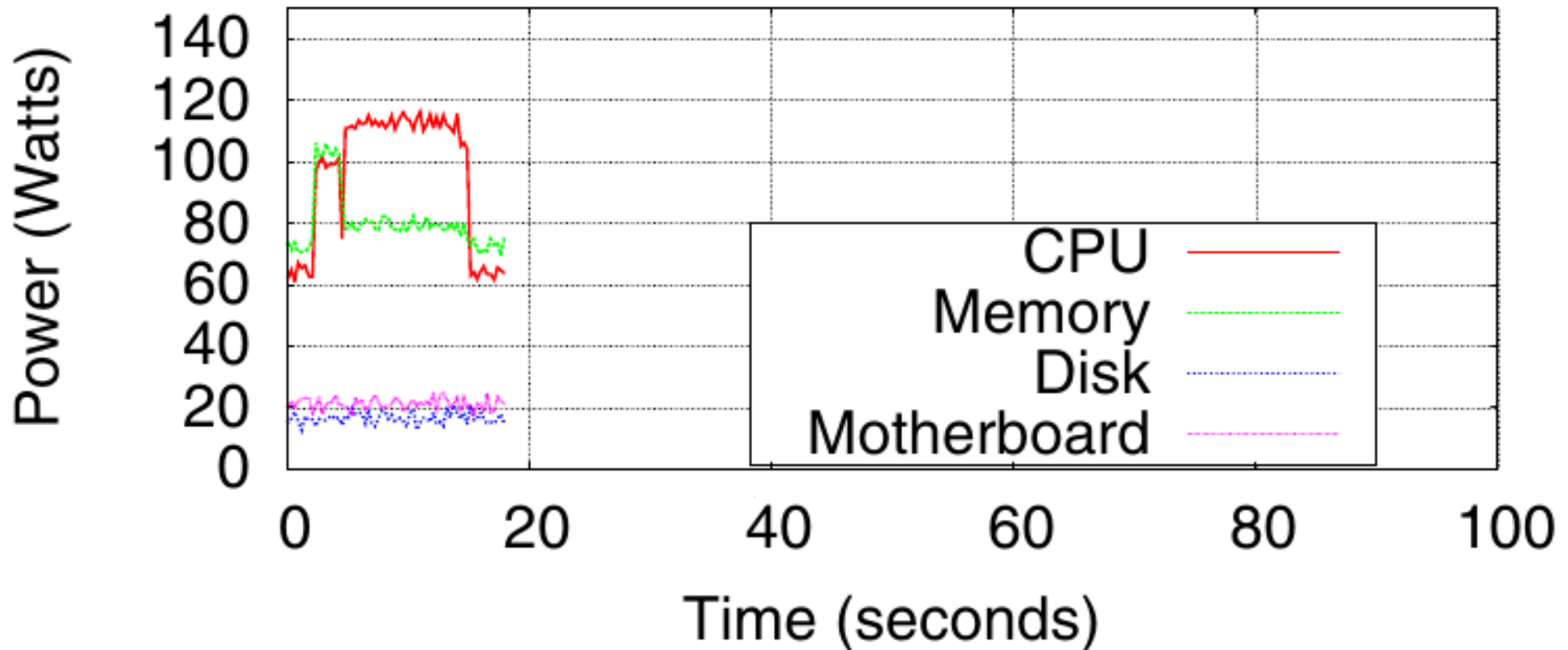
(a) LAPACK with multithreaded BLAS.

Energy consumption



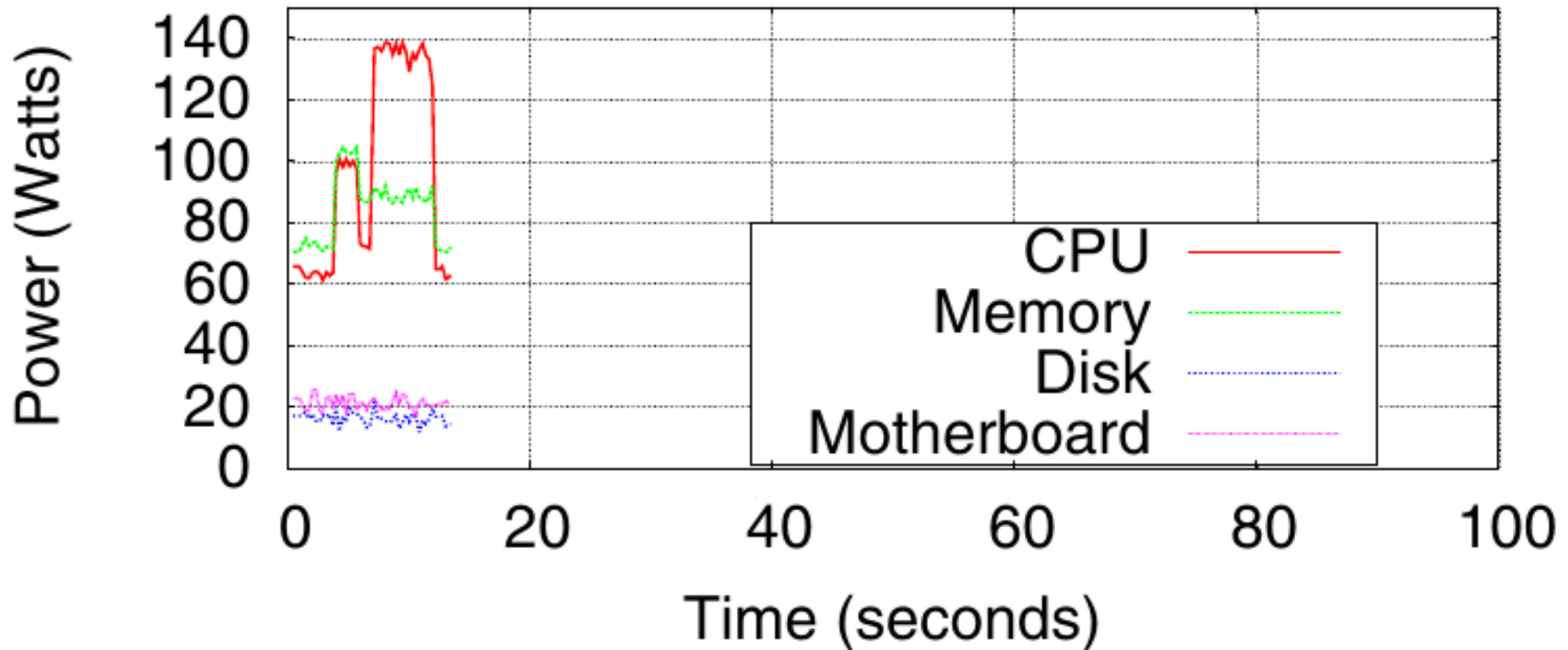
(b) MKL with multithreaded BLAS.

Energy consumption



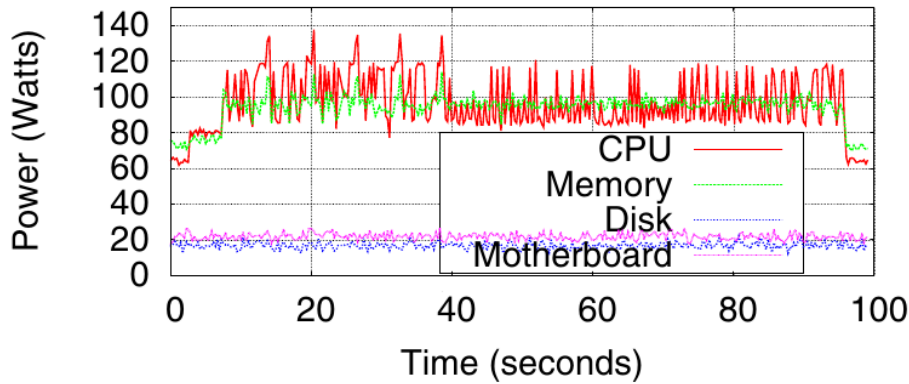
(c) PLASMA with sequential BLAS.

Energy consumption

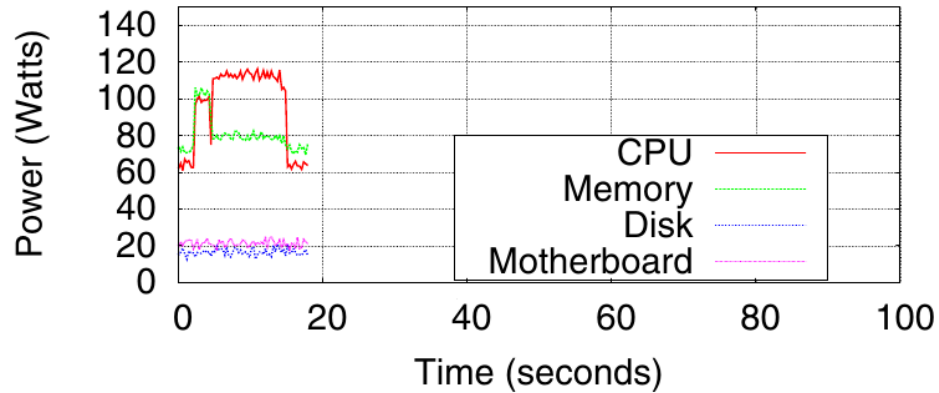


(d) PLASMA Tree Reduction with sequential BLAS.

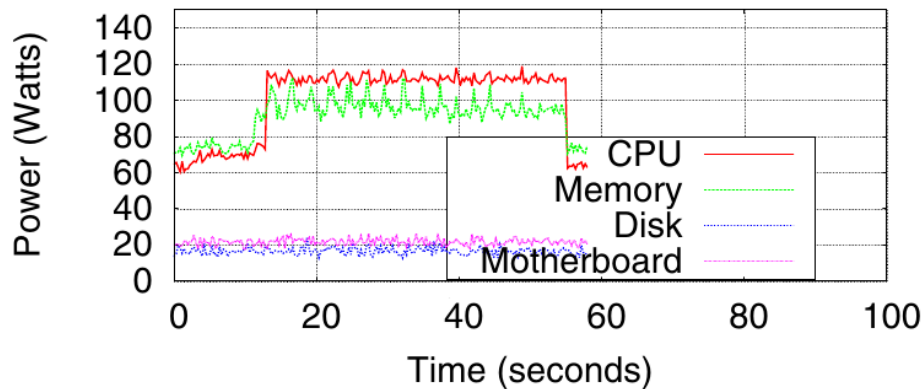
Energy consumption



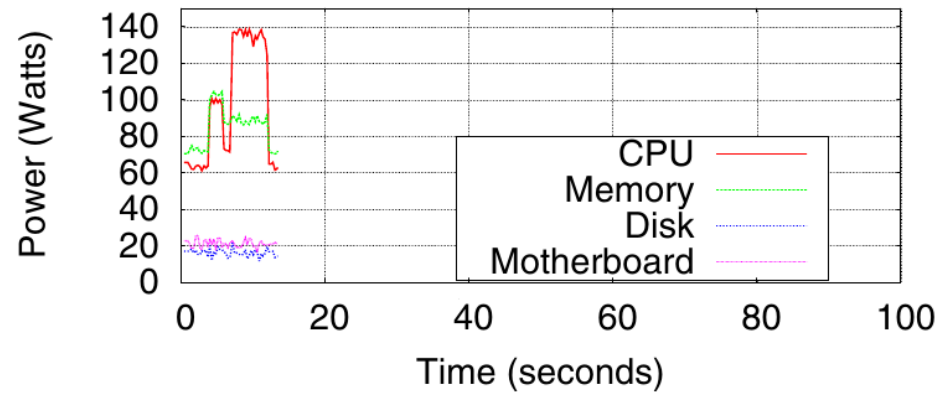
(a) LAPACK with multithreaded BLAS.



(c) PLASMA with sequential BLAS.



(b) MKL with multithreaded BLAS.

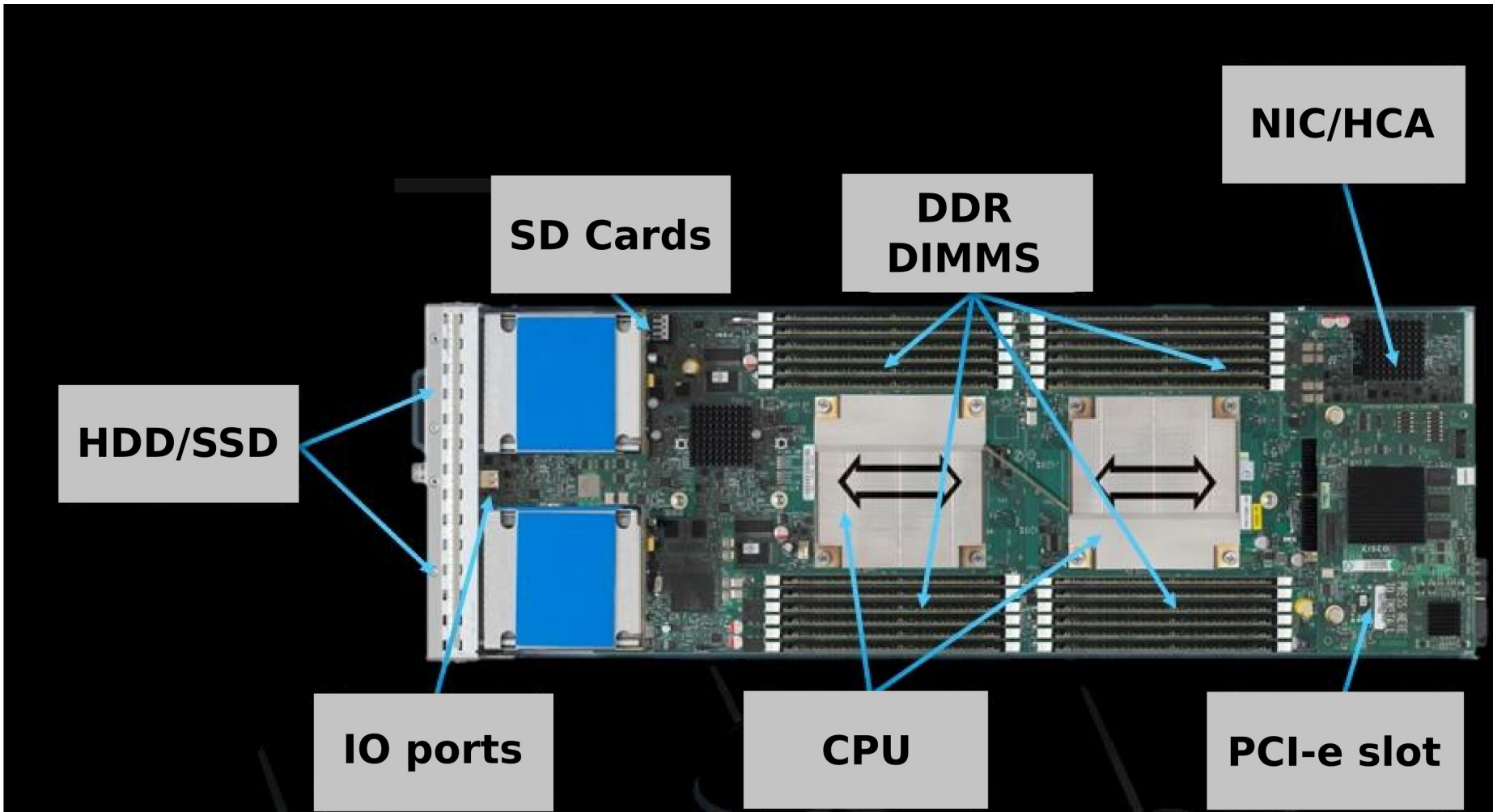


(d) PLASMA Tree Reduction with sequential BLAS.

Energy measurement

- **Different sources**
 - **Accuracy**
 - **Sampling rate**
 - **Overhead**
- **Processing measurements**
 - **Postprocessing**
 - **Visualization**
 - **Interpretation**

Node sensors



Dashboards and Profilers for greener software RVO KNGS

11/58 thomas.geenen@surfsara.nl



Node sensors

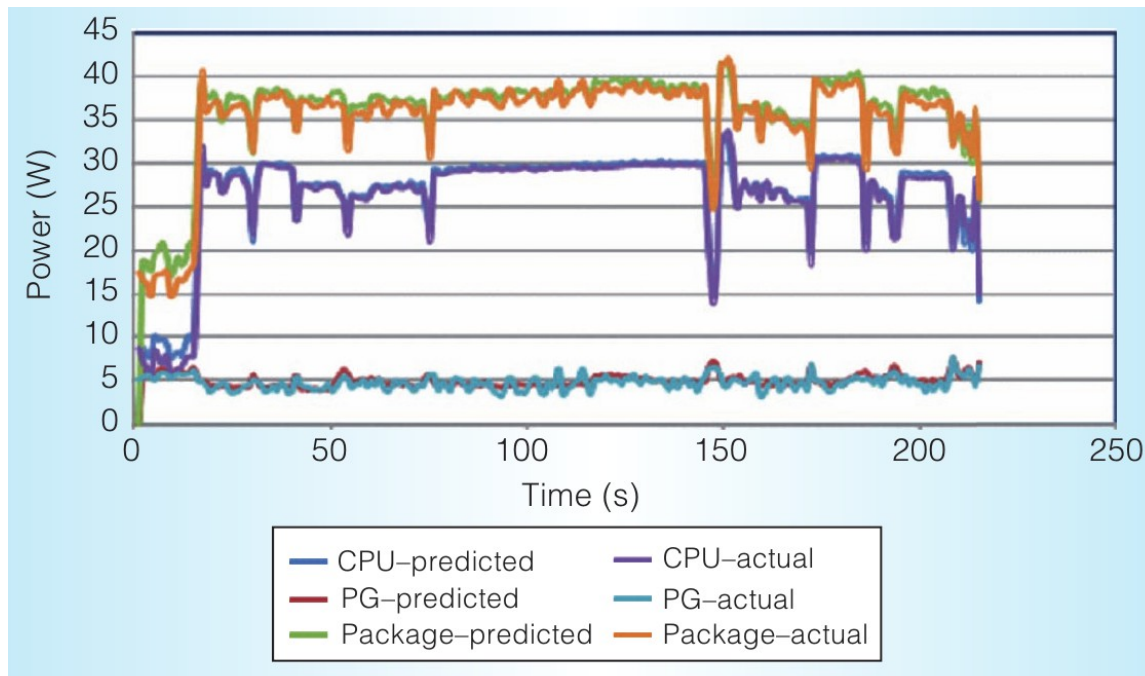
- **Different sources**
- **Direct from the CPU**
 - **Running average power limit (RAPL)**
 - **Performance Application Programming Interface (PAPI)**
- **From component**
 - **baseboard management controller (BMC)**
 - **Intel node manager**
 - **Intelligent Platform Management Interface (IPMI)**

RAPL

- **RAPL is not an analog power meter!**
- **RAPL uses a software power model**
 - **running on a helper controller**
- **Energy is estimated**
 - **using hardware performance counters**
 - **temperature, leakage models and I/O models**
- **The model is used for CPU throttling, turbo-boost**
- **Values are exposed to users**
 - **model-specific register (MSR)**

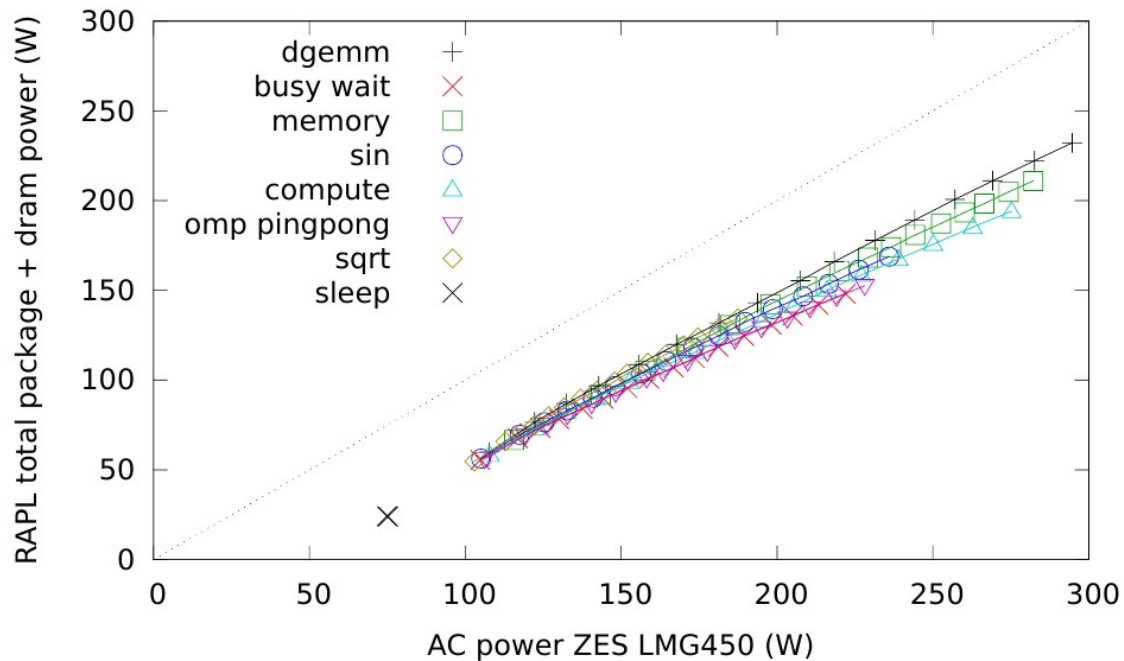
RAPL

- Intel Documentation indicates Energy readings are Updated roughly every millisecond (1 KHz)
- Rotem et al. show results match actual hardware *



RAPL

More detailed study shows small deviations for different loads



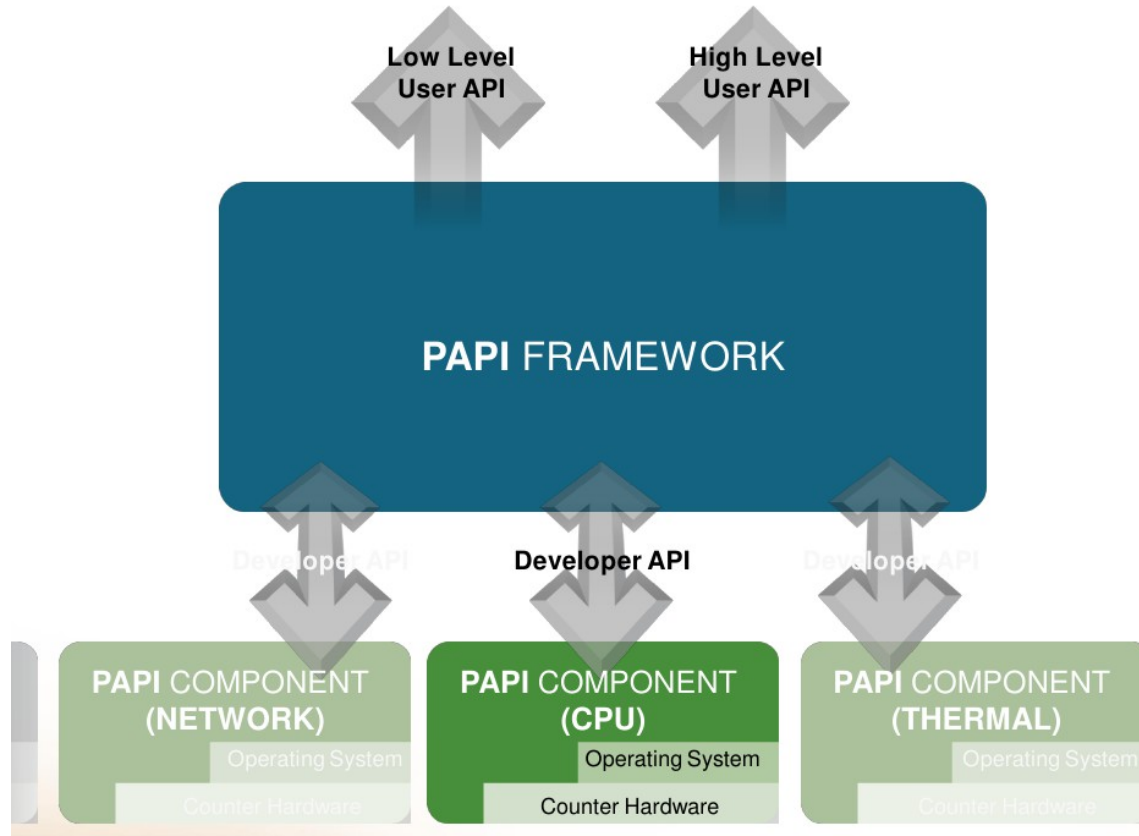
Dashboards and Profilers for greener software RVO KNCS

17/58 thomas.geenen@surfsara.nl



PAPI

performance application programming interface (PAPI)



Dashboards and Profilers for greener software RVO KNGS

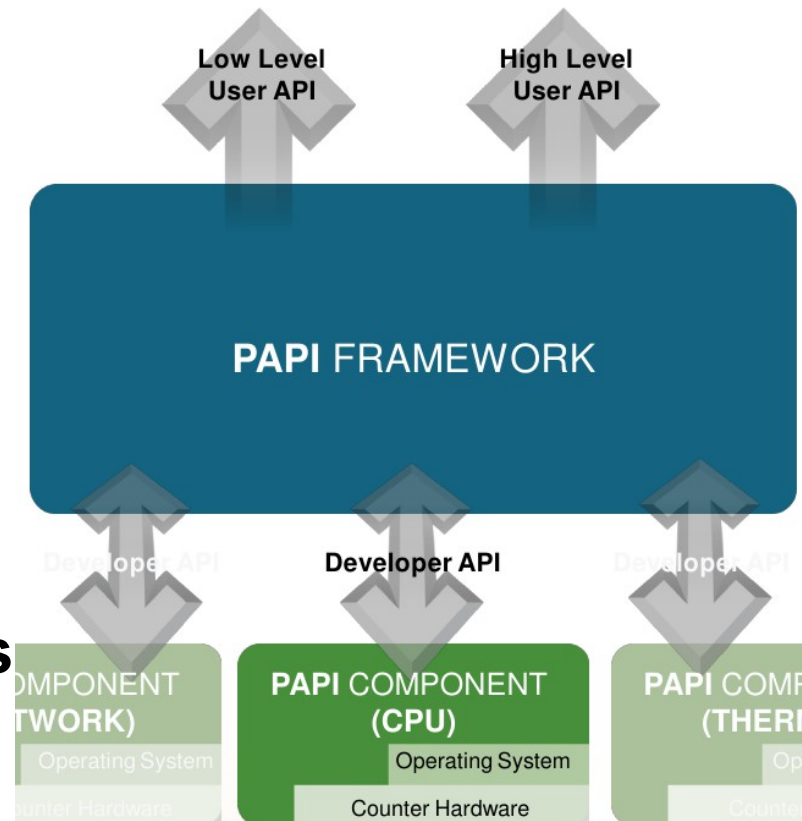
18/58 thomas.geenen@surfsara.nl



MSRs can be accessed via `/dev/cpu/*/msr`

PAPI

- Performance application programming interface (PAPI)
- Read special registers (MSR)
- Performance counter hardware
- Intel, AMD, NVIDIA, ARM
 - RAPL, APM, NVML, custom
- Measure energy and
 - Flops, cycles
 - Memory access, cache misses
 - Ivy bridge 11 counters



Dashboards and Profilers for greener software RVO KNGS

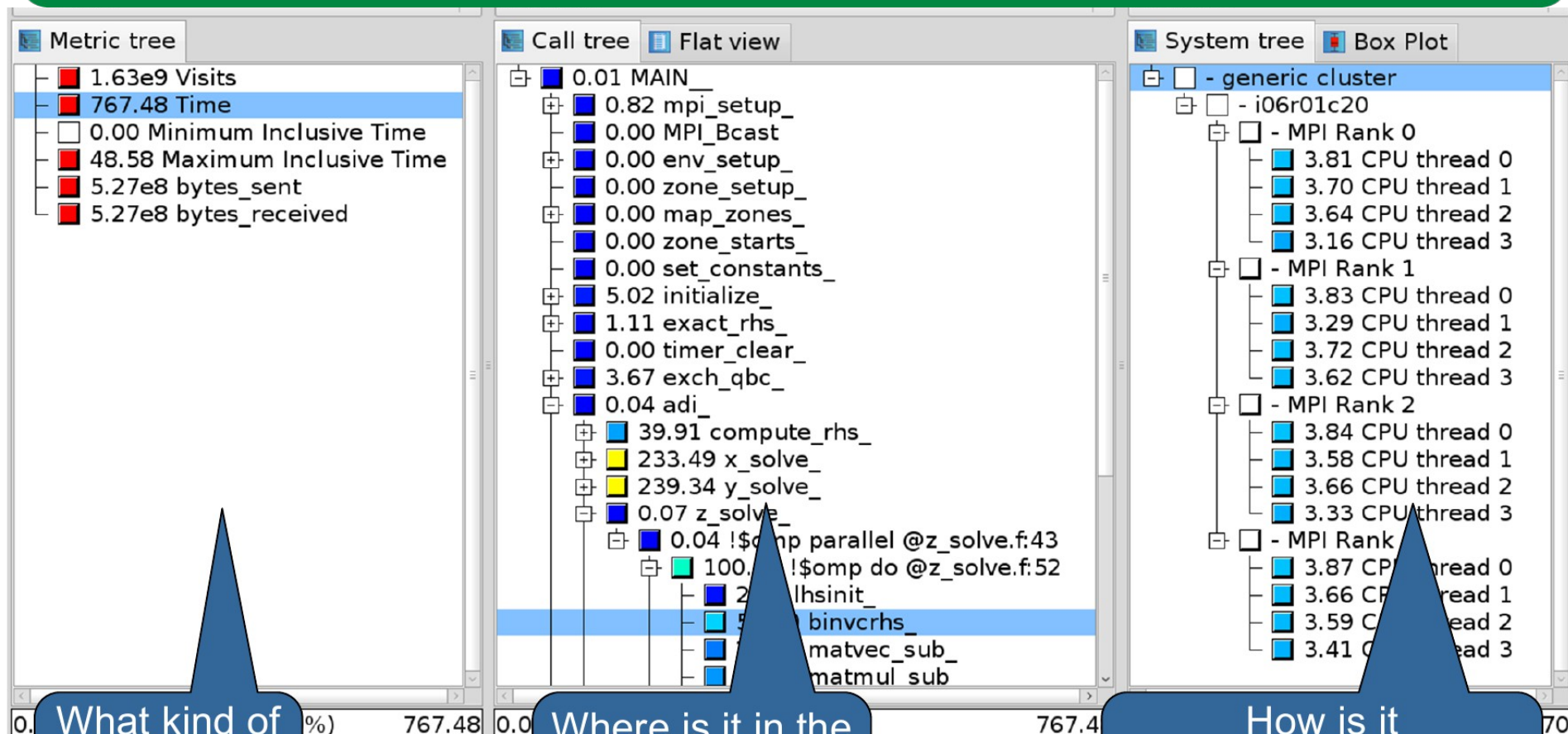
19/58 thomas.geenen@surfsara.nl



Profiling applications

- **Time**
 - **Where is the time spend**
- **What is the application doing**
 - **PAPI (hardware calls)**
 - **MPI (communication between processes)**
 - **OpenMP (communication between threads)**
- **Couple with energy consumption**
 - **Same profile**

Profiling applications

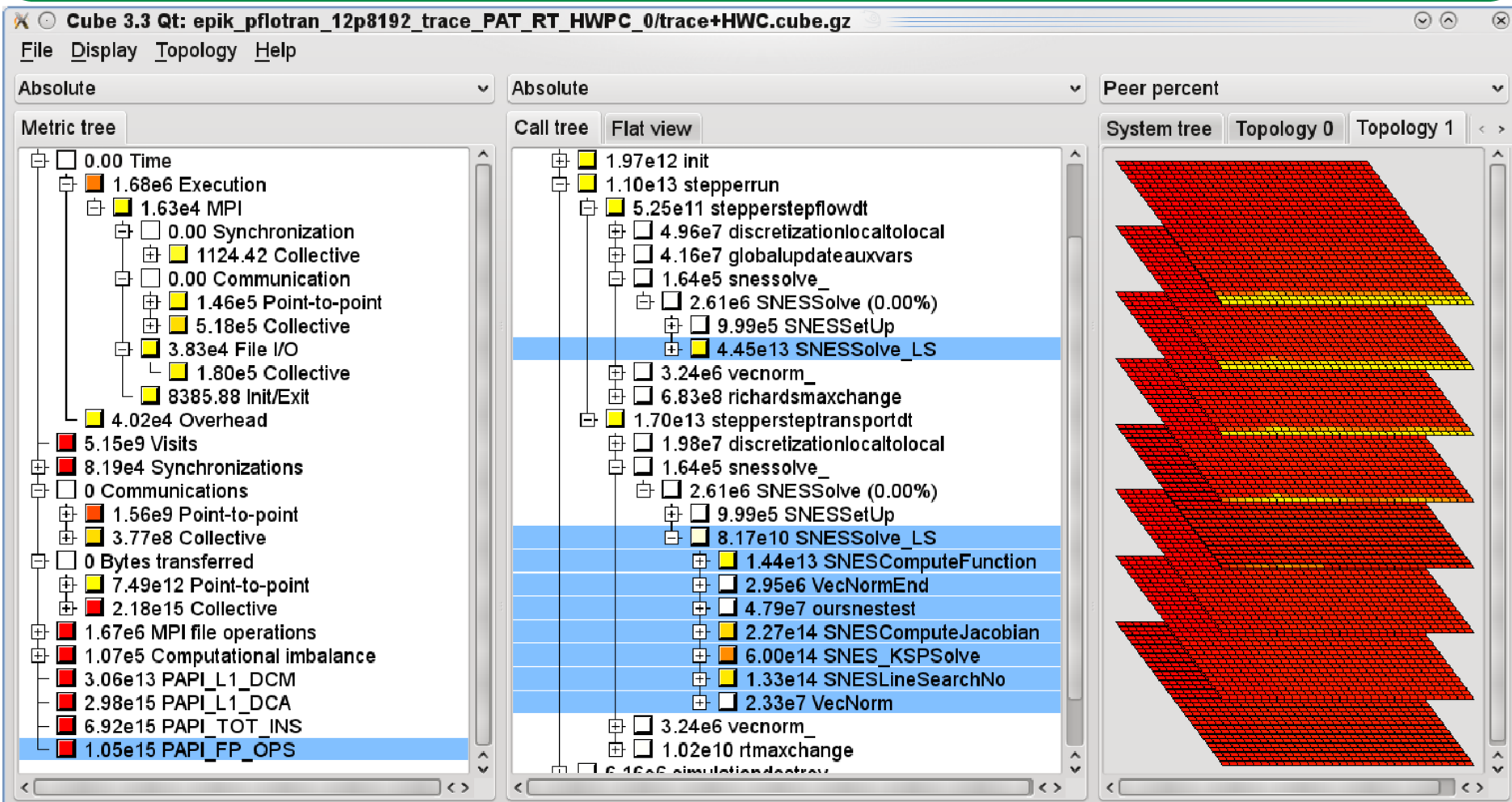


What kind of performance metric?

Where is it in the source code? In what context?

How is it distributed across the processes/threads?

Profiling applications



Dashboards and Profilers for greener software RVO KNCS

22/58 thomas.geenen@surfsara.nl



Profiling applications

Temperature

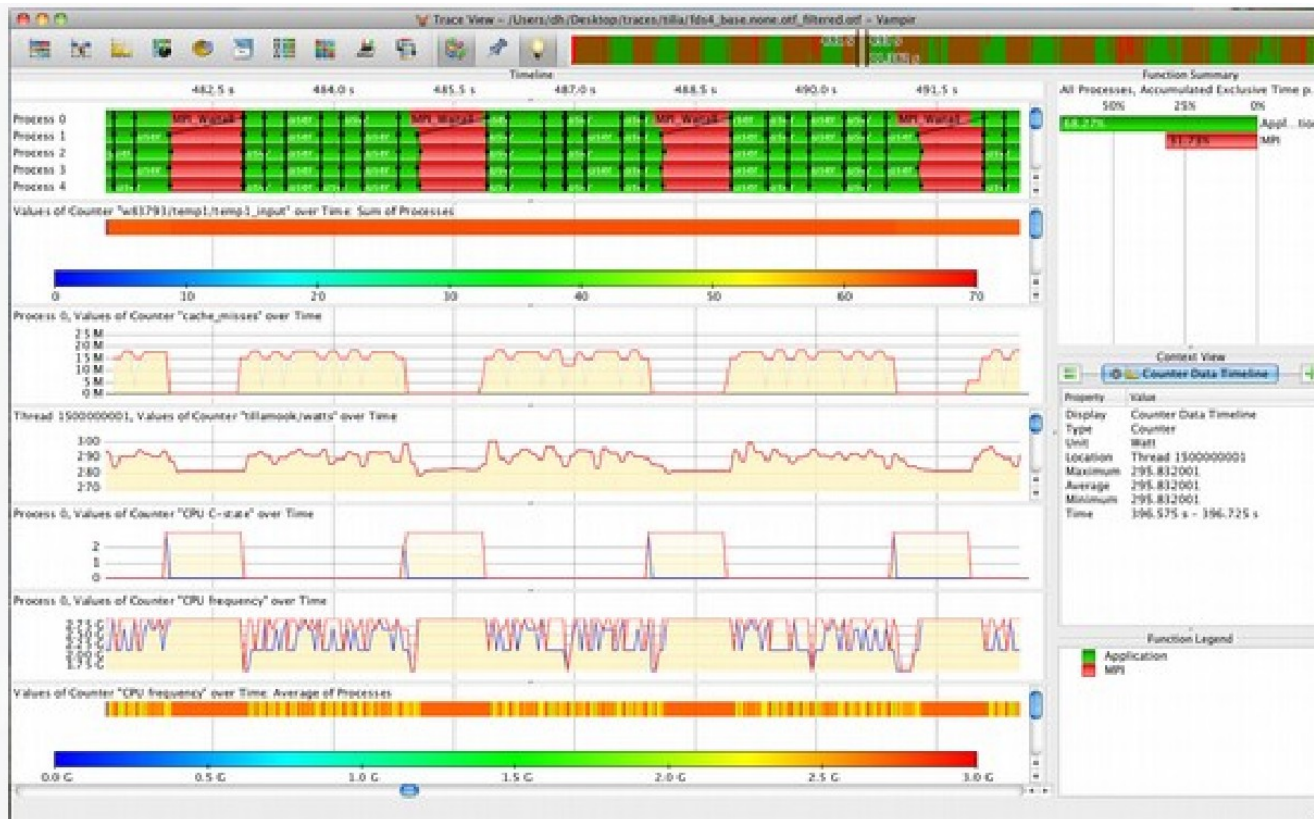
Cache Misses

Power consumption

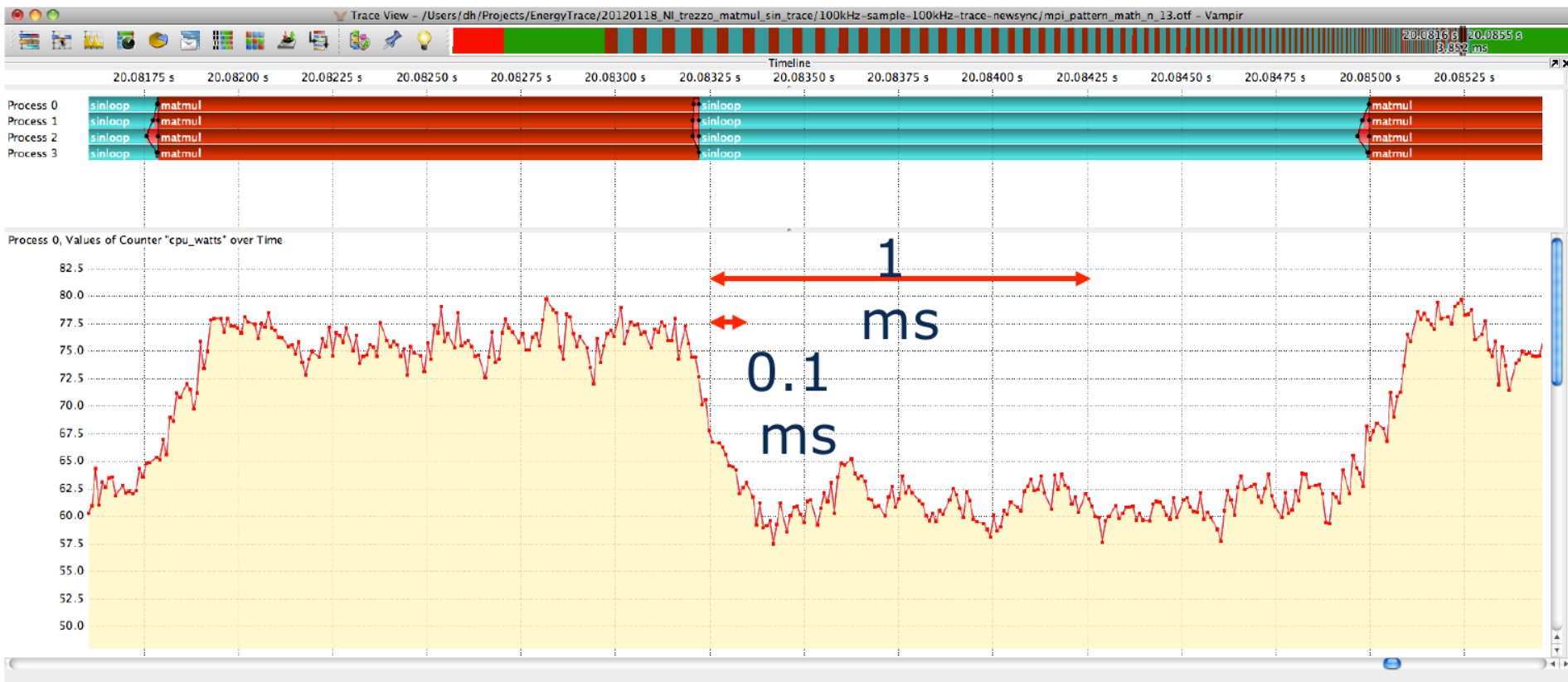
Proc. sleep state

Proc. frequency

Avg. proc. frequency



Profiling applications

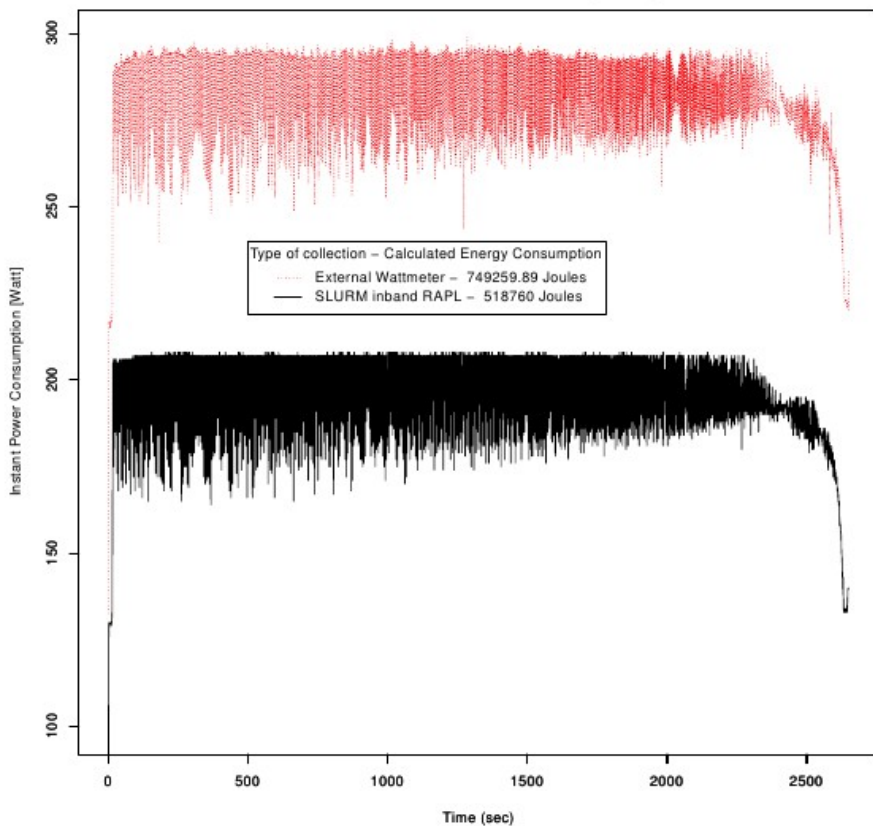




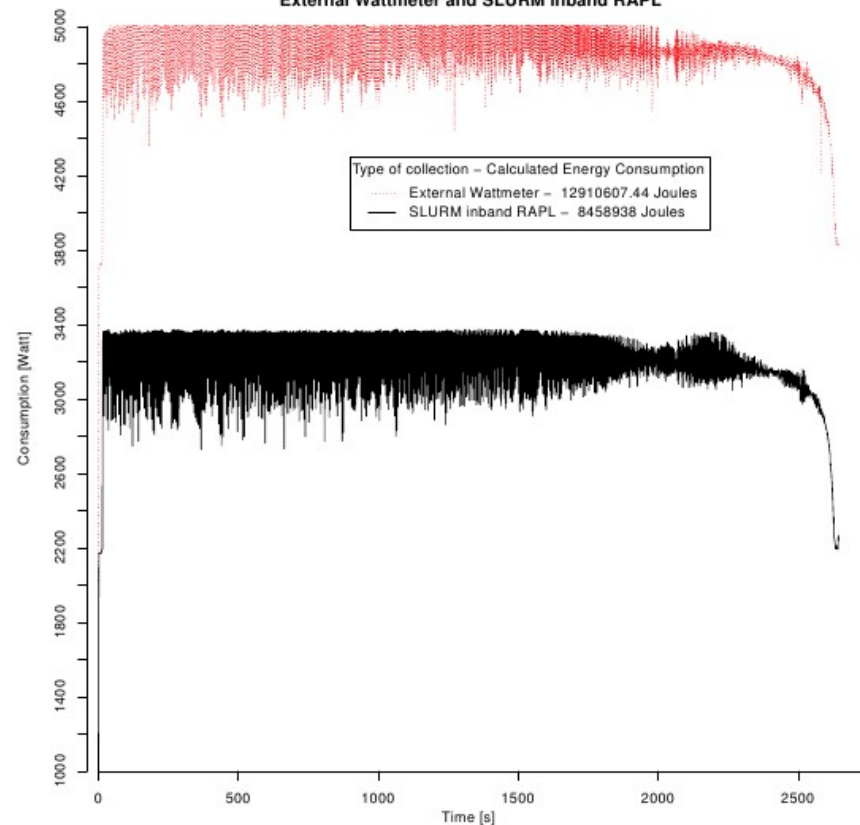
Line	Source	INST_RE... ANY by Package	CPU_CLK... THREAD by Package	CPU_CLK... REF by Package
31	void multiply1(int msize, int tidx, int numt, TYPE a[][NUM			
32	{			
33	int i,j,k;		2	
34				
35	// Naive implementation			
36	for(i=tidx; i<msize; i=i+numt) {			
37	for(j=0; j<msize; j++) {			13
38	for(k=0; k<msize; k++) {	37,145	234,511	221,
39	c[i][j] = c[i][j] + a[i][k] * b[k][j];	1,507	13,365	13,
40	}			
41	}			
42	}			

IPMI BMC

Power consumption of one node measured through External Wattmeter and SLURM inband RAPL during a Linpack on 16 nodes

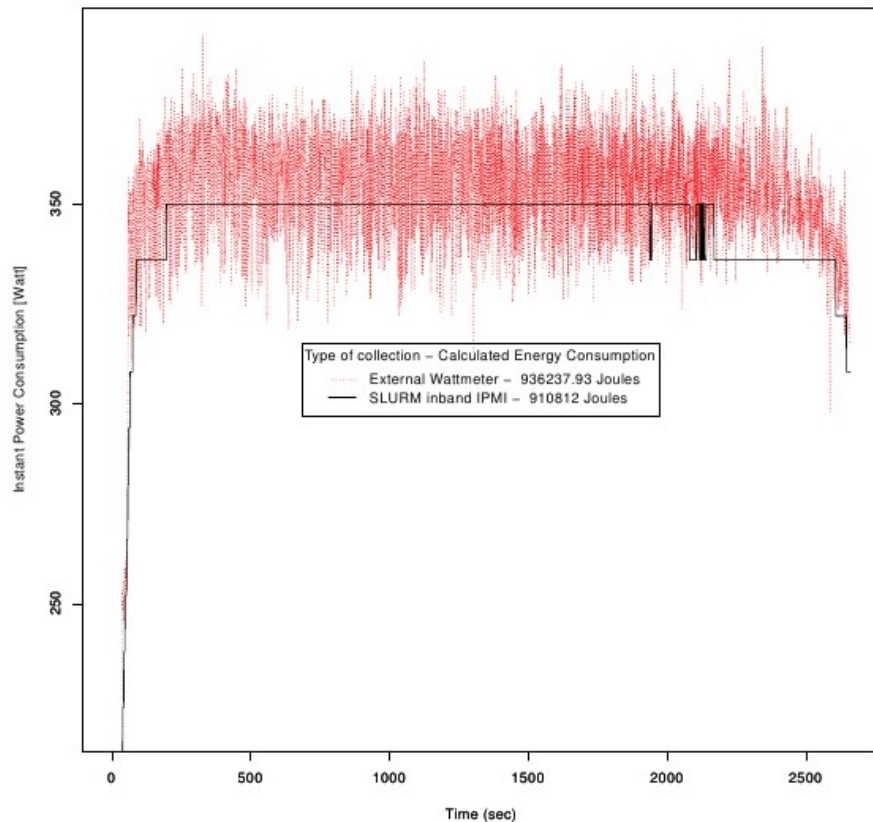


Power consumption of Linpack execution upon 16 nodes measured through External Wattmeter and SLURM inband RAPL

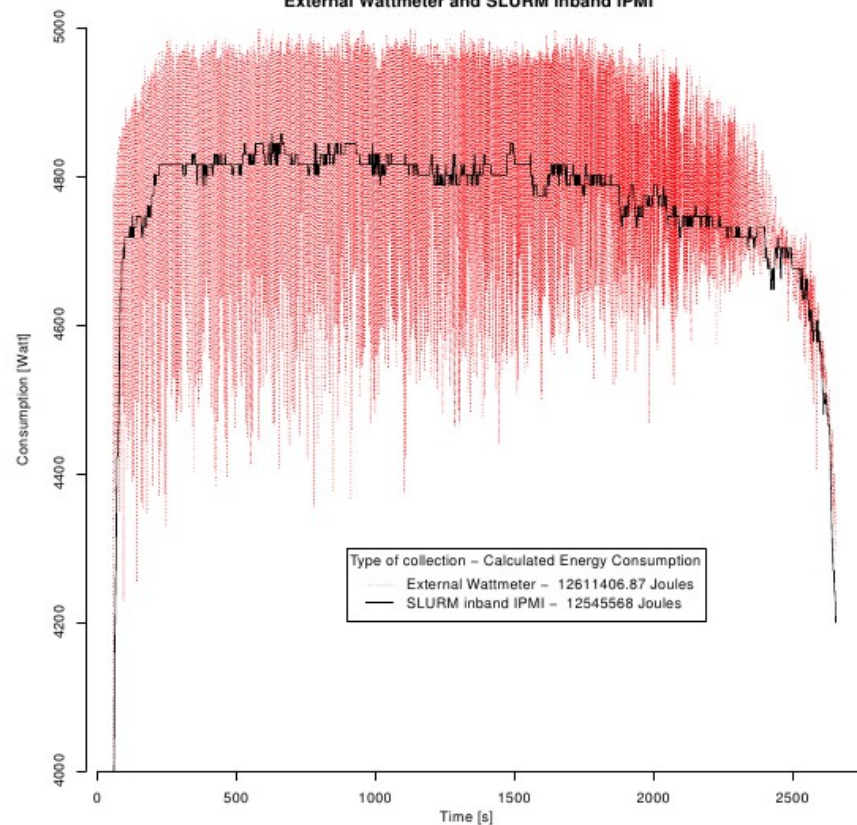


IPMI BMC

Power consumption of one node measured through
External Wattmeter and SLURM inband IPMI during a Linpack on 16 nodes



Power consumption of Linpack execution upon 16 nodes measured through
External Wattmeter and SLURM inband IPMI



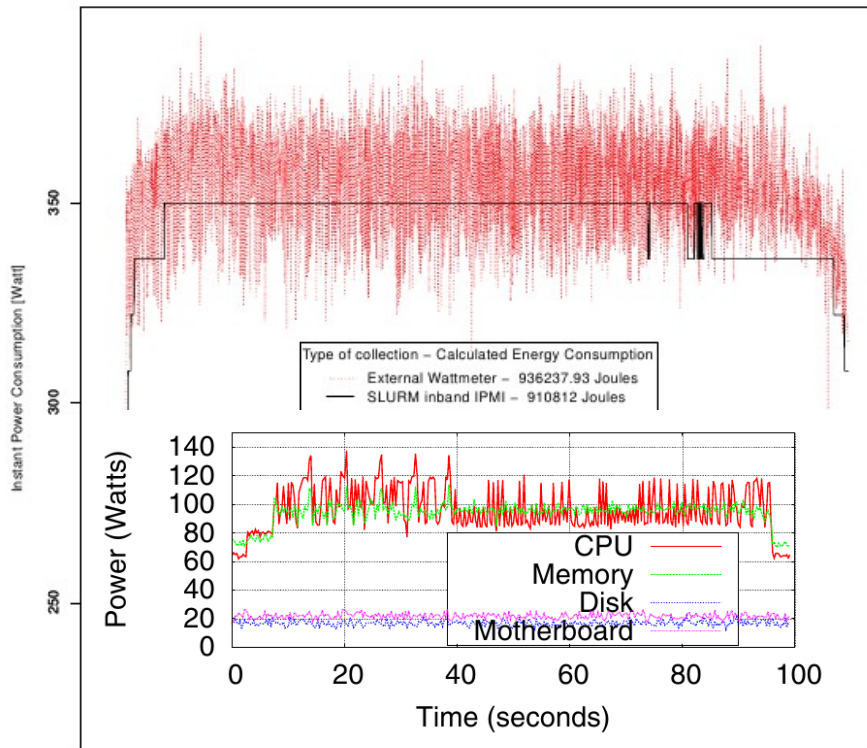
Dashboards and Profilers for greener software RVO KNGS

27/58 thomas.geenen@surfsara.nl



IPMI BMC

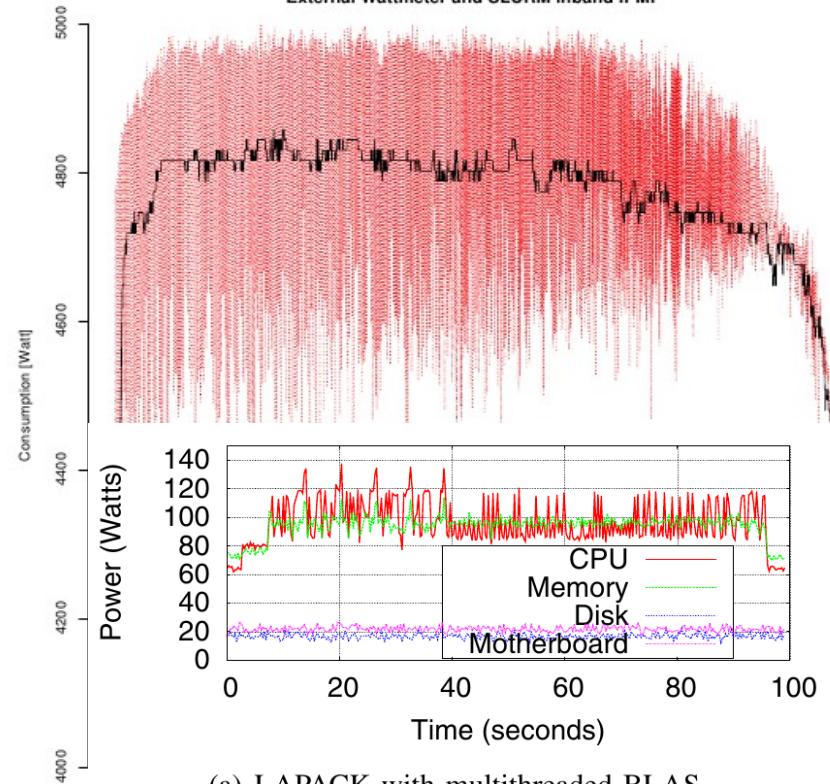
Power consumption of one node measured through External Wattmeter and SLURM inband IPMI during a Linpack on 16 nodes



(a) LAPACK with multithreaded BLAS.

Time (sec)

Power consumption of Linpack execution upon 16 nodes measured through External Wattmeter and SLURM inband IPMI



(a) LAPACK with multithreaded BLAS.

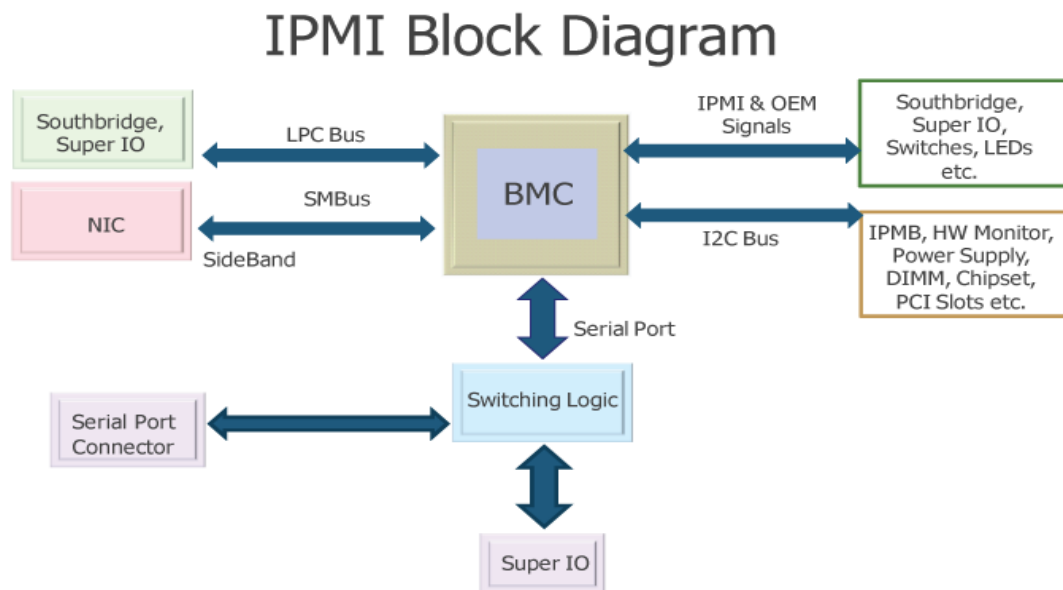
Dashboards and Profilers for greener software RVO KNGS

28/58 thomas.geenen@surfsara.nl



IPMI BMC

- Measure energy consumption of other components
- Baseboard Management Controller (BMC)
- IPMI
- Low sample rate
 - 1 – 4 Hz
- Overhead
- Improves
 - On chip averaging
 - Higher sample rate
 - Still low



Reporting

- **What do we want to present to the end user**
- **Can use PAPI and tools for detailed analysis**
 - **Misses part of the energy consumption**
- **Information on per-run level**
 - **Energy consumption per run (total)**
 - **More general view (total per component)**
 - **Timeline**
- **Correlate with other data**
 - **PAPI and BMC**

SLURM

Use the job scheduler to collect energy consumption data

Typical situation on HPC systems

- Many users on the same system
- Share resources
- Have to schedule jobs
 - Job is put in a queue
 - Runs when resources are available
- SLURM
 - Simple Linux Utility for Resource Management
 - Open source

SLURM

Use the job scheduler to collect energy consumption data

- **Modular design**
 - **Plugins for monitoring**
 - **Energy consumption**
 - **RAPL**
 - **IPMI**
 - **Grand total**
 - **Timeline**
 - **Uses additional threads to collect data (IPMI)**

SLURM

Use the job scheduler to collect energy consumption data

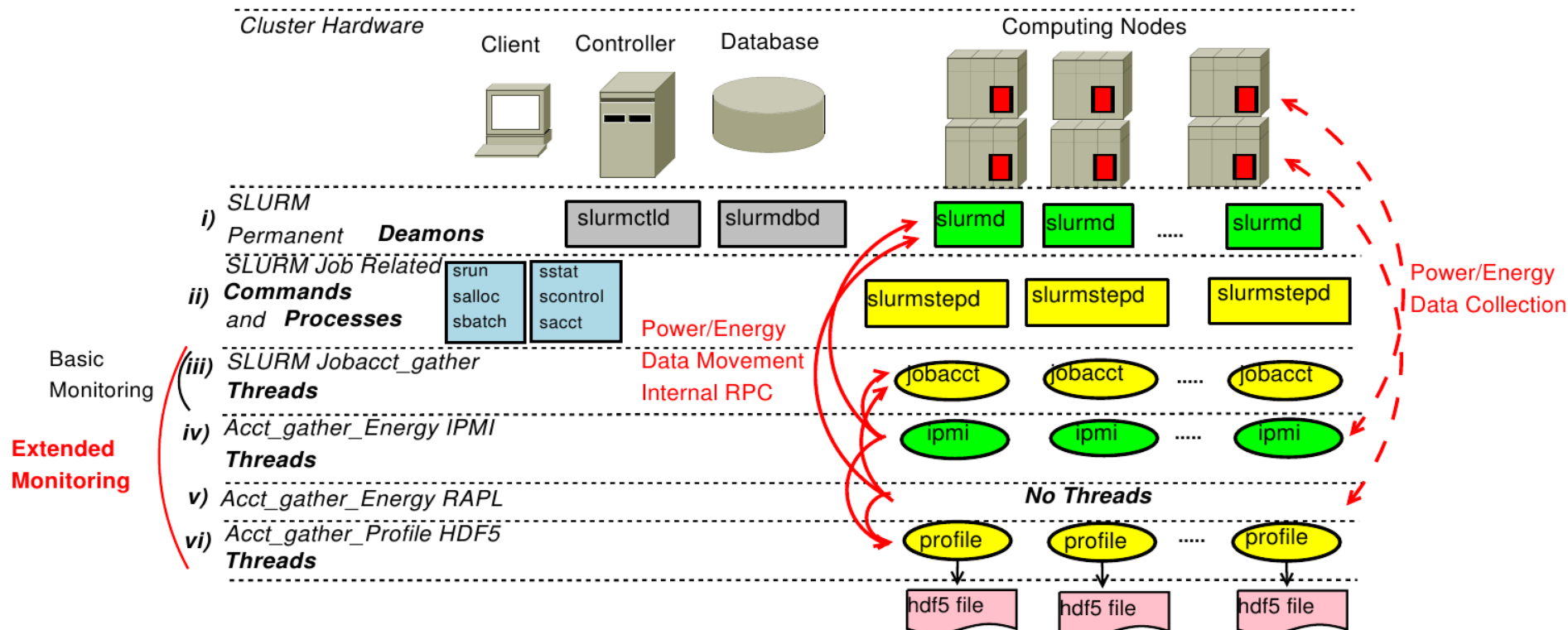


Fig. 1. SLURM Monitoring Framework Architecture with the different monitoring modes

SLURM

Use the job scheduler to collect energy consumption data

- Totals in database
- Timeseries in file
- HDF5 (XML)
 - Scalable data format
- Individual sensors (IPMI)
- RAPL
- External sensor

The screenshot shows a job scheduler interface with a tree view on the left and two data tables on the right. The tree view shows a hierarchy: RBS~42.h5 > Step~0 > Nodes > Node~0 > Energy > Energy Data. Below this, there are Tasks (Task~0 to Task~3) and Disk > Disk Data. The right side shows two data tables. The first table, titled 'Energy Data', has columns for Time, Temperature, and Power. The second table, titled 'Disk Data', has columns for reads, writes, read_bytes, and write_bytes.

	Time	Temperature	Power
0	1360343967	88	100.84693...
1	1360343967	87	101.71463...
2	1360343967	89	100.42423...
3	1360343967	83	101.64976...
4	1360343967	82	101.18964...
5	1360343967	84	101.35049...
6	1360343967	83	101.10252...
7	1360343967	89	101.96751...
8	1360343967	86	101.54038...
9	1360343967	81	101.30345...

	reads	writes	read_bytes	write_bytes
0	42	6211948	3989528	8634034

Conclusions

- **Many sensors available on current cluster hardware**
- **Different levels of detail**
- **Many profilers available**
 - **Common api PAPI**
 - **Combine with performance metrics**
- **Present totals to users**
- **Combine different measurements in one file (time series)**
 - **Slurm tools**

QUESTIONS?