This is the post peer-review accepted manuscript of:

Andrea Bartolini, Andrea Borghesi, Antonio Libri, Francesco Beneventi, Daniele Gregori, Simone Tinti, Cosimo Gianfreda, and Piero Altoè. 2018. The D.A.V.I.D.E. big-data-powered fine-grain power and performance monitoring support. In Proceedings of the 15th ACM International Conference on Computing Frontiers (CF '18). ACM, New York, NY, USA, 303-308. DOI: https://doi.org/10.1145/3203217.3205863

The published version is available online at: https://dl.acm.org/citation.cfm?id=3205863

The D.A.V.I.D.E. Big-Data-Powered Fine-Grain Power and Performance Monitoring Support

Andrea Bartolini [1], Andrea Borghesi[1], Antonio Libri [2], Francesco Beneventi[1], Daniele Gregori[3], Simone Tinti[3], Cosimo Gianfreda[3], Piero Altoè[4]
[1] DEI, University of Bologna, Bologna, Italy.
[2] IIS, D-ITET, ETH, Zurich, Switzerland.
[3] E4 Computer Engineering, Scandiano (RE), Italy.
[4] NVIDIA Corporate
a.bartolini@unibo.it,andrea.borghesi@unibo.it,a.libri@iis.ee.ethz.ch,francesco.beneventi@unibo.it,daniele.gregori@e4company.com,simone.tinti@e4company.com,cosimo.gianfreda@e4company.com,paltoe@nvidia.com

ABSTRACT

On the race toward exascale supercomputing systems are facing important challenges which limit the efficiency of the system. Among all, power and energy consumption fueled by the end of Dennard's scaling start to show their impact on limiting supercomputers peak performance and cost effectiveness.

In this paper we present and describe a new methodology based on a set of HW and SW extensions for fine-grain monitoring of power and aggregation of them for fast analysis and visualization. We propose a turn-key system which uses MQTT communication layer, NoSQL database, fine grain monitoring and in future AI technology to measure and control power and performance. This methodology is shown as an integrated feature of the D.A.V.I.D.E. supercomputing machine.

KEYWORDS

Big Data, High Performance Computing, Fine-Grain Power and Performance Monitoring, AMESTER, BeagleBone Black

1 INTRODUCTION

On the race toward exascale performance, supercomputers have become power and energy limited. Until June 2016, every new most powerful supercomputer in the world (1st in the Top500 list[1]) has marked an increasing in its power consumption. In 2013 Tianhe-2 reached 17.8 MW of IT peak power consumption, which becomes 24 MW when considering also the cooling power[2]. This has set the record for the power consumption of a single supercomputer installation, reaching the practical limit in power provisioning of 20 MW. Today's most powerful supercomputer (TaihuLight) consumes 15.4 MW underlining the fact that performance increase is nowadays possible only at a fix power budget: as a matter of fact supercomputers are power limited.

The Green500 list ranks the 500 most powerful supercomputers by an energy-efficiency metric, measured as Flops/W (Floating Point Operations per Second per Watt) [3]. From the Green500 perspective Sunway TaihuLight is ranked 20th with an energy-efficiency of 6 GFlops/W, while Tianhe-2 is ranked 137th with an energy-efficiency of 2 GFlops/W. Today, thirteen of the twenty most energy efficient supercomputers in the world use an heterogeneous design based on the NVIDIA Tesla P100 [4] accelerator card, which can reach up to 13.7 TFlops in double precision with 300 W of thermal design power (TDP).

To push the boundary of efficient computing in addition to best-in-class hardware components, supercomputing systems require to inspect and control their own power consumption under different applications workloads. Moreover, fast monitoring sampling frequencies are crucial to correlate energy and power measurements with application phases [5]. This must be achieved while seamlessly monitoring the performance metrics of the computing nodes that are executing the application. At the same time optimizing the entire supercomputer power consumption requires the aggregation in time and space of a large set ofinformation coming from job scheduler, user requirements, and computing resources. Clearly, doing this online and at large scales, without being intrusive and causing performance loss, is problematic.

In this paper we present a novel approach toward performance and power monitoring of supercomputers. Our solution combines hardware extensions on the node architecture for fine grain power measurements, with a scalable data collection backbone based on best-in class open-source big data framework and a lightweight communication protocol. This approach is implemented as part of the D.A.V.I.D.E. (Development for an Added Value Infrastructure Designed in Europe) computing system [6].

The paper is organized as follows. Section 2 outlines prior works. Section 3 presents D.A.V.I.D.E. Section 4 describes the power monitoring extensions. Finally, Section 5 presents the experimental results.

2 RELATED WORKS

Today, a per-node power profile can be obtained by the node BMC (Baseboard Management Controller) via IPMI interface. However, this mechanism is characterized by a slow sampling rate (seconds), no time-stamping, and does not allow an accurate energy accounting [7].

To overcome these problems, [7] proposes HDEEM, which allows power sampling up to 1 KS/s (kilo Samples per second) and accurate energy accounting, thanks to an extension of the BMC data monitoring features and a dedicated FPGA placed on each computing node. However, due to the use of the BMC as embedded monitoring system, their solution is not open and flexible to implementing new algorithms for on-board data processing, suffers from closed design, and it is limited in memory storage. Moreover, instantaneous readings are possible only at 1 S/s.

Works in [8] and [9] are based on low cost open hardware embedded computers (Arduino Mega 2560 and the Beaglebone Black - BBB - respectively) to read and collect power measurements. As [7], both approaches provide sampling rate up to few kilo samples per second. However, the measurements are provided via custom interfaces, which cannot be easily integrated with other system components. Moreover, [8] is unfeasible to be used in a large scale HPC infrastructure.

3 D.A.V.I.D.E.

D.A.V.I.D.E. (Development for an Added Value Infrastructure Designed in Europe) [6] is an Energy Aware PetaFlops Class High Performance Cluster, based on Power Architecture and coupled with NVIDIA Tesla Pascal GPUs with NVLink. The innovative design of D.A.V.I.D.E. is based on OpenPOWER platform and is among the harbingers of a new generation of HPC systems which deliver high performance while being environmentally conscious. It is built using best-in-class components plus custom hardware and an innovative middleware system software.

D.A.V.I.D.E. is composed by 45 nodes connected with an efficient Infiniband EDR 100 Gbytes networking, with a total peak performance of 990 TFlops and an estimated power consumption ofless than 2 Kwatt per node. Each node is a 2 Open Unit (OU) Open Compute Project (OCP) form factor and hosts two IBM POWER8 Processors with NVIDIA NVLink and four Tesla P100 data center GPUs, with the intra-node communication layout optimized for best performance.

The system is ranked #440 in TOP500 and #18 in GREEN500 in the November 2017 list.

Following a short description of the main system elements.

The Compute Node is derived from IBM POWER8 System S822LC, with two IBM POWER8 with NVlink and four NVIDIA Tesla P100 HSXM direct liquid cooled. It uses Open Rack Enclosure with integrated piping and power distribution and each node is modified to fit the OCP form-factor. Each compute node has a peak performance of 22TFlops and a power consumption ofless than 2 KW. Each node hosts four NVIDIA Tesla P100 HSMX2, with NVLink interconnect capable of 5.3 TFlops of double precision floating point (FP64) performance, 10.6 TFlops of single precision (FP32) performance, 21.2 TFlops of halfâĂŘprecision (FP16) performance. The NVLink implementation in NVIDIA Tesla P100 supports up to four links, enabling ganged configurations with aggregate maximum bidirectional bandwidth of 160 Gbyte/s (A single link supports up to 40 Gbyte/s of bidirectional bandwidth.).

The Liquid Cooling is based on direct hot-water cooling (27°C) for CPUs and GPUs. Each rack has an independent liquid/liquid or liquid/air heat exchanger unit with redundant pumps. The system has internal pumps on GPUs. Each Rack has its Coolant Distribution Unit (CDU). Finally the compute nodes are connected to the heat exchanger through pipes and a side bar for water distribution.

A key feature of D.A.V.I.D.E. is an innovative technology to further improve energy efficiency, which consists of measuring, monitoring and capping the power consumption of each node and of the whole system, using data from several components (processors, memory, GPUs, fans). This is described in the next section.

4 POWER MONITORING EXTENSIONS

The Power monitoring extensions consist of a set of agents running outside the computing components of the nodes, but tightly coupled with them. These agents monitor the power consumption of each computing node at the plug as well as performance and utilization metrics. The monitored values are exchanged to a data management backbone, through a communication layer based on the open-source MQTT (MQ Telemetry Transport) protocol.

4.1 The Communication Layer

The power management and monitoring framework takes advantage of the MQTT protocol which implements the publish-subscribe messaging pattern. MQTT required three different agents to work: (i) The "publisher", which sends data on a specific âĂIJtopicâĂİ. (ii) The "subscriber", which subscribes to appropriate topics. (iii) The "broker", which (a) receives data from publishers, (b) makes topics available to subscribers, (c) delivers data to subscribers. The MQTT communication works as follow. The publisher agent sends some data with a certain topic as a protocol parameter, the topic is generated and available to the broker. Any subscriber listening to that topic will receive the associated data as soon as the broker receives them. Collector agents have the role of "publishers" in this scenario.

4.2 The Power Monitoring Agent

The power monitoring agents allow to measuring the power consumption of the several HPC nodes at the power source level as well as measuring its internal component performance. From the hardware point of view, they are composed by (i) a power sensing module, which contains the sensors for measuring current and voltage, and (ii) an embedded monitoring board, which sample, preprocess and send data via MQTT to the framework data collection backbone. The power sensing module is placed between the busbar and the DC-DC converters (that supply all the processing and electrical components within the node). The current is measured with a current mirror connected to a shunt resistor, while the voltage with a voltage divider. We use an open hardware platform as embedded monitoring board, namely the Beaglebone Black. The optimized software running on the BBB, exploits the built-in ADC (connected to the shunt resistor and the voltage divider) to sampling data with Watt precision at 50 kS/s, which is 50x faster than best state-of-the-art systems. The data are first pre-processed on-board (linear conversion from 12-bit integer to Ampere, Volt and Watt), and then sent to the data collection backbone on 2 MQTT topics, at 1 ms and 1 s.

In the following sections, we refer to this software component as Node Power Monitoring Daemon. Thanks to the Network Time

Protocol (NTP) and the Precision Time Protocol (PTP - which is supported in hardware on the Beaglebone Black), the power monitoring data can be synchronized up to microsecond scale [10].

For an out-of-band monitoring of the nodes performance we use the IBM Amester commands, which exploit the IPMI interface to the OpenPOWER POWER8 on-chip controller (OCC), to get OCC sensor readings. The IPMI Amester commands are sent to the OCC, through the board management controller (BMC), using a python script. The python script executes on the embedded monitoring board (BBB). We will refer to this SW component in the following sections as theOCC Daemon The received data are then sent to the MQTT backbone to be processed by Examon. To increase the spatial and time granularity (at which data are exposed to Examon), we modified the OCC firmware to implement a new set of commands. These commands use an ad-hoc internal format to increase the sampling speed up to 10x as well as integral readings (average of the metrics in between two cosecutive samples) to avoid aliasing problems. Moreover, we implemented the NTP algorithm to synchronize the OCC sensor readings with the BMC and take reliable timestamps.

4.3 The Data Collection Backbone

The monitoring framework provides a mechanism to store metrics mainly for visualization and analysis of historical data. We use a distributed and scalable time series database (KairosDB) that is built on top of a NoSQL database (Apache Cassandra) as back-end. A specific MQTT subscriber (MQTT2Kairos) is implemented to provide a bridge between the MQTT protocol and the KairosDB data insertion mechanism. The bridge leverages the particular MQTT topics structure of the monitoring framework, to automatically form the KairosDB insertion statement. This gives a twofold advantage: firstly, it lowers the computational overhead of the bridge, since it is reduced to a string parsing operation per message; secondly, it makes easy to form the database query starting only from the knowledge of the matching MQTT topic. All these services execute in containers in the frontend nodes, without stealing computing resources from the cluster. In addition to these data handling services, frontend nodes execute a plugin for collecting the power sensors data from the AMESTER sensors, which allows to extract the power consumption of the internal components. The metrics stored in the database are visualized in real time using a web-based tools, namely Grafana [11]. Grafana is an open-source project aimed to provide a web based environment oriented to build and manage general purpose monitoring dashboards. It supports many time-series databases as back-ends including KairosDB.

5 EXPERIMENTAL RESULTS

This section focuses on the experimental results of the D.A.V.I.D.E. power monitoring framework. The first set of results quantifies the load of the monitoring framework in terms of monitored and processed metrics per second, as well as the load of the computing resources running the monitoring framework. We recall that the entire monitoring framework described in the paper is out-of-band and do not executes in the computing resources. The second set of results shows the performance and quality of the monitored data in terms of visual insights in a series of Linpack runs on the 45 D.A.V.I.D.E. nodes.

Source	Total Cluster Samples/s
IPMI Sensors	801
OCC Sensors	1089
Node Power Monitoring	45k

Table 1: Number of metrics density (Samples/s) for the entire D.A.V.I.D.E cluster.

Component	RAM Usage	#Processes	CPU% (Avg/Peak)
Monitoring Node (Container)			
Cassandra	12GB	171	25/300
Kairosdb	4GB	117	190/600
Data Collection	4GB	470	450/1200
Grafana	512MB	52	0.01/0.02
Monitoring Agent (BeagleBone Black)			
Node Power Monitoring Daemon		3	39/41
OCC Daemon		1	11/22

Table 2: Resource usage for the monitoring framework (Monitoring Node and Agents). The monitoring node containers are executed in the management node. The monitoring agents are executed on each Beaglebone Black placed on each node. The CPU% represents the sum of the core usage in percentage.

5.1 Monitoring Framework Load

Table 1 reports the volume of metrics monitored in real-time on the entire cluster and received by the data collection backbone. From the table we can notice that the gross of the collected metrics comes from the Node Power Monitoring daemon and its fine-grain measurements. Indeed, the monitoring framework reports the power consumption of each node continuously at 1 KS/s speed. In addition the system monitors 89 IPMI metrics per node every 5s, and 242 metrics per node every 10s (e.g. core, memory buffer, DIMM performance, physical metrics, etc.). As reported in Table 1, this corresponds to an overall volume of 801 and 1089 Samples/s, for IPMI and OCC, respectively.

Table 2 reports the resource usage for the monitoring framework. The monitoring node container (running on the management node) involves Cassandra, KairosDB, Data Collection (Broker, data pre-processing, IPMI publisher) and Grafana. The Monitoring Agent (running on the BeagleBone Black) involves the node power monitoring daemon, which reads and transmits the node power consumption at 1s and 1ms, and the OCC Daemon which reads all the sensors from the OCC of the computing nodes.

The table shows the Beaglebone Black has enough resources for taking care of reading and publishing to the MQTT broker the node power consumption at fine granularity as well as reading and publishing all the OCC metrics for the node. Moreover, the management node, which features 16 physical POWER8 cores, has enough power to collect all the monitored data in real-time, preprocess them, keep a historical buffer of all the monitored metrics,

and visualize them. As the Table 2 shows, the hardware resources needed by the monitoring framework are limited on average to 7 cores and 21 GBs of memory.

5.2 Monitoring Framework Visual Insights

In this section we will report a set of snapshot of the Grafana front end, which is used to visualize the collected data during a set of Linpack runs. This helps to underline the importance and role of the different power monitoring agents: The Node Power Consumption Daemon, and OCC and IPMI Daemon.

Figure 1 reports the power consumption of all D.A.V.I.D.E. nodes, measured through the values transmitted by the Node Power Monitoring Daemon, running on each power sensing module. These values transmitted at 1 Hz are collected by the data monitoring backbone and visualized via Grafana. The top plot reports all the node power consumption stacked. This allows to visualize each node contribution to the total node power consumption. The bottom plot instead reports the power consumption for each node overlapped. From the first plot we can conclude that the total power consumption has significant power oscillations (above 30%). From the bottom plot we can see that this is due by synchronous workload phases on the different nodes. Someone could argue that 1 s is already a good temporal resolution for node level power consumption, as it already allows to see important patterns on the node and cluster power consumption.

The answer is in Figure 2, which reports the power consumption measured at 1 s and at 1 ms for all the D.A.V.I.D.E. nodes for a shorten time windows. In particular, the top plot shows a time window of 4 seconds, while the bottom plot a time window of 1 s, only.

From these two plots, we can see that the node power consumption at 1 ms unveils additional workload phases with more than 25% (500 W) of the total power variations in few milliseconds. As a matter offact, the deployed monitoring framework allows to easily inspect the power consumption oflarge scale systems up to millisecond scales.

Finally, Figure 3 shows the capability of the proposed monitoring framework in terms of correlating the node power consumption measurements with performance and architectural metrics, which are continuously acquired. The top plot reports (for the same time window of previous plots) the power consumption of the main different components in a single node of D.A.V.I.D.E. (davide37 - Fans:PWR_FAN, Memory:PWR_MEM, Cpus:PWR_P0, GPUs:PWR_GPUs). These metrics are collected from the IPMI sensors, and sent to the data collection backbone by the IPMI Daemon running on the management node. We can notice from it that the main power variations are caused by the GPUs power consumption (PWR_GPU metric). The bottom plot reports the overall node power consumption measured by the Node Power Monitoring Daemon in the BBB at both 1 s and 1 ms. It is visible that the power consumption obtained from the single components (top plot) follows the one reported by the Node Power Monitoring Daemon. Further works will study the direct correlation with application phases of the fine grain power measurements.

6 FUTURE WORKS

In future works on D.A.V.I.D.E. we will expand the current infrastructure for artificial intelligence supported management of the cluster. This will be done by exploiting the depicted integrated monitoring system with deep learning and edge-computing targeting predictive maintenance and power management.

7 CONCLUSION

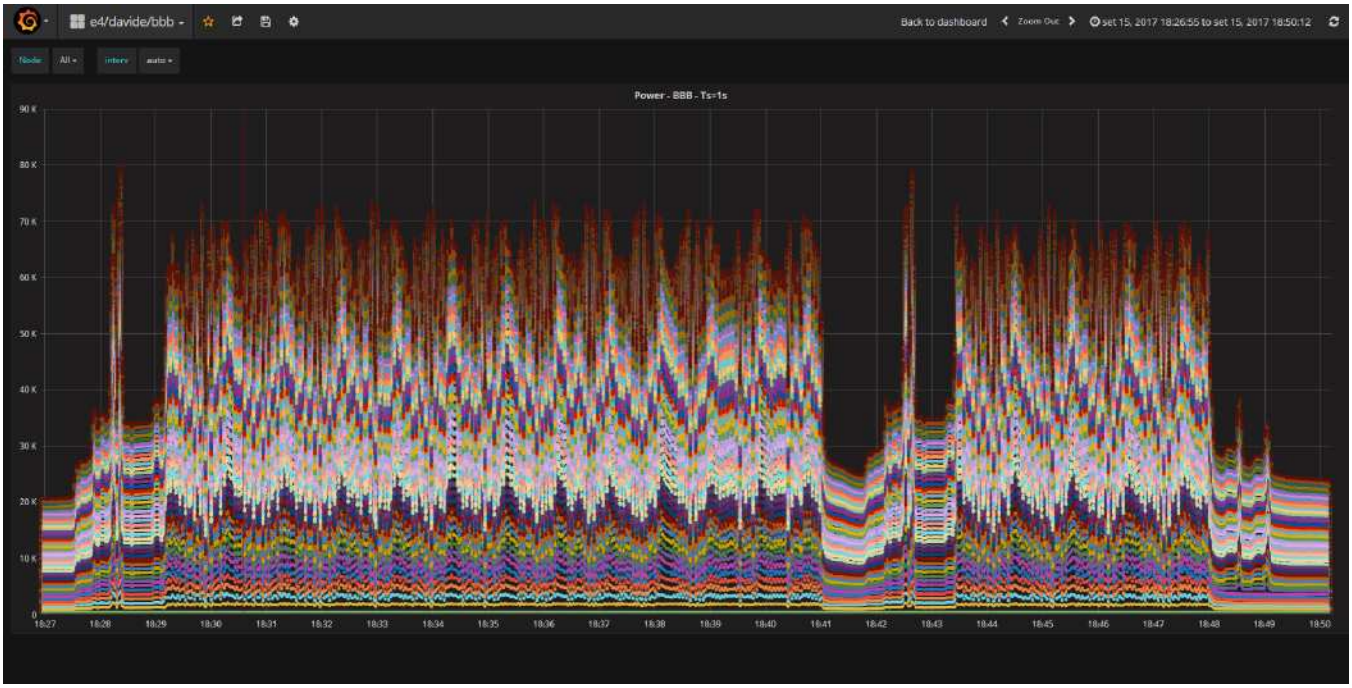
In this paper we presented a novel power and performance monitoring framework integrated inside a 45 nodes supercomputer based on IBM OpenPOWER and NVIDIA pascal accelerators. The proposed approach is completely out-of-band and capable of measuring, process and store the power consumption. We achieve that by using custom software and hardware modifications on the node and cluster architecture. The proposed approach is capable of uncover significantly power variation at the cluster and node level down up to millisecond scale. This is done with no impact on the computing resource availability as the power monitoring is carried out outside the computing nodes of the cluster.

ACKNOWLEDGMENTS

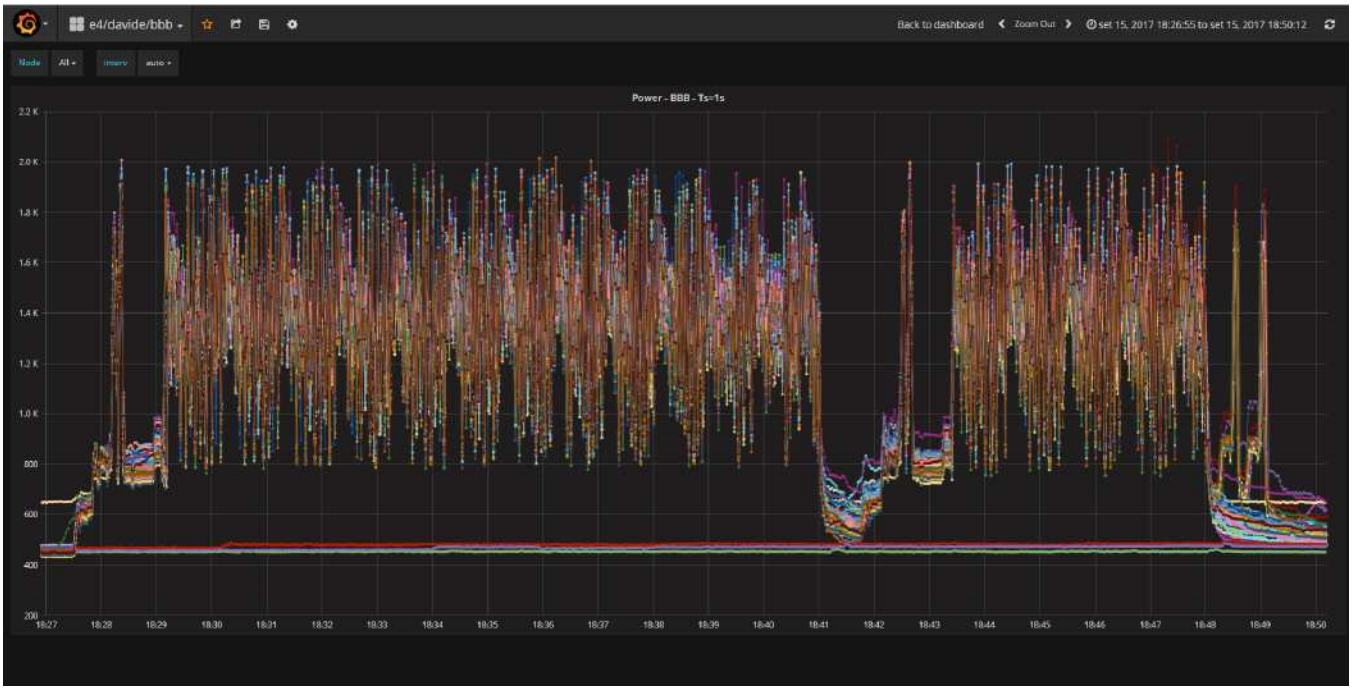
This work was partially supported by the EU FETHPC project ANTAREX (g.a. 671623), EU project ExaNoDe (g.a. 671578), EU ERC Project MULTITHERMAN (g.a. 291125) and commissioned by CINECA acting on its own behalf and on behalf of CSC - Tieeteen Tietotekniikan Keskus Oy, EPCC (University of Edinburgh), Forschungszentrum Julich GmbH and GENCI. The views expressed in this publication are those of the author(s) and not necessarily those of the aforementioned entities.

REFERENCES

[1] Jack J Dongarra, Hans W Meuer, Erich Strohmaier, et al. Top500 supercomputer sites. Supercomputer, 11:133–133, 1995.
[2] Jack Dongarra. Visit to the national university for defense technology changsha. China, University of Tennessee, 199, 2013.
[3] Green500 List https://www.top500.org/green500 [last accessed: Sep 16, 2017].
[4] Ryan Smith. Nvidia updates gpu roadmap; unveils pascal architecture for 2016, 2014.
[5] Thomas Ilsche, Daniel Hackenberg, Stefan Graul, Robert Schöne, and Joseph Schuchart. Power measurements for compute nodes: improving sampling rates, granularity and accuracy. In Green Computing Conference and Sustainable Computing Conference (IGSC), 2015 Sixth International, pages 1–8. IEEE, 2015.
[6] W. Abu Ahmad, A. Bartolini, F. Beneventi, L. Benini, A. Borghesi, M. Cicala, P. Forestieri, C. Gianfreda, D. Gregori, A. Libri, F. Spiga, and S. Tinti. Design of an energy aware petaflops class high performance cluster based on power architecture. In 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) pages 964–973, May 2017.
[7] D. Hackenberg, T. Ilsche, J. Schuchart, R. Schöne, W. E. Nagel, M. Simon, and Y. Georgiou. Hdeem: High definition energy efficiency monitoring. In Energy *Efficient Supercomputing Workshop (E2SC), 2014*, pages 1–10, Nov 2014.
[8] M. F. Dolz, M. R. Heidari, M. Kuhn, T. Ludwig, and G. Fabregat. Ardupower: A low-cost wattmeter to improve energy efficiency of hpc applications. In Green Computing Conference and Sustainable Computing Conference (IGSC), 2015 Sixth International, pages 1–8, Dec 2015.
[9] J. H. Laros, P. Pokorny, and D. DeBonis. Powerinsight - a commodity power measurement capability. In Green Computing Conference (IGCC), 2013 International, pages 1–6, June 2013.
[10] A. Libri, A. Bartolini, M. Magno, and L. Benini. Evaluation of synchronization protocols for fine-grain hpc sensor data time-stamping and collection. In 2016 International Conference on High Performance Computing Simulation (HPCS) pages 818–825, July 2016.
[11] Francesco Beneventi, Andrea Bartolini, Carlo Cavazzoni, and Luca Benini. Continuous learning of hpc infrastructure models using big data analytics and in-memory processing tools. In Proceedings of the Conference on Design, Automation & Test in Europe, pages 1038–1043. European Design and Automation Association, 2017.



(a) Stacked power plot. One line for each node power consumption. Absolute value: cluster power consumption.



(b) Not Stacked power plot. One line for each node power consumption. Absolute value: node power consumption.

Figure 1: Grafana Snapshot on a time window of 23 minutes (18:27-18:50). Nodes power consumption measured @1s by the Node Power Monitoring Daemon

(a) All cluster nodes power monitoring daemon. Zoom of 4 seconds. (Top) Topic at 1 s. (Bottom) Topic at 1 ms.

(b) All cluster nodes power monitoring daemon. Zoom of 1 seconds. Topic at 1 ms, only.

Figure 2: Grafana Snapshot on time windows of 4 seconds (18:30:40-18:30:44) and 1 seconds (18:30:41-18:30:42). Zoom and comparison in between 1 s and 1 ms readings.For each subplot: top plot1 s; bottom plot1 ms.

Figure 3: Grafana Snapshot on time windows of 15 minutes (18:27-18:42). OCC and IPMI Daemon per component power measurement (Top Plot) vs. Node Power Consumption Daemon (Bottom Plot).