Energy Accounting and Control on HPC clusters

Yiannis Georgiou

R&D Software Engineer

Architect of an Open World™

Objectives

Issues that we wanted to deal with:

- Measure power and energy consumption on HPC clusters
- Attribute power and energy data to HPC components
- Calculate the energy consumption of jobs in the system
- Extract power consumption time series of jobs
- Control the Energy usage during the job execution

Objectives

Issues that we wanted to deal with:

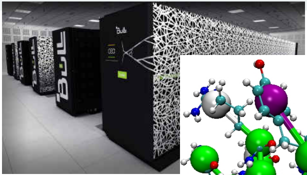
- Measure power and energy consumption on HPC clusters
- Attribute power and energy data to HPC components
- Calculate the energy consumption of jobs in the system
- Extract power consumption time series of jobs
- Control the Energy usage during the job execution

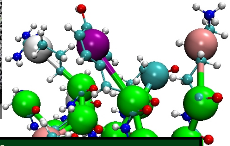
Why?

- **Measuring Energy** will enable us to **use it more efficiently**:
  - Motivate users to better exploit the power of their resources
  - Use the power/energy data as input information for central software that can take actions

Context

High Performance Computing





Infrastructures:

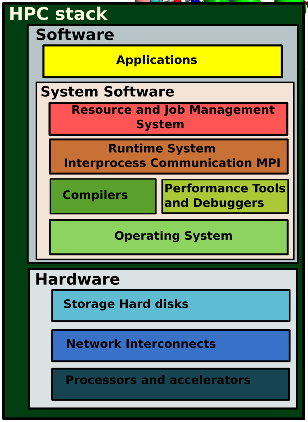
- Supercomputers, Clusters, Grids, Clouds

Applications:

- Climate Prediction, Protein Folding, Crash simulation, High-Energy Physics, Astrophysics, Rendering

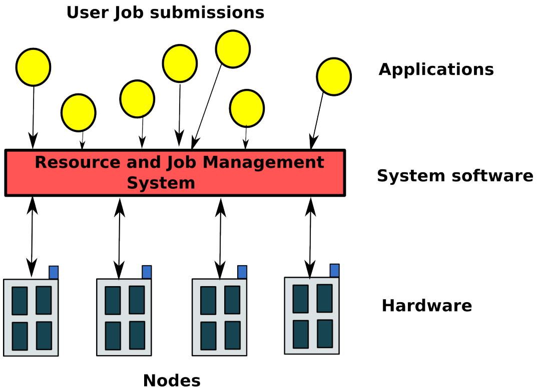
System Software

- System Software: Operating System, Runtime system, Resource Management, I/O Systems, Interfacing to External Environments



Resource and Job Management

The goal of a Resource and Job Management System (RJMS) is to satisfy users' demands for computation and assign user jobs upon the computational resources in an efficient manner.



RJMS Importance

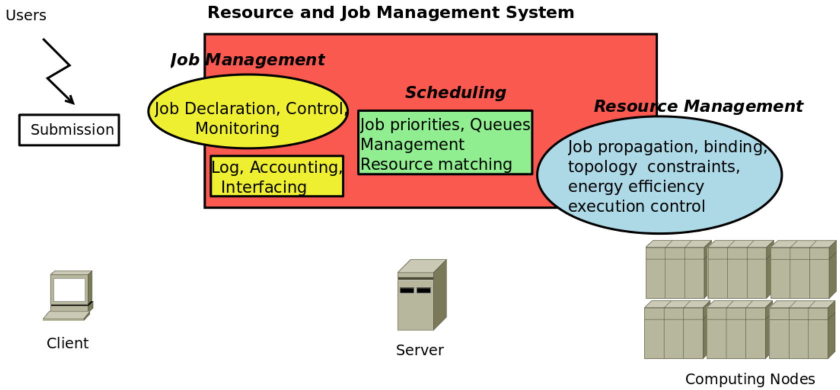
Strategic position but complex internals:

- **Direct and constant knowledge** of resources and jobs
- **Multifacet procedures** with complex internal functions

RJMS abstraction layers

This assignement involves three principal abstraction layers:

- **Job Management** layer
- the **Scheduling** layer
- and the **Resource Management** layer



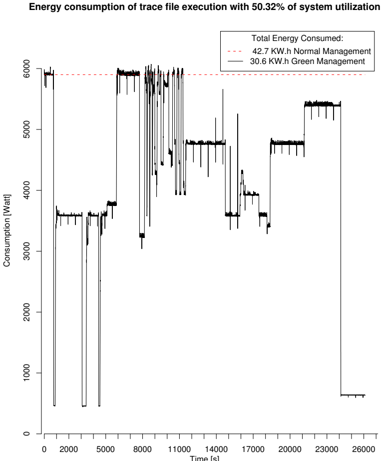
RJMS and Power Management

- Constant knowledge of the resources and jobs
- Take advantage of the **strategic position** of the RJMS software
- To treat **Energy** as a **new type of Resources** to be used by jobs

RJMS and Power Management

Existing mechanisms for energy reductions on most of todays RJMS (System side feature):

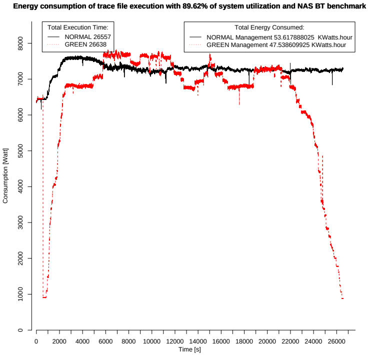
- Framework for energy saving through unutilized nodes
  - Administrator configurable actions (hibernate, DVFS, power off,etc)
  - Automatic “Wake up” when jobs arrive

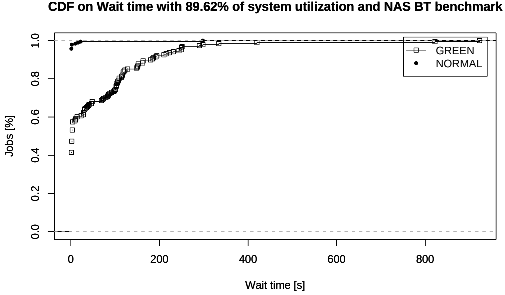


RJMS and Power Management

Issues:

- Multiple Reboots: **Risks** for node crashes or other hardware components problems
- Most of production HPC clusters have a nearly constant 90% or **higher utilization** hence the gain can be trivial
- **TradeOffs:** Jobs Waiting time increases significantly





RJMS and Power Management

New issues to deal with:

- To enable energy reductions even with 100% system utilization

RJMS and Power Management

New issues to deal with:

- To enable energy reductions even with 100% system utilization

How?

- Make energy consumption a user concern:
  - **Energy Accounting**: Turn Energy Consumption to a new job characteristic
  - **Energy Control**: Allow users to control the energy consumption of their jobs

RJMS and Power Management

New issues to deal with:

- To enable energy reductions even with 100% system utilization

How?

- Make energy consumption a user concern:
  - **Energy Accounting**: Turn Energy Consumption to a new job characteristic
  - **Energy Control**: Allow users to control the energy consumption of their jobs

Basic need:

- Ways to monitor and measure energy consumption

Plan

Plan

SLURM scalable and flexible RJMS

- **SLURM open-source** Resource and Job Management System, sources freely available under the GNU General Public License. `https://github.com/SchedMD/slurm/`
- **Portable:** written in C with a GNU autoconf configuration engine.
- **Modular:** Based on a plugin mechanism used to support different kind of scheduling policies, interconnects, libraries, etc
- **Robust:** highly tolerant of system failures, including failure of the node executing its control functions.
- **Scalable:** designed to operate in a heterogeneous cluster with up to tens of millions of processors. It can accept 1000 job submissions per second and fully execute 500 simple jobs per second (depending upon hardware and system configuration).

SLURM History and Facts

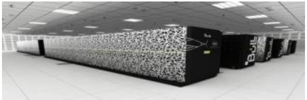
- Developed in LLNL since 2003, passed to **SchedMD since 2011**
- **Multiple enterprises and research centers** have been contributing to the project (LANL, Cray, Intel, CEA, HP, BULL, BSC, etc)
- **Large international community** Active mailing lists (support by main developers)
- Contributions (various external software and standards are integrated upon SLURM)

SLURM History and Facts

- Used in more than **50% of worlds largest supercomputers of Top500** list amongst which the following 5 from Top10:
  - #1: Tianhe-2 at the National University of Defense Technology
  - #3: Sequoia, an IBM BlueGeneQ at Lawrence Livermore National Laboratory
  - #6: Piz Daint, a Cray XC30 at the Swiss National Supercomputing Center
  - #7: Stampede, a Dell cluster at the Texas Advanced Computing Center
  - #9: Vulcan, an IBM BlueGeneQ at Lawrence Livermore National Laboratory

BULL and SLURM

- BULL initially started to work with SLURM in 2005
- At least 5 BULL active developers since then:
  - **Development** for new SLURM features
  - **Bug Corrections and Support** for clients
  - **All BULL developments are given to community(open-source)**, no code is kept proprietary except some small parts related to particular BULL hardware or software
- Integrated into the bullx Extreme computing software stack since 2006
  - Used as the default RJMS of the bullx stack
  - Deployed upon the **BULL-CEA petaflopic supercomputers** Curie, Tera100, Helios
- Close development collaboration between SchedMD,CEA and BULL
- Annual User Group Meeting (User, Admin Tutorials + Technical presentation for developpers)
  `http://www.schedmd.com/slurmdocs/publications.html`



bullx supercomputer suite

Plan

Summary of the energy accounting and control features

- Power and Energy consumption monitoring per node level.
- Energy consumption accounting per step/job on SLURM DataBase
- Power profiling per step/job on the end of job
- Frequency Selection Mechanisms for user control of job energy consumption

Summary of the energy accounting and control features

- Power and Energy consumption monitoring per node level.
- Energy consumption accounting per step/job on SLURM DataBase
- Power profiling per step/job on the end of job
- Frequency Selection Mechanisms for user control of job energy consumption

How this takes place:

- Dedicated Plugins for Support of **in-band collection of energy/power data** (IPMI / RAPL)
- Dedicated Plugins for Support of **out-of-band collection** of energy/power data (RRD databases)
- **Power data job profiling** with HDF5 file format
- SLURM Internal power-to-energy and energy-to-power calculations

Summary of the energy accounting and control features

- Power and Energy consumption monitoring per node level.
- Energy consumption accounting per step/job on SLURM DataBase
- Power profiling per step/job on the end of job
- Frequency Selection Mechanisms for user control of job energy consumption

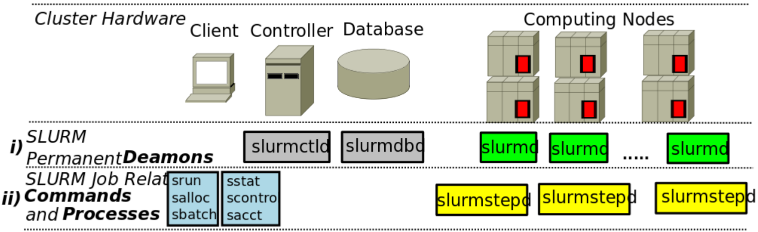
How this takes place:

- Dedicate **gy/power data** (IPI
- Dedicate energy/p
- **Power d**
- SLURM Internal power-to-energy and energy-to-power calculations

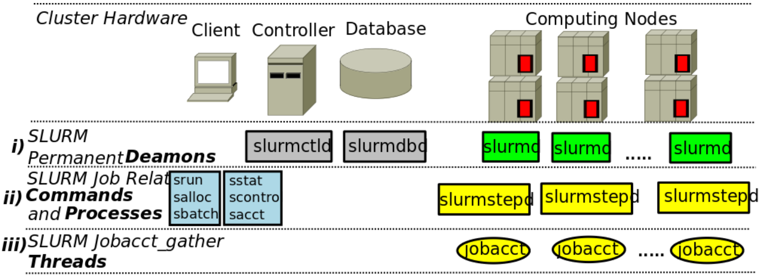
Important Issues:

- **Overhead**: In-band Collection
- **Precision**: of the measurements and internal calculations
- **Scalability**: Out-of band Collection

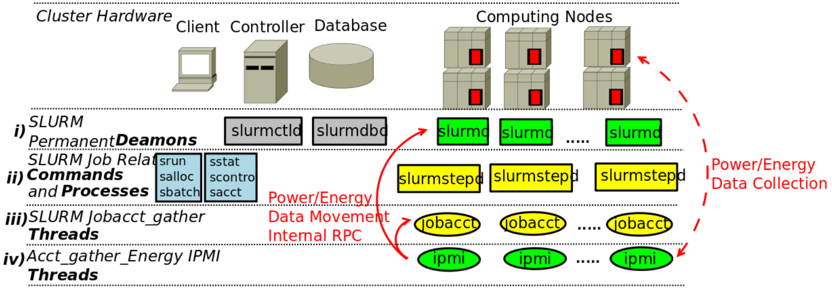
Framework Architecture



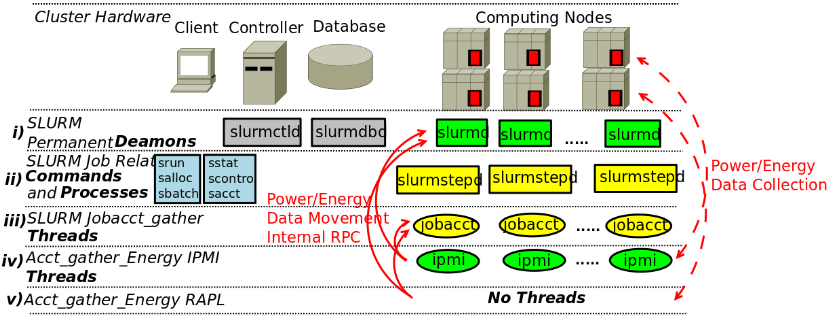
Framework Architecture



Framework Architecture

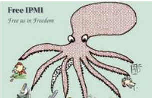


Framework Architecture



In-band collection of power/energy data with IPMI

- IPMI is a message-based, hardware-level interface specification (may operate in-band or out-of-band)
- Communication with the Baseboard Management Controller **BMC** which is a specialized microcontroller embedded on the motherboard of a computer
- SLURM support is based on the **FreeIPMI** API `http://www.gnu.org/software/freeipmi/`
  - FreeIPMI includes a userspace driver that works on most motherboards without any required driver.
  - No thread interferes with application execution
- The data collected from IPMI are currently instantaneous **measures in Watts**
- SLURM **individual polling frequency** ($>=$1sec)
  - direct usage for power profiling
  - but internal SLURM calculations for energy reporting per job



In-band collection of power/energy data with IPMI

- IPMI is a message-based, hardware-level interface specification (may operate in-band or out-of-band)
- Communication with the Baseboard Management Controller **BMC** which is a specialized microcontroller embedded on the motherboard of a computer
- SLURI
  http:
  - Fr
    m
  - N
- The da
  measu
- SLURI
  - di
  - bu

Pros - Cons

Advantages:

- Complete Node measurements

Disadvantages:

- There could be **overhead** and the read may be **time consuming**
- Complex process in **calculating energy consumption** per step/job basis using **averaging measurements window**
- No finer granularity (neither socket nor core level yet)

In-band collection of power/energy data with RAPL

- **RAPL** (Running Average Power Limit) are particular interfaces on Intel Sandy Bridge processors (and later models) implemented to provide a mechanism for keeping the processors in a particular user-specified power envelope.
- Interfaces can estimate current energy usage based on a software model driven by hardware performance counters, temperature and leakage models
  - Linux supports an 'MSR' driver and access to the register can be made through `/dev/cpu/*/msr` with priviledged read permissions
- The data collected from RAPL is energy consumption in Joules (since the last boot of the machine)
- SLURM **individual polling frequency** (>=1sec)
  - direct usage for energy reporting per job
  - but internal SLURM calculations for power reporting

In-band collection of power/energy data with RAPL

- **RAPL** (Running Average Power Limit) are particular interfaces on Intel Sandy Bridge processors (and later models) implemented to provide a mechanism for keeping the

Pros - Cons

Advantages:

- **No overhead** during capturing power/energy data (read hardware registers)
- Simple process in calculating energy consumption per step/job basis
- 2 values are sufficient (start/end of the job), no need of collecting big number of instant watts (IPMI case)

Disadvantages:

- **No whole node data**, only processor and part of memory (DRAM) is supported (no motherboard, disk, external GPU, etc)
- Per socket granularity exists (not per core yet)

Out-of-band collection of power/energy data

- External Sensors Plugins to allow out-of-band monitoring of cluster sensors
- Possibility to Capture **energy usage and temperature** of various components (nodes, **switches, rack-doors**, etc)
- Framework generic but initial **Support for RRD databases** through rrdtool API (for the collection of energy/temperature data)
  - Plugin to be used with real wattmeters or out-of-band IPMI capturing
- Power data captured used for per node power monitoring (scontrol show node) and per job energy accounting (Slurm DB)
  - direct usage for energy reporting per job
  - but internal SLURM calculations for power reporting

Out-of-band collection of power/energy data

- External Sensors Plugins to allow out-of-band monitoring of cluster sensors
- Possibility to Capture **energy usage and temperature** of variou
- Frame throug data)
  - P c
- Power (scont DB)
  - d
  - b

Pros - Cons

Advantages:

- No overhead during capturing power/energy data (out-of-band)
- Power/Energy data of various components may be used (not possible through in-band mechanisms)

Disadvantages:

- Scalability issues

Online Tutorial for more details:

`www.schedmd.com/slurmdocs/SUG13/energy_sensors.pdf`

Power profiling

- Job profiling to periodically capture the task's usage of various resources like CPU, Memory, Lustre, Infiniband and Power per node
- Resource Independent polling frequency configuration
- **Based on hdf5** file format `http://www.hdfgroup.org` open source software library
  - versatile data model that can represent very complex data objects and a wide variety of metadata
  - portable file format with no limit on the number or size of data objects stored
- Profiling per node (one hdf5 file per job on each node)
- Aggregation on one hdf5 file per job (after job termination)
- Slurm built-in tools for extraction of hdf5 profiling data

slurm
workload manager

The HDF Group

Power profiling

- Job profiling to periodically capture the task's usage of various resources like CPU, Memory, Lustre, Infiniband and Power per node
- Resource Independent polling frequency configuration
- **Based on hdf5** file format `http://www.hdfgroup.org` open source software library
  - versatile data model that can represent very complex data objects and a wide variety of metadata
  - portable file format with no limit on the number or size of data objects stored
- Profiling per node (one hdf5 file per job on each node)
- Aggregation on one hdf5 file per job (after job termination)
- Slurm built-in tools for extraction of hdf5 profiling data

Online Tutorial for more details:

`www.schedmd.com/slurmdocs/SUG13/profile_hdf5.pdf`

Job Energy Control through CPU Frequency Setting

- Support of kind of DVFS technique through **CPU Frequency setting**
- **Static** since it may not be changed during the execution (user does not usually have root access on the computing nodes)
- The user may ask either a **particular value in kilohertz** or use low/medium/high and the request will match the closest possible numerical value
- Implementation based on tasks confinement to those cpus (cgroups or cpusets).
- Implemented through manipulation of the `/sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq` and governors drivers

Issues regarding the Framework

- **Overhead:** In-band monitoring on computing nodes. **What is the overhead of the framework?**
- **Precision:** Usage of built-in models/interfaces and internal calculations. **How precise are the reported energy and power data?**
- **Scalability:** Out-of-band monitoring of computing nodes. **What is the scalability of the mechanism?** *Not treated here*

Issues regarding the Framework

- **Overhead:** In-band monitoring on computing nodes. **What is the overhead of the framework?**
- **Precision:** Usage of built-in models/interfaces and internal calculations. **How precise are the reported energy and power data?**
- **Scalability:** Out-of-band monitoring of computing nodes. **What is the scalability of the mechanism?** *Not treated here*

Overhead:

- in terms of the following resources which are shared with application [a]:
  - CPU
  - Memory
  - Energy

[a] Network excluded because in most cases at least 2 network cards are used in HPC clusters and Slurm RPCs pass from the Admin network

Precision:

- Compared to External Wattmeters measurements

Plan

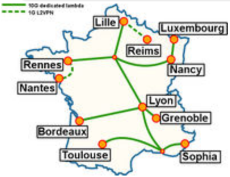
Experimental Platform

Hardware

- Grid5000 platform resources
- Lyon clusters orion and taurus
- 17 nodes cluster (DELL PowerEdge R720 with Intel Xeon E5-2630 2.3GHz 2 sockets per node /6 cores per socket, 32 GB of Memory and 10 Gigabit Ethernet Network.)

Software

- SLURM 2.6.0 with 1 slurmctld and 16 slurmd
- Execution of HPL Linpack
  - with 80% of memory
  - Duration 44min



Grid'5000

Experimental Methodology

Execution of the same job with each different monitoring mode (with polling frequency $= 1 sec$)

- NO_JOBACCT
- JOBACCT_0_RAPL
- JOBACCT_RAPL
- JOBACCT_RAPL_PROFILE
- JOBACCT_0_IPMI
- JOBACCT_IPMI
- JOBACCT_IPMI_PROFILE

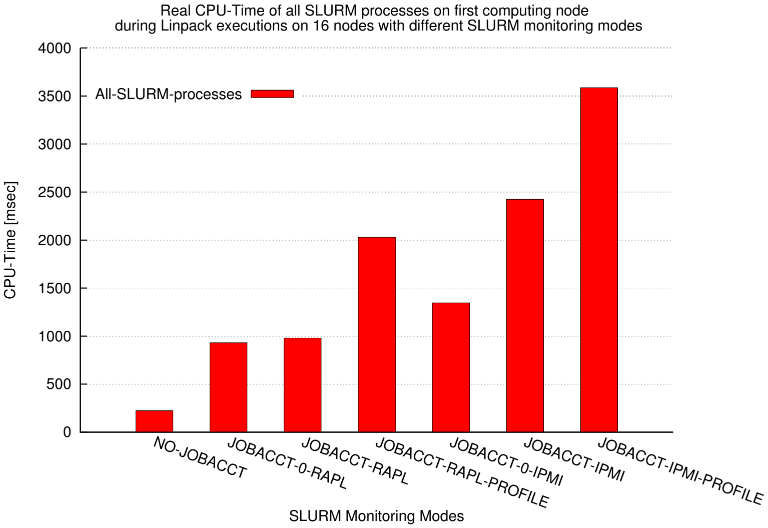
Profile the usage of:

- **CPU**: (cgroups/cpuacct subsystem)
- **Memory**: (RSS of ps command)

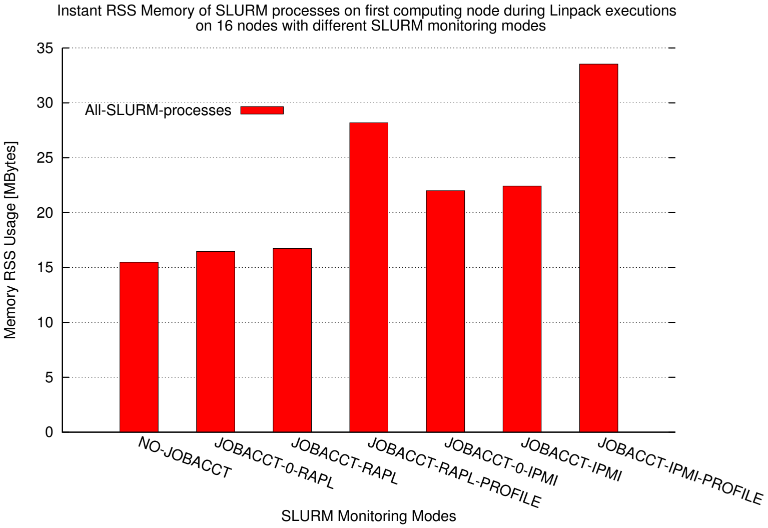
Overhead in terms of:

- **Execution Time**
- **Energy Consumption**

Framework CPU Overhead



Framework Memory Overhead



Monitoring Modes Comparison

Monitoring Modes	Time (s)	Energy (J)	Time Overhead	Energy Overhead
NO_JOBACCT	2657	12623276.73	-	-
JOBACCT_0_RAPL	2658	12634455.87	0.04%	0.09%
JOBACCT_RAPL	2658	12645455.87	0.04%	0.18%
JOBACCT_RAPL_PROFILE	2658	12656320.47	0.04%	0.26%
JOBACCT_0_IPMI	2658	12649197.41	0.04%	0.2%
JOBACCT_IPMI	2659	12674820.52	0.07%	0.41%
JOBACCT_IPMI_PROFILE	2661	12692382.01	**0.15 %**	**0.54%**

Plan

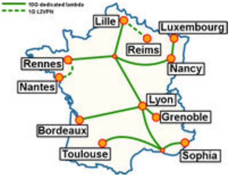
Experimental Platform

Hardware

- Grid5000 platform resources
- 17 nodes cluster (DELL PowerEdge R720 with Intel Xeon E5-2630 2.3GHz 2 sockets per node /6 cores per socket, 32 GB of Memory and 10 Gigabit Ethernet Network.)
- Intergrated Wattmeters on all computing nodes

Software

- SLURM 2.6.0 with 1 slurmctld and 16 slurmd
- Execution of HPL Linpack and Stream benchmarks



Grid'5000

Experimental Methodology

Interchanging the following parameters:

- Execution of **Long running** (Linpack) and **Short running** (Stream) jobs
- Testing all possible CPU Frequencies
- SLURM Monitoring through **RAPL** or **IPMI** with **profiling** (HDF5) activated on both cases

Evaluate the:

- Precision of job's **overall energy** calculation for IPMI case (with polling frequency =1sec)
  - Comparison of the SLURM Reported DB value with the Wattmeters calculated value
- Precision of job's **instant power profiling** for IPMI and RAPL case (with polling frequency =1sec)
  - Comparison of the SLURM profiling (HDF5) date with the Wattmeters data

Plan

Precision of job's energy calculation with IPMI

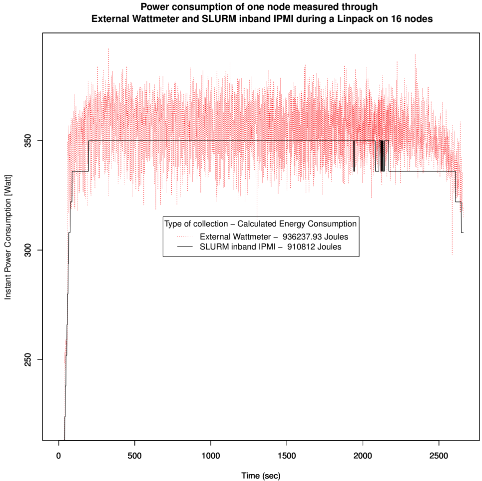
Monitoring Modes / CPU-Frequencies	2.301	2.2	2.0	1.8	1.4	1.2
External Wattmeter Postreatment Value	12754247.9	12106046.58	12034150.98	12086545.51	12989792.06	13932355.15
SLURM IPMI Reported Accounting Value	12708696	12116440	11998483	12093060	13107353	14015043
Error Deviation	0.35%	0.08%	0.29%	0.05%	0.89%	0.58%

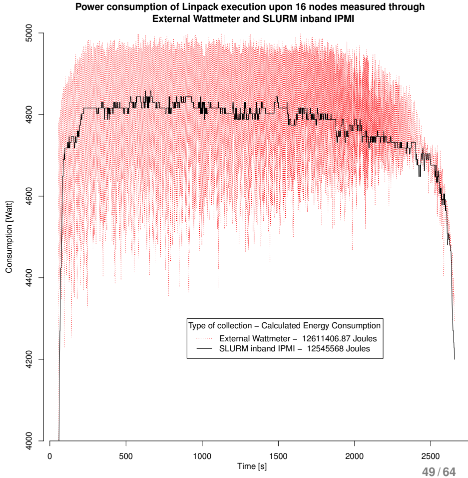
Table: SLURM IPMI precision in Accounting for Linpack executions

Precision of power profiling with IPMI

Graphs featuring:

- first computing node (left)
- all computing nodes (right)

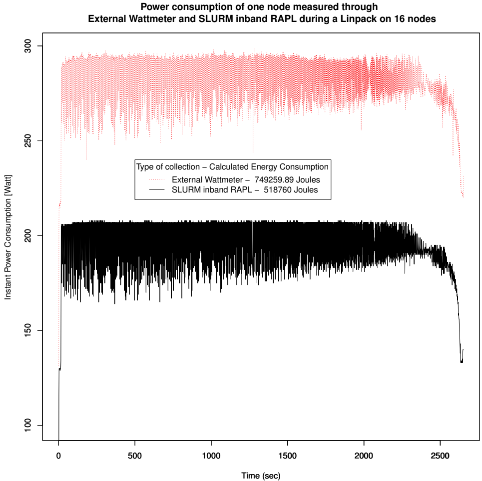


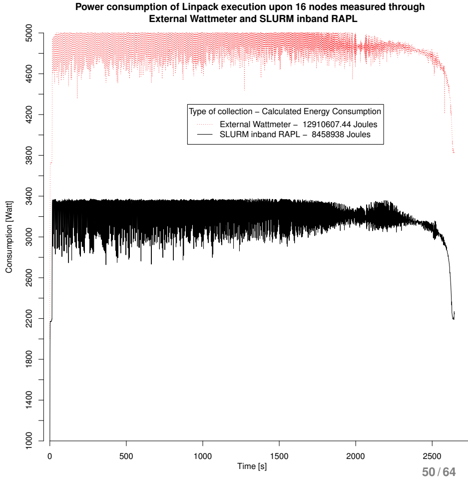


Precision of power profiling with RAPL

Graphs featuring:

- first computing node (left)
- all computing nodes (right)





Plan

Precision of job's energy calculation with IPMI

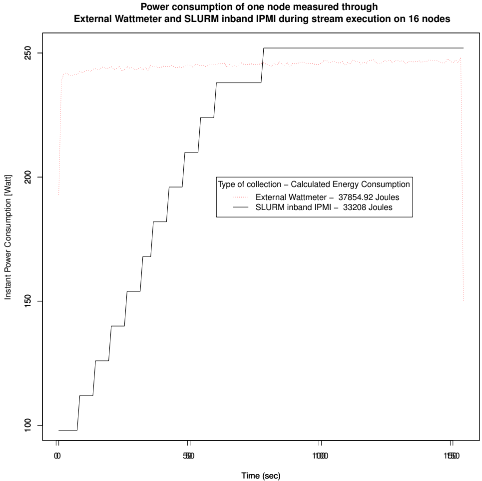
Monitoring Modes / CPU-Frequencies	2.301	2.2	2.0	1.8	1.4	1.2
External Wattmeter Postreatment Value	627555.9	567502.72	540666.46	518834.96	504505.76	529519.07
SLURM IPMI Reported Accounting Value	544740	500283	482566	467719	460005	497637
Error Deviation	13.19%	11.84%	10.74%	9.85%	8.82%	6.02 %

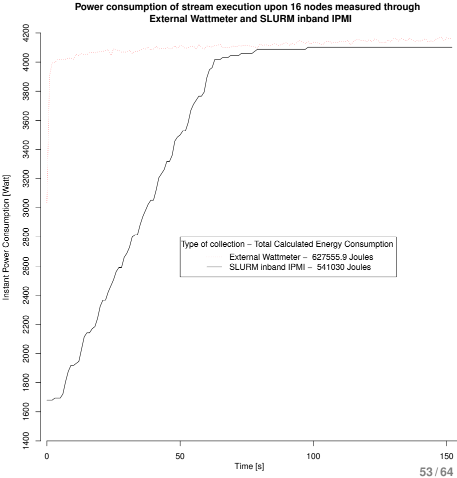
Table: SLURM IPMI precision in Accounting for stream executions

Precision of power profiling with IPMI

Graphs featuring:

- first computing node (left)
- all computing nodes (right)

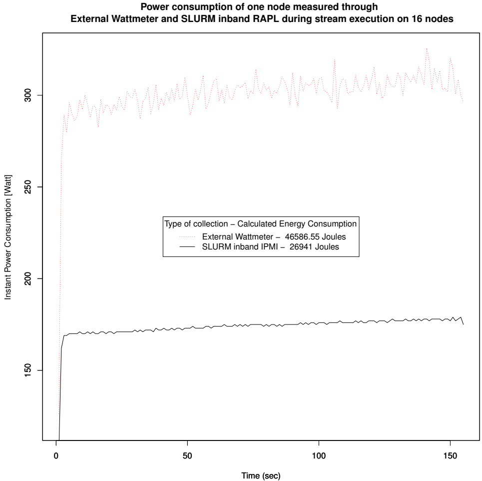


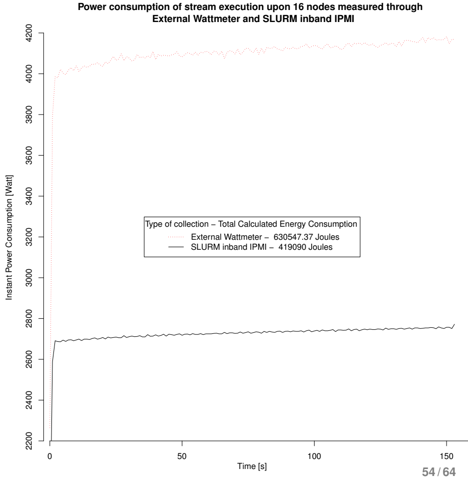


Precision of power profiling with RAPL

Graphs featuring:

- first computing node (left)
- all computing nodes (right)





Error Deviations Explanations

Setup on 2 different hardwares

- **DELL PowerEdge R720** and **BULL B710** nodes (12 cores, 32GB Memory each)
- with **different BMC** models

Software

- SLURM 2.6.0 with 1 slurmctld and 1 slurmd each
- Execution of simple multi-threaded prime number calculation

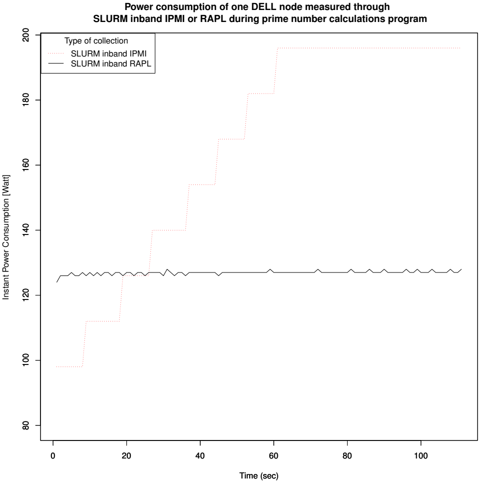
Observe and Compare:

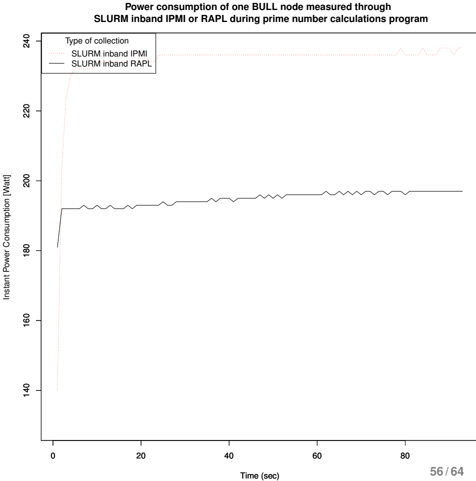
- **Precision** of SLURM RAPL and IPMI monitoring

Error Deviations Explanations

Graphs featuring different BMC models:

- DELL node (left)
- BULL node (right)

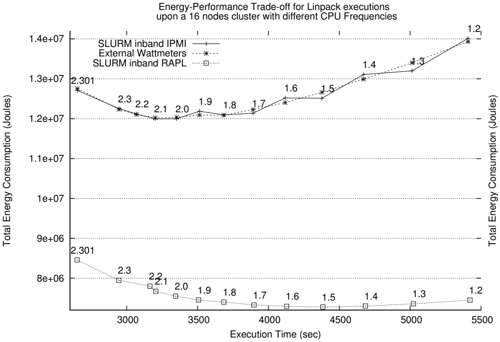


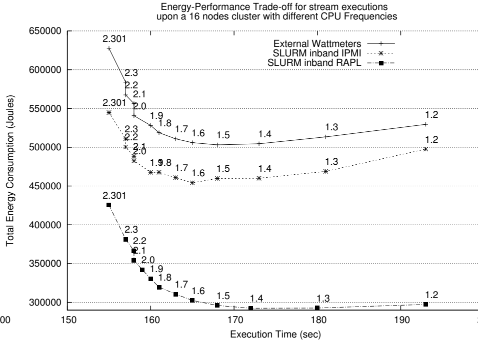


Plan

Usage of Energy Accounting values

- Profiling Performance and Energy based on CPU Frequencies
- Graphs featuring Linpack(left) and Stream (right) executions





Plan

Conclusions

Framework Overhead:

- Less than 0.6% of energy consumption
- Less than 0.2% of execution time
- Safe to use on production environment with $>=$ 1sec sampling frequency

Job's energy consumption precision (SLURM Energy Accounting):

- Good results for **long running jobs**
- Not good results **with short running** jobs because of BMC issues, newer BMC models show improved behaviour
- Jobs with **regular power variations** that can cause **strong aliasing effects** are not considered and will be treated in following studies

Power profiling precision:

- IPMI: Correct average values but poor sensitivity (BMC responsible)
- RAPL: Not complete node values but very good sensitivity

Current State

- Framework Design and evaluations appear in proc of *[ICDCN-2014]* [a]
- Developments appeared in Slurm-2.6.0 stable version, on July 2013
- Today **used in production** on *MeteoFrance* supercomputer (1080*nodes*) and *TU-Dresden* cluster (600*nodes*) (IPMI version with profiling activated)
- Energy Accounting features in SLURM enable institutions to **charge users for energy consumption** (besides CPU-Time).
- Control of Energy Consumption through the selection of the best frequency enables energy reductions with 100% system utilization.
- Workload trace files (SWF) with energy extracted from supercomputers in production to be used as heuristics for research in scheduling

---

[a] *Y. Georgiou et al. Energy Accounting and Control with SLURM Resource and Job Management System in M. Chatterjee et al.: ICDCN 2014, LNCS 8314, pp. 96–118, Springer-Verlag*

SWF trace with Consumed Energy field

```
JobID Submit Wait Elapsed CPUs CPUTime Mem ...ConsumedEnergy
3541439 0 271 67 1 59 374756 ......................  20960
3541440 5 266 50 1 41 356628 ......................  12150
```

Ongoing Works

Power/Energy Monitoring

- IPMI and RAPL should report their results in the same time, to deduce the evolution of the consumption of other parts of nodes (motherboard, network cards, etc)
- Finer-grained monitoring of various node components will help us to better characterize the usage effectiveness of nodes.

Power/Energy Measurements Precision

To deal with the worst case of jobs with **regular power variations**:

- Development of new BULL specific BMC firmware to provide **internal energy** consumption **calculations** with **higher precision** and **less overhead** (Proprietary)
  - Technique based on start/get/stop technique for BMC energy calculation
  - 4Hz internal polling sampling
- **Development of Slurm plugin** to support this new BULL BMC firmware through ipmi raw data collection with FreeIPMI
- New mechanisms will be deployed on TUDresden on December 2013

Future Works

Energy: a new type of resource

- Treating **Energy as a new job characteristic** opens new doors to treat it as a new resource.
  - **Energy fairsharing** will keep the fairness of energy distribution amongst users.
- Explore other techniques for Control of Energy Consumption during job execution (RAPL power-capping, GPU frequency setting, Network cards power off, etc)
- **Energy Aware Scheduling**: Optimize system utilization with respect to energy availability
  - **Power capping** techniques reflecting the evolutions of the amount of available power over time

Architect of an Open World™