DiG: Enabling Out-of-Band Scalable High-Resolution Monitoring for Data-Center Analytics, Automation and Control

Conference Paper

Author(s):
Libri, Antonio; Bartolini, Andrea; Benini, Luca

Publication date:
2018

Permanent link:
https://doi.org/10.3929/ethz-b-000306925

DiG: Enabling Out-of-Band Scalable High-Resolution Monitoring for Data-Center Analytics, Automation and Control

Antonio Libri
D-ITET, ETH Zurich
Zurich, Switzerland
a.libri@iis.ee.ethz.ch

Andrea Bartolini
DEI, University of Bologna
Bologna, Italy
a.bartolini@unibo.it

Luca Benini
D-ITET, ETH Zurich
Zurich, Switzerland
lbenini@iis.ee.ethz.ch

***Abstract*—Data centers are increasing in size and complexity, and we need scalable approaches to support their automated analysis and control. Performance, power consumption and reliability are their key "vital signs". State-of-the-Art monitoring systems provide built-in tools to collect performance measurements, and custom solutions to get insight on their power consumption. However, with the increase in measurement resolution (in time and space) and the ensuing huge amount of measurement data to handle, new challenges arise, such as bottlenecks on the network bandwidth, storage and software overhead on the monitoring units. To face these challenges we propose a novel monitoring platform for data centers, which enables real-time high-resolution profiling (*i.e.*, all available performance counters and the entire signal bandwidth of the power consumption at the plug - sampling up to 20 µs) and analytics, both on the edge (node-level analysis) and on a centralized unit (cluster-level analysis). The monitoring infrastructure is completely out-of-band, scalable, technology agnostic and low cost.**

***Index Terms*—Data centers, HPC, High-Resolution Monitoring, Edge Analytics, Machine Learning, Deep Neural Networks**

I. Introduction

Scientific computing centers are becoming increasingly complex and the need for novel methods to support their automation, analytics and control is garnering considerable attention [1]. In this direction, industry and academia researchers are pushing to the use of Artificial Intelligence (AI) and Machine Learning (ML) techniques to address non-trivial challenges such as efficient management of computational / infrastructure resources, detection of anomalies and failures, and predictive maintenance. As an example, Duplyakin et al. [2] shown how to get high-confidence predictions of the amount of time and energy of scientific applications (which can help for a more efficient usage of the resources) via Active Learning techniques applied to regression problems. Other examples are works based on unsupervised learning, such as [3] which show a method for real-time anomaly detection on streaming data (useful for an early warning about problems in the system and applications), and [4] that shows a way to detect malware using hardware features.

All these techniques exploit a low level monitoring of the hardware of the data-center infrastructure (*i.e.*, application and system performance, and related power and energy consumption). In particular, depending on the target use-case, some features can reveal more information than others: a highly flexible monitoring has to collect as many metrics as possible.

This work is partially funded by the EC under program grant H2020 FET-HPC ANTAREX 671623 and by E4 Computer Engineering SpA.

On the other hand, this implies to face 3 main bottlenecks related to the large amount of monitoring data produced: (i) overhead on the network's bandwidth, (ii) overhead on the data storage capacity (to save measurements for post-processing analysis) and (iii) overhead on the software tools that have to handle the measurements (in real-time and offline).

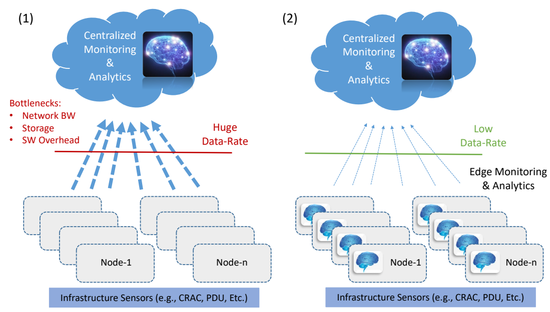

Fig. 1. Data-center monitoring design and bottlenecks.

This is depicted in Figure 1 (left), which shows that the node dedicated for the monitoring software stack has the complete view of the status of the cluster (and thus can exploit measurements for ML analysis), but has to face with such bottlenecks. To give an idea, Ilsche et al. developed a high resolution power monitoring system (*i.e.*, HAEC [5]) that supports a sampling rate of 500 kS/s (kilo Samples per second) on 4 custom sensors. In general, high resolution power monitoring instrumentation is the current trend for both industrial and academic High Performance Computing (HPC) facilities and data centers [5]–[7]. This is useful to appreciate the power consumption of applications phases, but of course the finer the granularity the more difficult is to scale to a large number of nodes in a cluster (*e.g.*, instrumenting with HAEC the supercomputer *Sunway TaihuLight* - 2nd in Top500 of June 2018 and that includes around 41 thousand computing nodes [8] - it would require a data collection network bandwidth of ~82 GS/s, with obvious overheads on software and storage to handle it).

An intuitive solution is to bring some of the intelligence to the edge, and codesign the monitoring infrastructure to leverage data analysis between distributed monitoring agents and a centralized unit. This is represented in Figure 1 (right), which shows distributed smart monitoring agents that can carry out real-time analysis per node (*e.g.*, feature extraction, ML inference, etc.) and share information with the centralized monitoring at a much lower rate (*e.g.*, detection of an anomaly in a node, plus other measurements at a lower rate needed for cluster level analysis). In this codesign, each monitoring agent has the complete knowledge of the status of its node, while the centralized monitoring unit has the complete view of the cluster, thus can carry out analysis at higher level.

Current State-of-the-Art (SoA) monitoring solutions allow to collect measurements in-band and out-of-band by mean of built-in tools (*e.g.*, Amester [9] or RAPL [10], which expose hardware performance counters) or custom sensors (*e.g.*, HDEEM [6] or HAEC [5], which expose fine grain power measurements), where the benefit of the out-of-band solution is no overhead on the computing resources. However, there is not yet a monitoring infrastructure for data centers that provides a flexible way to analyze all possible features (*i.e.*, all hardware performance counters and the entire signal bandwidth of the node's power consumption). This work focuses on a novel scalable and high resolution monitoring infrastructure for data centers and HPC systems, which is completely out-of-band and provides a highly flexible environment to work both on the edge and at a cluster-level for data centers analytics, automation and control.

Contributions of the work:

1) design of an out-of-band monitoring infrastructure - we named it DiG (*i.e.*, Dwarf in a Giant) - that exploits edge monitoring agents and centralized cluster-level monitoring for data centers analytics, automation and control. The platform design provides a highly flexible environment to work on different challenges. We added a custom power sensor at the plug to monitor the power consumption at high resolution, covering the entire signal bandwidth (sampling up to $20\,\mu s$) with a measurements precision below $1\,\%$ (σ) (therefore also suitable for the most rigorous power measurement requirements to benchmark a computing system in Top500 [11]). The system allows to interface with existing out-of-band telemetry (*e.g.*, Amester [9], IPMI [12]), but also with in-band built-in tools if required (*e.g.*, RAPL [10]). All the measurements are synchronized at sub-microseconds precision to obtain a detailed picture over time of the nodes and cluster state. We adopted a scalable and lightweight interface to the centralized monitoring (*i.e.*, MQTT [13]) to support large-scale computing centers. The monitoring infrastructure is technology agnostic (*i.e.*, already tested in different architectures, such as Intel, ARM and IBM) and low cost (*i.e.*, the custom power sensor does not require any motherboard redesign).
2) we report the performance of the monitoring agents (*i.e.*, measurements granularity, precision, synchronization, software overhead and scalability), along with some preliminary benchmarks of ML inference in the dedicated embedded computers to roughly understand their real-time capabilities, and some tests based on frequency-domain analysis to show the capability of the high resolution monitoring in unveiling high-frequency components directly related to the computation activity.
3) we validated and calibrated the high-resolution power measurements against a reference meter, and integrated the monitoring infrastructure in a SoA computing cluster (*i.e.*, D.A.V.I.D.E. [14]) that is in production and available to the users community since November 2017.

Outline: Section II presents the monitoring architecture. Its performance are analyzed in Section III, together with several case studies of frequency-domain analysis on the high-resolution power measurements. We report related works in Section IV and conclude the paper in Section V.

II. Monitoring System Architecture

One of the main challenges we faced during the monitoring system design was to make it suitable for any hardware architecture and low cost. With this goal we targeted only what is currently missing [5]: a custom power sensor that allows high resolution monitoring. We placed it at the node power source in order to do not require any motherboard redesign. We then interfaced it with a dedicated low cost embedded computer (one per node) that is suitable for monitoring applications.

A second challenge was to make it highly flexible in terms of monitoring capabilities. With this goal we interfaced the embedded computer with built-in tools to get a per-component monitoring (*i.e.*, hardware performance counters) and have the complete knowledge of the status of the node. This information along with the high resolution power monitoring can reveal not only insights on application behaviours, but also patterns on performance / failures of components (*e.g.*, FANs, HDD).

Finally, we exploited a scalable and lightweight interface (*i.e.*, MQTT) to send information to a centralized monitoring unit and perform cluster-level analytics. Figure 2 shows the main components of the monitoring system that will be described in this section.

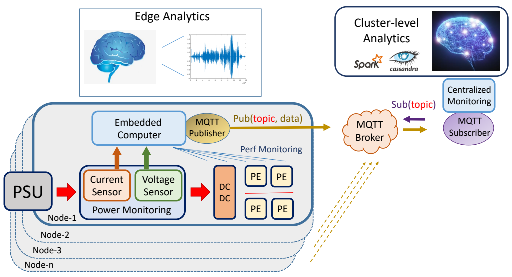

Fig. 2. Sketch of the monitoring system architecture.

A. *High-Resolution Power Sensing*

To provide high resolution measurements of the nodes power consumption we placed a power sensing module between the Power Supply Unit (PSU) and the DC-DC converters that provide power for all the processing elements (PE) / electrical components within the node. We use a voltage divider based on high precision resistors to measure the voltage and a current transducer to measure the current. Their outputs are then connected to an ADC integrated in the embedded monitoring board, via first-order low-pass filters needed to countering aliasing effects. We have chosen a voltage divider as it provides a simple but powerful solution to properly scale the voltage in input to the ADC without any additional hardware (*e.g.*, a power supply would be needed if using active components). For the current transducer we tested two configurations: one based on a Hall Effect (HE) sensor - *i.e.*, *Allegro MicroSystems ACS770* [15] - and one based on a current mirror and shunt resistor. Thanks to the accurate output linearity, both solutions report similar results. In particular, we tested the first configuration with Intel Xeon E5-2600 v3 (Haswell) and ARM Cavium TunderX architectures, while the second one on a cluster based on OpenPOWER IBM Power8.

B. Embedded Monitoring and Edge Analytics

We selected a Beaglebone Black (BBB) [16] as embedded computing board as it provides several interesting features off-the-shelf: (i) it includes a 12-bit 8-channels ADC needed for the power-sensing module, (ii) hardware support for PTP which allows sub-microsecond measurements synchronization [17], and (iii) an ARM Cortex-A8 processor with NEON technology, useful for DSP processing and edge ML inference (*e.g.*, by mean of Arm NN SDK [18], which enables efficient translation of existing neural network frameworks, such as TensorFlow, to Arm Cortex-A CPUs). Moreover, it includes two programmable real-time units (PRUs) useful for real-time acquisition and extra processing on-board.

It should be noted that standard computing servers already integrate an embedded system used for real-time monitoring and management, namely the Baseboard Management Controller (BMC). This is usually a closed platform with no access to the firmware, but thanks to the recent OpenBMC project [9] few vendors started to release it open-source. We decided to do not use the BMC for this purpose as (i) we needed to add extra-hardware to integrate an ADC and interface it with the custom power sensor (the BBB already includes an ADC), (ii) it does not provide hardware support for PTP, and (iii) it is based on an old ARM processor family (*i.e.*, ARM11 [19]) which is not a good choice for edge analytics. Moreover, (iv) it is a critical component to ensure safe working conditions of the nodes, thus it is more convenient to do not overwhelm the processing resources with our monitoring / edge-analytics software stack.

The power monitoring software exploits the ADC continuous sampling mode, therefore the two input channels (*i.e.*, current and voltage) are continually sampled, averaged in hardware and stored in a FIFO that is managed by a kernel driver. The power measurements are then exposed to the user-space monitoring daemon, which is in charge of converting data from integer to Watt and associate them with a timestamp. This daemon also collect node's performance measurements from hardware performance counters via built-in tools (*e.g.*, IPMI [12], Amester [9] and RAPL [10]). In this way we can perform edge analytics on a target use-case (*e.g.*, ML inference for anomalies detection) and send the results, together with the power and performance measurements at a lower rate, to the centralized monitoring unit for cluster-level analytics.

C. Centralized Monitoring Unit and Cluster-Level Analytics

We exploit a centralized monitoring, based on the open-source ExaMon [20], [21], to carry out cluster-level analytics with data coming from multiple nodes. To send data from the distributed monitoring agents to the centralized monitoring unit, we adopted MQTT [13], which is a robust, lightweight and scalable protocol, already used for large-scale systems both in industry and academia (*e.g.*, Amazon, Facebook, [13], [21]). Figure 2 outlines its publish / subscribe communication model, where the publishers (running in the embedded computers) send measurements to a broker, along with a topic that corresponds to the monitored metric (*e.g.*, power consumption). The broker resides in the centralized monitoring unit together with the subscriber. The latter is used to filter and collect the data that is interested on, and expose them to a Big Data engine, namely Apache Spark. The measurements are also stored in a scalable database - Apache Cassandra - enabling ML analytics both in streaming and batch mode.

III. Experimental Results

This section reports the performance of the monitoring agents and a set of benchmarks based on Fourier analysis, to show the capability of the high-resolution monitoring in revealing fine grain computation activity.

A. Monitoring Agents Performance

Performance Measurements: To provide a completely out-of-band monitoring, we integrated our infrastructure in a SoA OpenPOWER computing cluster, namely D.A.V.I.D.E. [14], that consists of 45 nodes (3 racks with 15 nodes each) based on IBM Power8. In this system we take advantage of its out-of-band telemetry and collect via Amester through the On Chip Controller (OCC) [9] 242 metrics per-component every 10 s (*e.g.*, performance of Core, Cache, FAN, etc.), and via IPMI 89 metrics per-component every 5 s. All these measurements are sent to the centralized monitoring for cluster-level analytics, but can also be exploited on-board for real-time edge analysis on a target use-case (*e.g.*, anomalies detection).

Power Measurements: To cover the entire signal bandwidth of the node power consumption at the plug (*i.e.*, tens of microseconds, observed with a professional oscilloscope - *Keysight DS0X3054T* - attached to the power sensor), we modified the ADC driver to reach a sampling rate of 800 kS/s per channel and set a hardware averaging every 16 samples: this is equivalent to 50 kS/s (*i.e.*, 20 µs) and allows to obtain, after calibration against a reference meter, a precision below 1 % (σ) of uncertainty (a.k.a. oversampling and averaging method [22]). It is noteworthy that this precision make our system suitable for the most rigorous requirement needed to benchmark a HPC system in Top500 [11]. Finally, we send the measurements to the centralized monitoring for cluster-level analytics at the rates of 1 s and 1 ms, while measurements at higher resolution can be analyzed directly on board.

Software overhead: The CPU usage of the monitoring software stack is below 46 %. In particular, the performance monitoring requires ~11 % and the power monitoring ~35 %. We have run also some preliminary benchmarks to evaluate the computing capabilities of the embedded platform: we can perform (i) real-time Fast Fourier Transform (FFT) of the high resolution measurements (*e.g.*, useful for feature extraction [23]), in a time window of ~40 ms with around 7 % of CPU usage (4096 B per FFT), and (ii) ML inference via Tensorflow, implementing Resnet [24] with 8 layers and channels $\{16, 16, 32, 64\}$, reaching a test accuracy of ~87 % on CIFAR-10 dataset [24] and respecting a real-time constraint of 30 ms per image (3072 B per image). Moreover, it must be noted that we did not use any optimization to run ML inference (*e.g.*, ARM NN SDK [18] or Tensorflow Light [25]), which means these preliminary results can be further improved.

Synchronization: To ensure accurate and precise timestamping of the measurements, we exploit PTP hardware obtaining sub-microsecond synchronization (*i.e.*, smaller than sampling period) across multiple nodes.

Scalability: To benefit from a scalable interface to the centralized monitoring unit, we exploit Mosquitto [21] which is a Linux implementation of MQTT that consists of a single thread process. Our tests show that Mosquitto broker (*i.e.*, bottleneck of the network) running in an Intel E5-2600 Haswell can handle up to 16 publishers that sends data every millisecond using just 30 % of a core, and of course it is possible to increase the number of brokers if needed. In our current configuration of the monitoring system integrated in the OpenPOWER computing cluster, we use one broker for all performance counters and three brokers (one per rack) for the power measurements, with no particular issue for the users / system admins of the data center since November 2017. Moreover, we tested MQTT with a similar configuration on all 516 computing nodes of GALILEO at CINECA, Italy (Intel Xeon E5-2630v3 processors), proving that this interface is suitable for large-scale systems.

B. Feature Extraction Benchmarking

This section wants to show the capability of the high resolution monitoring in unveiling high-frequency components directly related to the computation activity. We exploit Fourier analysis as an example of feature extraction technique for time series that is suitable for deep learning algorithms (*e.g.*, Deep Neural Networks - DNNs [23]). Future works can extend this real-time analysis targeting specific use-cases (*e.g.*, anomalies detection in workloads) and (i) exploit it, together with performance counters, as input data for DNN models running inference in the monitoring agents [23]; or (ii) just send it with lossy compression algorithms (data are sparse, as shown by the following tests) to the centralized monitoring for cluster-level analytics. We note that due to the limitations of SoA power monitoring support for computing nodes, up until now this kind of analysis could not be performed in production data centers.

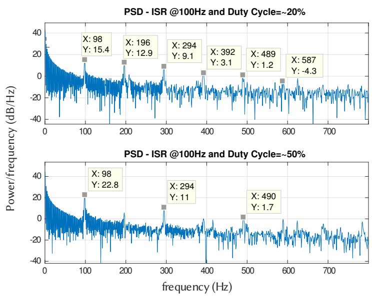

Fig. 3. PSD of an ISR at 100 Hz with different set of instructions.

We start the evaluation with a synthetic benchmark on the computing node, consisting of an Interrupt Service Routine (ISR) that we run every ~10 ms (~100 Hz) with different set of instructions. In particular, the first set of instructions corresponds to a duty cycle of 20 % in power consumption (*i.e.*, 2 ms of workload and 8 ms of sleep), while the second set to 50 %. Figure 3 shows the Power Spectral Density (PSD) for the two cases, computed in a time window of 40 ms. According to Fourier analysis, the set of instructions with duty cycle 20 % (top) shows the fundamental at ~100 Hz plus all its harmonics, while the one at 50 % (bottom) correctly reports only the fundamental and the odd harmonics (even harmonics are not completely null due to the not exact 50 % duty cycle). This example shows that our monitoring can really capture spectral properties of different workloads in execution.

The second set of benchmarks, reported in Figure 4, wants to show some frequency-domain patterns of real bottlenecks and scientific applications. Goal of this test is not to analyze in depth the reasons behind the peaks, but instead to show that different patterns emerge in the power spectrum with different workloads, which can be used as input features for DNN algorithms. In particular, comparing the first plot which portrays the PSD of the computing node in idle, with the second and third plots that depict respectively a memory bound application and a real scientific application (*i.e.*, Quantum Espresso - QE [26]), we can clearly see three different patterns (peaks highlighted with dark / light circles to indicate stronger / weaker magnitude). More in detail, (i) the system in idle reveals six main peaks (dark red circles) plus other weaker peaks (light red circles) spread in the entire bandwidth 0–12 kHz (richer activity in 0–4 kHz); these main peaks persist also in the other two benchmarks, but (ii) the tested memory bound application shows also three main peaks more (dark green circles) and activity only up to 6 kHz (almost flat for frequencies above), while (iii) QE reports four main peaks more with respect to idle (dark yellow circles) and rich activity in all the spectrum.

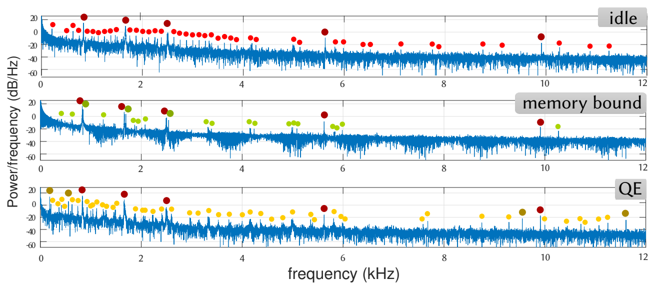

Fig. 4. Example of PSD patterns of real bottlenecks and applications.

IV. Related Work

Existing off-the-shelf methods to measure power and performance of computing nodes in data centers, rely on in-band or out-of-band telemetry depending on the technology vendors. In particular, an example of in-band solution is Intel RAPL [10], while examples of out-of-band solutions are IBM Amester [9] and the two standards IPMI [12] and Redfish [27] (*i.e.*, new protocol for managing data centers hardware, that fixes the security vulnerabilities of IPMI [9]). All these built-in tools allow a fine grain per-component monitoring (*i.e.*, based on hardware performance counters), but not high-resolution power monitoring (*i.e.*, covering the entire signal bandwidth).

Pushed by the growing interest on fine-grained power monitoring, industry and academia researchers are providing custom solutions for data centers. Examples are HDEEM [6], PowerInsight [7] and HAEC [5]. The first two systems provide power consumption measurements up to milliseconds, while the last one has a much more fine grain insight, with a sampling rate up to 500 kS/s. All these custom solutions focus on only monitoring the power consumption (*i.e.*, no performance knowledge) and send all the measurements to a centralized monitoring unit for analysis. Thanks to them, new opportunities for research on energy efficiency and other challenges are now possible, but going toward high resolution measurements this kind of monitoring design entails scalability issues (*e.g.*, as claimed in [5], HAEC is suitable for high resolution monitoring in just a node, but not in a cluster).

Comparison with SoA: In our system (*i.e.*, DiG) we combined all these features to enable research on several challenges for analytics, automation and control of data centers, with a highly-flexible monitoring platform: (i) we work completely out-of-band (*i.e.*, no impact / perturbation on the computing resources); (ii) we collect all performance counters and (iii) the full power bandwidth at the plug (*i.e.*, sampling at 50 kS/s) (iv) with high precision (*i.e.*, below 1 % - σ); (v) we provide highly synchronized measurements (*i.e.*, sub-microsecond) for a detailed correlation of the activities within the cluster; (vi) we leverage the monitoring between edge and a centralized unit, by exploiting dedicated embedded computers to collect measurements (they have complete knowledge of the status of their node) and have the possibility to carry out both edge and cluster-level analytics; (vii) the system is scalable (thanks to our flexible design, based on edge monitoring agents and a robust and scalable protocol - MQTT - to the centralized monitoring, where we analyze data at a lower rate), (viii) technology agnostic (*i.e.*, tested on ARM, Intel and IBM) and (ix) low cost (*i.e.*, no motherboard redesign required).

V. Conclusion and Future Works

This work reports on the design of a novel monitoring infrastructure - namely DiG - that enables real-time high-resolution profiling and analytics of data centers, for their automation and control. Main design choices include a complete out-of-band monitoring of power and performance, with a dedicated embedded computer per node to perform edge analysis, and a custom power sensor at the plug for high-resolution and high-precision measurements. We report (i) architecture design choices, (ii) monitoring platform performance and (iii) benchmarks based on Fourier analysis to show the high resolution monitoring insights.

In future works we will exploit the two PRUs for measurements acquisition and pre-processing, completely freeing the main processor from the monitoring software stack and use it only for edge analytics purposes. Moreover, we plan to use the system installed in the OpenPOWER computing cluster for research on ML techniques to address current data centers challenges (*e.g.*, detection of anomalies in workloads).

Acknowledgment

The authors would like to thank the Italian supercomputing center CINECA and Francesco Beneventi from University of Bologna for their support.

References

[1] Z. Liu *et al.*, "Toward a smart data transfer node," *Future Generation Computer Systems*, vol. 89, pp. 10 – 18, 2018.
[2] D. Duplyakin *et al.*, "Active learning in performance analysis," in *2016 IEEE International Conference on Cluster Computing (CLUSTER)*, Sept 2016, pp. 182–191.
[3] S. Ahmad *et al.*, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134 – 147, 2017.
[4] A. Tang *et al.*, "Unsupervised anomaly-based malware detection using hardware features," in *Research in Attacks, Intrusions and Defenses*, A. Stavrou, H. Bos, and G. Portokalidis, Eds. Cham: Springer International Publishing, 2014, pp. 109–129.
[5] T. Ilsche *et al.*, "Power measurement techniques for energy-efficient computing: reconciling scalability, resolution, and accuracy," *Computer Science - Research and Development*, Apr 2018.
[6] D. Hackenberg *et al.*, "Hdeem: High definition energy efficiency monitoring," in *Energy Efficient Supercomputing Workshop (E2SC), 2014*, Nov 2014, pp. 1–10.
[7] J. H. Laros *et al.*, "PowerInsight - A commodity power measurement capability," in *Green Computing Conference (IGCC), 2013 International*, June 2013, pp. 1–6.
[8] H. Fu *et al.*, "The sunway taihulight supercomputer: system and applications," *Science China Information Sciences*, vol. 59, no. 7, p. 072001, Jun 2016.
[9] T. Rosedahl *et al.*, "Power/Performance Controlling Techniques in OpenPOWER," in *High Performance Computing*, J. M. Kunkel, R. Yokota, M. Taufer, and J. Shalf, Eds. Cham: Springer International Publishing, 2017, pp. 275–289.
[10] K. N. Khan *et al.*, "Rapl in action: Experiences in using rapl for power measurements," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 3, no. 2, pp. 9:1–9:26, Mar. 2018.
[11] E. W. Group, *Energy efficient high performance computing power measurement methodology (v.2.0 RC 1.0)*, 2017, https://eehpcwg.llnl.gov/assets/sc17_bof_methodology_2_0rc1.pdf.
[12] Intel, Hewlett-Packard, NEC, Dell, and T. Rep., *IPMI Specification, V2.0, Rev. 1.1*, 2013.
[13] U. Hunkeler *et al.*, "Mqtt-s – a publish/subscribe protocol for wireless sensor networks," in *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, Jan 2008, pp. 791–798.
[14] W. A. Ahmad *et al.*, "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)*, May 2017, pp. 964–973.
[15] Allegro MicroSystems, *ACS770xCB Datasheet Rev. 4, 2015*.
[16] Texas Instruments, *BeagleBone Black System Reference Manual*, Rev. C.1, 2014.
[17] A. Libri *et al.*, "Evaluation of synchronization protocols for fine-grain hpc sensor data time-stamping and collection," in *2016 International Conference on High Performance Computing Simulation (HPCS)*, July 2016, pp. 818–825.
[18] *Arm NN SDK*. [Online]. Available: https://developer.arm.com/products/processors/machine-learning/arm-nn
[19] ASPEED, *AST2500 Advanced PCIe Graphics & Remote Management Processor*, AST2500 Datasheet.
[20] *Examon HPC Monitoring*. [Online]. Available: https://github.com/EEESlab/examon
[21] F. Beneventi *et al.*, "Continuous learning of hpc infrastructure models using big data analytics and in-memory processing tools," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 1038–1043.
[22] C. Villa-Angulo *et al.*, "Bit-resolution improvement of an optically sampled time-interleaved analog-to-digital converter based on data averaging," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 4, pp. 1099–1104, April 2012.
[23] S. Lin *et al.*, "Fft-based deep learning deployment in embedded systems," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 1045–1050.
[24] K. He *et al.*, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
[25] *TensorFlow Lite*. [Online]. Available: www.tensorflow.org/mobile/tflite/
[26] P. Giannozzi *et al.*, "Quantum espresso: a modular and open-source software project for quantum simulations of materials," *Journal of Physics: Condensed Matter*, vol. 21, no. 39, p. 395502, 2009.
[27] P. Kumari *et al.*, "Validation of redfish: The scalable platform management standard," in *Companion Proceedings of the10th International Conference on Utility and Cloud Computing*, ser. UCC '17 Companion. New York, NY, USA: ACM, 2017, pp. 113–117.