Dwarf in a Giant: Enabling Scalable, High-Resolution HPC Energy Monitoring for Real-Time Profiling and Analytics

Antonio Libri, Student Member, IEEE, Andrea Bartolini, Member, IEEE, and Luca Benini, Fellow, IEEE

Abstract—Energy efficiency, predictive maintenance and security are today key challenges in High Performance Computing (HPC). In order to be addressed, accurate monitoring of the power and performance, along with real-time analysis, are required. However, modern HPC systems still have limited power introspection capabilities, lacking fine-grain and accurate measurements, as well as dedicated systems for live edge analysis. With the goal of bridging this gap, we developed DiG (Dwarf in a Giant), an enabler framework for green computing, predictive maintenance and security of supercomputers. DiG provides high quality monitoring of power and energy consumption of HPC nodes. It is completely out-of-band and can be deployed in any hardware architecture/large-scale datacenter at a low cost. It supports (i) fine-grained power monitoring up to 20 μs (50x improvement in resolution than state-of-the-art - SoA); (ii) below 1 % (σ) of uncertainty on power measurements, which makes it suitable for the most rigorous requirements of HPC ranking lists (i.e. Top500); (iii) high-precision time-stamping (sub-microsecond), which is three order of magnitude better than SoA; (iv) real-time profiling, useful for debugging energy aware applications; (v) possibility for edge analytics machine learning algorithms, with no impact on the HPC computing resources. Our experimental results show it can capture key spectral features of real computing applications and network intrusion attacks, opening new opportunities for learning algorithms on power management, maintenance and security of supercomputers.

Index Terms—Green HPC, Energy Efficiency, Power Monitoring, Intrusion Detection System (IDS), Embedded Software, Fast Fourier Transform (FFT).

I. INTRODUCTION

Power and performance monitoring are essential for today and future large-scale High Performance Computing (HPC) systems [1]–[3]. While performance requirements are constantly growing, computing infrastructure scaling is facing technological walls moving towards exascale systems (i.e. power/thermal walls). Comparing the current worldwide most powerful supercomputer (Sunway TaihuLight) with the previous one (now second, Tianhe-2), we observe that the improvement in performance matches by a factor of three the one in energy efficiency1. This is a direct effect of hitting the power wall in supercomputing installations: we are clearly in an era of power limited HPC evolution, driven by new hardware technologies along with high quality monitoring solutions of energy and performance.

Work in [2] reports four criteria to assess the quality of a performance measurement infrastructure: (i) high accuracy and (ii) sampling rate, (iii) measurement of individual components, and (iv) scalability to large number of sampling points. Moreover, [2] highlights the challenge of energy accounting by defining a special aspect of accuracy, energy correctness: instantaneous power readings used to compute the energy have to be updated frequently enough to ensure precise calculation of total energy.

Despite HPC built-in sensors can provide per component monitoring with a good quality [3]–[5], they are still missing the power information of the overall nodes, which is needed to correctly account for the total energy (i.e. including FANs, HDDs, etc.). Moreover, it misses an infrastructure that can provide high resolution monitoring at a large scale in real-time [6], where by high resolution we mean the combination of high temporal resolution, precise synchronization, accuracy and precision of the measurements. This is important not only for energy efficiency, but also for other challenges when dealing with large scale systems: several works in literature proved that by means of a high quality monitoring it is possible to use signal processing techniques based on pattern recognition, to predict possible failures and avoid expensive equipment replacement [7]–[9]. When facing large-scale systems this becomes crucial to save cost and improve resources availability, and efficiency. Moreover, using high quality monitoring along with state-of-the-art (SoA) machine learning (ML) algorithms allows to characterize user applications (e.g. detect memory/CPU bound phases) for a more efficient power capping/management of the computing nodes [10], [11], or also improving the security of datacenters by detecting anomalies/malicious activities [12], [13], which is nowadays a hot topic.

There is a clear need to develop novel cost-effective solutions for node level energy monitoring, capable to raise the bar on resolution, while ensuring real-time access at a large scale, and the flexibility and robustness needed for production environments. With this aim we designed DiG (Dwarf in a Giant), an enabler framework for green computing, predictive

1According to the most recent Top500 list (Nov. 2017), Sunway TaihuLight reaches 93 PFlops (Floating point operations per second) with an energy efficiency of 6 GFlops/W, while Tianhe-2 33.8 PFlops with energy efficiency of 2 GFlops/W.
maintenance and security of HPC systems. DiG provides an (i) out-of-band (zero impact on the computing resources), (ii) scalable, and (iii) high resolution monitoring for (iv) real-time profiling and edge analytics of the computing nodes (e.g. predictive maintenance, failure/anomaly detection, etc.). Moreover, it is (v) highly flexibility (it can be easily interfaced with existing built-in sensors), and (vi) low cost (it can be installed to any hardware architecture/existing datacenter, with no need for motherboard redesign).

It is noteworthy that DiG can be used not only on HPC systems and datacenters, but also in any large scale system where high resolution, real-time access to the measurements, flexibility (to interface with existing monitoring infrastructures, e.g. Wireless Sensor Networks), and cost play an important role (e.g. industrial electrical systems or power grids [14], [15]). Moreover, it can also be used as simple, cost effective solution for EMC/EMI (Electromagnetic Compatibility/Electromagnetic Interference) pre-compliance measurements [16]–[18], which is important for vendors to save cost, and time for doing tests in industrial facilities.

Organization of the paper: Section II investigates previous work. In Section III we present DiG and its hardware/software architecture. The framework performance are analyzed in Section IV, together with several case studies that prove the monitoring insights gained with the framework. We conclude the paper in Section V.

II. PREVIOUS WORK

Table I gives a quantitative and qualitative (Pros/Cons: +/-) overview of existing HPC monitoring solutions. The common way to measure the power consumption in HPC installations is by means of built-in sensors, via the Intelligent Platform Management Interface (IPMI) that queries the board management controller (BMC) [6]. This mechanism is mostly designed for administration purposes, and thus it lacks in accuracy and provides only a coarse time granularity (order of seconds) and no timestamping of the measurements.

Another out-of-the-box monitoring method relies on tools provided by HPC vendors, such as Intel RAPL [4] or IBM Amester [19]. RAPL provides an in-band per-component power measurement (e.g. processor sockets, DRAM, etc.) with a time granularity up to 1 ms, and sub-microsecond synchronization enabled by the Precision Time Protocol (PTP) [21], available in most of the new HPC node installations. Even though no official accuracy is reported by Intel, work in [4] showed starting from the Intel Haswell architecture RAPL measurements are accurate. Power measurement of the whole blade server is missing. Amester allows a per-component measurement up to 1 ms [2], [19], but no official power measurement accuracy is provided by IBM.

The use of external calibrated power meters provide accurate but coarse grain power readings (seconds) [2]. Work in [6] showed that by measuring blade servers at the DC input of the main board it is possible to appreciate power consumption details in the order of 100 µs. However, their approach is costly and limited in scalability as their goal was not to propose a feasible monitoring solution for large-scale systems, but to demonstrate that application’s details in the order of hundred of microseconds are visible.

Pushed by the growing interest on fine-grained monitoring, Bull-HDEEM [2] currently provides a time resolution of 1 ms at the power plug and 10 ms on voltage-regulators (CPU, DDR), with and accuracy of 2% and 5%, respectively. Power values are synchronized at few milliseconds using Network Time Protocol (NTP) [2] and can be accessed by reading a file stored in the BMC (up to 8 h of data recording max) via IPMI. This makes feasible post analysis of long power measurement acquisitions, and also a run-time usage of those values, as instantaneous readings are possible only at 1 S/s (sample per second) (due to IPMI limitations). Moreover, the usage of BMC as embedded monitoring device does not allow the implementation of new algorithms for run-time analysis.

The interest of using open low cost embedded platforms for data acquisition and processing on large-scale systems is growing fast [22], [23]. Works in [20] (ArduPower) and [1] (PowerInsight) focused on HPC monitoring using Internet of Things (IoT) devices. ArduPower provides a 16-channels per-component monitoring with a time resolution of ~2 ms. To the best of our knowledge it requires for each node a dedicated server connected with a serial interface to readout the power measurements. This becomes unfeasible as the number of nodes to be monitored increases. No measurement precision is provided. PowerInsight provides a scalable power monitoring up to 1 ms, with an average accuracy on the current channel of 1.8 %. No accuracy of the power measurements is provided [6]. Measurements are stored in log files and used for post-processing (no real-time profiling can be performed).

In this paper, we focus on a high resolution (with regards to time granularity, synchronization and precision) power and performance monitoring system that is suitable for large scale deployment, and can deliver run-time access to the measurements for live analysis (both on a centralized computational unit and distributed on the edge). As outlined in the Introduction, we believe this can open new opportunities for energy efficiency, maintenance and security of supercomputers and datacenters. Moreover, it is suitable for EMC/EMI per-compliance measurements, useful for vendors to save cost and time for doing tests.

Design goals included keeping the monitoring framework out-of-band (zero impact on the computing elements), flexible (in order to be easily interfaced with built-in monitoring tools, e.g. IBM Amester, Intel RAPL) and low cost (the system can be installed in any hardware architecture and existing datacenter, without any redesign of computing nodes). The paper contributions are as follows:

1) we present the first out-of-band high resolution monitoring system for large scale HPC/datacenters that provides run-time access to the power measurements for both edge analysis and live/post-processing profiling via a centralized computational unit. It provides 20 µs of time resolution (50x higher than SoA [1], [2], [4], [19], [20]), sub-µs time synchronization (smaller than sampling period and crucial for correct data correlation between multiple nodes), and below 1 % (σ) of precision, which makes it suitable for the most rigorous requirement of

2 [4] compares RAPL power readings with a calibrated power meter (LMG450 with accuracy 0.07 % + 0.23 W) and the curve fits well. 3BMC is a critical and closed component for the node’s power management.
TABLE I
SUMMARY OF RELATED WORK

| In-band       | Time Resolution | Sync Protocol   | Precision | Real-Time Profiling | Edge Analytics Capability | Flexibility | Scalability | Cost       |
|---------------|-----------------|-----------------|-----------|---------------------|---------------------------|-------------|-------------|------------|
| RAPL (Intel)  | 1 ms            | NTP/PTP         | -         | +                   | +                         | +           | +           | +          |
| IPMI [2]      | 1 s             | -               | -         | +                   | +                         | +           | +           | +          |
| AC Power Meters [2] | 1 s         | NTP             | below 1 % | -                   | -                         | -           | -           | -          |
| Amster (IBM)  | 1 ms            | -               | -         | +                   | +                         | +           | +           | +          |
| Out-of-band   |                 |                 |           |                     |                           |             |             |            |
| HDEEM [2]     | 10 ms / 1 ms    | NTP             | 5 % / 2 % | -                   | +                         | +           | +           | +          |
| PowerInsight  | ~2 ms (16 ch)   | NTP             | -         | +                   | +                         | -           | -           | -          |
| PowerInsight  |                 | NTP/PTP         | avg 1.8 % (1) | -                   | -                         | +           | +           | +          |
| DIG (this work) | 20 μs          | NTP/PTP         | below 1 % | +                   | +                         | +           | +           | +          |

Top500 ranking list [24]. The system can be easily integrated with built-in monitoring tools for a per-component performance monitoring. Moreover, it is low cost. We tested it on Intel architecture (results showed in this paper) and IBM Power8 [3], without any redesign of the motherboards. We also provide calibration and validation of the measurement precision against a reference meter, showing that neither systematic errors and distortions are introduced.

2) run-time edge analysis of HPC nodes through FFT. To the best of our knowledge, this is the first system that provides this capability.

3) a run-time technique to cap via software measurement precision (trading off time granularity) when required.

4) an extensive test campaign to show the insights of this monitoring. Tests show it is possible to characterize applications (e.g. memory/CPU bound applications or catch software interrupt service routines) and network attacks.

III. DiG FRAMEWORK

One of the main challenges we faced during the design of the framework was to make it suitable for any existing hardware architecture, and thus really low cost. With this goal we targeted only what is really missing: the power consumption of the overall node. A high resolution monitoring of this information can reveal not only insights on application behaviours, but also patterns on performance/failures of components (e.g. FANs, HDD, etc.). A second challenge was to make it flexible, in order to use existing built-in tools for a per-component monitoring and correlate our data with other sensors. With this goal we needed a scalable and flexible interface. This section introduces the three main components involved in the DiG framework. As regards to Figure 1, we have:

A. a power sensing module, which contains the sensors for measuring current and voltage, and is placed between the Power Supply Unit (PSU) and the DC-DC converters (which provide the power for all the processing and electrical components within the node).

B. an embedded monitoring board, namely the Beaglebone Black (BBB) [25], which implements the framework back-end together with the power sensing module. It is used for acquiring the measured signals, carry out on-board processing, and send them to a central unit for aggregation and correlation with data collected from other nodes and sources. The central unit implements the framework front-end and resides in remote nodes (e.g. management nodes in the HPC infrastructure).

C. a scalable interface to the central unit, namely the MQTT protocol, which organizes the data exchange in a topic/subscriber communication (see Section III-C) and allows to interface power measurements with data coming from other built-in sensors.

A. Power sensing module

Figure 2 shows the schematic of the power sensing module. We use a current transducer to measure the current and a voltage divider to measure the voltage. For the current transducer we tested two configurations: one based on a Hall Effect (HE) sensor, which is the one presented in this paper, and one based on a current mirror and shunt resistor [3]. Thanks to the accurate output linearity, both solutions report similar results. Following the description of the first configuration.

The HE sensor is the Allegro MicroSystems ACS770 [26]. It can measure currents in the range 0–100 A with low-intrusiveness, good linearity and high precision sensitivity (40 mV/A). It has a typical bandwidth of 120 kHz and an internal conductor resistance of 100 μΩ, which translates into a negligible power loss. As reported in the schematic, the HE sensor output is scaled to the maximum voltage allowed by the ADC on the embedded monitoring board (1.8 V), via a voltage divider based on high precision resistors. We have chosen voltage dividers as they provide a simple but powerful solution to scale the input voltage, without any additional hardware.

4We used this second configuration on D.A.V.I.D.E., a 45-node cluster based on IBM Power8, which was ranked 18th in Green500, Nov. 2018.
Power Source
12V
GND
Computing Node
Current Transducer
BBB
R1 R2 C1
Vout R3
Voltage Divider
Vout
VIout
Config. 1
HE sensor
12V
GND
Voltage Divider
Config. 2
VIout
12V
GND
Current Mirror
Rshunt

Fig. 2. Power Sensing Module circuit schematic.

(e.g. power supply is needed when using active components). We use then a first-order low-pass filter to counter aliasing effects. Indeed, due to the high operating frequencies of HPC nodes, the power consumption is highly dynamic, and therefore an anti-aliasing filter is required. We use a similar design to measure the voltage, with a divider network based on high precision resistors and a first-order anti-aliasing filter.

To evaluate the performance of the proposed framework, we manufactured a Printed Circuit Board (PCB) prototype to be interposed between the PSU and the motherboard (Figure 1). The PCB integrates the HE sensor, the voltage dividers and the anti-aliasing filters, and is suitable to be integrated within standard HPC nodes. During the PCB design, we have taken the following precautions to avoid a possible measurement accuracy degradation due to heating effects on the resistors of the voltage dividers (we discuss measurement accuracy in Section IV-A). Namely, (i) we chose resistors with the same Temperature Coefficient of Resistance (TCR), and (ii) we placed them close to each other (same temperature on both) and (iii) far from external sources of heating (to avoid uneven heating effects). Indeed, as the voltage divider output depends only on the ratio of the two resistors, the impact of the resistance drifts (due to TCR) on the correctness of the output voltage, and thus on the measurement precision, is negligible. Furthermore, to face a measurement accuracy degradation over lifetime (several years), we have chosen high precision resistors with an endurance of 0.5 %. In the worst case scenario, this results in a drift of ~0.5 % of the maximum power consumption (1200 W), which we can calibrate in software as we know the expected maximum value.

B. Embedded power monitoring

We used a Beaglebone Black as embedded monitoring board. We selected it over other IoT platforms, as already proven to be suitable for monitoring applications [23], as it provides several interesting features out-of-the-box. It is based on the TI Sitara AM335x SoC, which includes an ARM®-Cortex-A8 processor, two programmable real-time units (PRU), useful for real-time on-board processing, and a 12-bit 8-channels ADC. Figure 3.1 shows an overview of the hardware and software components of the data-acquisition chain. The bottom layer represents the ADC hardware module. We set the continuous sampling mode, therefore the input channels are continually sampled and stored in a hardware fifo (HW-FIFO). Before being stored in the HW-FIFO, the samples can be averaged in hardware by a factor of 1 (no_avg), 2, 4, 8 and 16. We define oversampling frequency, $F_{os}$, the actual rate at which samples are acquired, and sampling frequency, $F_s$, the rate after hardware averaging. $F_s$ gives the actual time granularity at which samples enters the software layers. When monitoring two channels (current and voltage), the oversampling frequency can be set in the range 100–800 kHz by tuning the ADC clock between 3–24 MHz.

When the HW-FIFO reaches a pre-set watermark on the number of samples, an interrupt is raised and the HW-FIFO is flushed into the main memory by the CPU via the ADC driver. This is shown in the kernel-space layer. The ADC driver involves two IRQ handlers: the top half [27]. The top half is the routine that responds to the threshold interrupt of the HW-FIFO, while the bottom half flushes the samples into a software fifo at kernel space (K-FIFO). Finally, at user space level the monitoring daemon is in charge of (i) flushing the K-FIFO into the user-space memory via the IIO Subsystem API, (ii) generating a timestamp and (iii) pre-processing. The pre-processing consists of a step to convert data from integer to Ampere, Volt and Watt, and a step to average the values in software, if required (see Dynamic Software Average in Section IV-A3). The coefficients for these conversions (linear gain and offset) are obtained by a calibration phase. By means of a compile time flag, the monitoring daemon can also be set to carry out run-time FFT analysis on these data. As shown in [7], [10], [12], this is useful for a run-time detection of anomalies (e.g. failures or network attacks), as well as a characterization of user applications (aiming at improving the HPC energy efficiency).

The monitored data are then sent to the framework front-end via MQTT and RJ45 connector.

Thanks to the continuous sampling method, the hardware guarantees a negligible uncertainty on the acquisition time of consecutive samples, ensuring correctness of the energy computation within a certain time period [2]. Moreover, the monitoring daemon exploits the continuous sampling mode to generate timestamps at each K-FIFO flush only, and not for every sample5, making negligible both overhead and uncertainty introduced by the timestamp function call.

On top of the data-collection, we exploit PTP hardware to provide accurate and precise timestamping, with sub-microsecond synchronization across multiple nodes. This is smaller than sampling period (20 μs) and crucial to avoid introducing jitter that would create problems during measurement correlation with data coming from other nodes and sources [2], [21].

C. Scalable interface to the framework front-end

We selected the open protocol MQTT [28] as a scalable interface to the framework front-end. It was designed by IBM and Eurotech to minimize device resources and network bandwidth, and already several works proved to be suitable for large-scale systems implementation (e.g. [5], [28], Facebook Messenger, Amazon Web Service). Another feasible option would be Power API [29], which is a new interesting project.

5The timestamps for each sample are then derived from this information.
that aims at standardizing power monitoring measurements, and we plan to make DiG in compliance also with it in our future works.

Figure 1 outlines the MQTT publish/subscribe communication model. Three entities are involved: a publisher, a broker, and a subscriber. The publisher resides in the monitoring board (it is implemented as part of the monitoring daemon), while the broker and the subscriber reside in remote nodes (framework front-end). The main idea is that the publisher sends the monitored data to the broker tagging them with a topic (the monitored metric, e.g. power consumption of node \( x \)). The subscriber then can connect to the broker and easily filter the data it is interested on. Its main goals are to (i) collect the power and energy measurements, and (ii) correlate them with data coming from other nodes and sources (e.g. built-in sensors used by RAPL or Amester), to carry out ML analysis [5] (e.g. for a smart management of the resources, anomaly detection, etc.). Finally, it is noteworthy that increasing the number of brokers accordingly to the system scaling (each broker associated with a group of publishers), it is possible to keep negligible the load on the amount of data each broker has to handle [5].

IV. RESULTS

This section reports: (A.) the evaluation of the DiG’s performance (in terms of software overhead, measurements validation and precision); (B.) a campaign of tests to show what is possible to appreciate with this high resolution, together with (C.) a real intrusion use case where we carry out a port scanning network attack; (D.) a quantification of the absolute and relative costs increase, incurred when adding DiG in a HPC installation. The server node used for the tests is a double socket Intel Xeon E5-2630 v3 (8 cores per CPU) with 128 GB of DDR3 SDRAM.

A. DiG Performance Evaluation

1) Embedded monitoring performance: the first set of tests wants to identify the best configuration for the BBB internal parameters in terms of sampling rate, usage of the CPU resources and signal-to-noise ratio (SNR) of the ADC readings. This is described in Figure 3.2, where the \( y \)-axis reports the percentage of CPU load and the \( x \)-axis the sampling rate. Each bar shows the three main components involved in the monitoring software stack: monitoring daemon (yellow), top half (orange) and bottom half (blue). On top of each bar we report the label "not safe" to indicate that samples are lost due to the high CPU utilization (and subsequent delay of the bottom half process to flush the HW-FIFO into the main memory), and the label "safe" otherwise. At the bottom of the figure we report the hardware average and ADC clock pairs for each sampling frequency in the plot. The figure shows an almost linear trend with the best trade-off in performance (empirically obtained) at 50 kS/s, which corresponds to a CPU usage below 35% and no loss of samples. Keeping constant this operating point, we exploit the oversampling and averaging method [30], by increasing the oversampling frequency to 800 kS/s (ADC clock at 24 MHz) and setting a hardware average of 16. From the CPU’s point of view, this is equivalent to 50 kS/s, but we enhanced the signal-to-noise ratio (SNR) on the power measurements from 25.4 to 40.7 dB\(^6\). This corresponds to an increment of the ADC resolution from 12 to 16 bit, and a measurements precision \( \sigma \) from 8.3 to 1.73 W (0.96% of uncertainty) considering a baseline power consumption of 180 W (system in idle). An in-depth analysis on the DiG measurement precision is reported in Section IV-A3.

We want to highlight that this is a very significant result with respect to HPC ranking list requirements (i.e. Top500, Green500). Indeed, this precision is suitable for the most rigorous level of measurement required by the Energy Efficient HPC Working Group (EE HPC WG) to rank HPC systems in the Top500 (Level 3, precision equal or better than 1% [24]). As remarked in [31], it is not common for HPC monitoring systems to have this high level of quality and we reach it with a low cost open-hardware infrastructure. Moreover, considering that the power consumption of a core is roughly 4 W, this precision would be enough to catch variations in the utilization of a single core, as well as power management effects.

Finally, the monitoring daemon can be set to carry out FFT run-time analysis of the measurements. When this future is enabled we use roughly 6.5% more of CPU usage, for a total of 41.5% considering also the monitoring stack (when sampling at 50 kS/s). It is noteworthy that the greatest part of the CPU workload is used by the two IRQ handlers to serve the HW-FIFO. Even if this is not a problem for the selected configuration we plan in future to use the dedicated hardware components already present in the Sitara SoC (such as the Direct Memory Access - DMA - or one of the two PRUs), freeing resources for more intensive digital processing on board.

2) DiG validation under different server workloads: to validating DiG both in the time and frequency domain, and under different working conditions of the HPC node, we first calibrated with a reference meter the conversion factor for the voltage and current. Then we carried out the following tests: (i) server off, (ii) server in idle and (iii) pulse train of instructions with fundamental frequency at 100 Hz (we use a custom synthetic benchmark to alternate high load computation phases with idle phases). In particular, the former allows to understand if the HE sensor introduce any distortion, as we are measuring a stable signal at 0 A. The second allows

\[ \text{The SNR is computed as } 10 \log(\mu^2/\sigma^2). \]
TABLE II
DiG VALIDATION RESULTS

| Average [mV] | σ [mV] | FFT main peaks [kHz] |
|--------------|--------|----------------------|
| Node-off     | Scope  | 206.17               | 2.29                  | 5.6                  |
|              | DiG    | 206.16               | 1.9                   | 5.6                  |
| Idle         | Scope  | 446.56               | 26.85                 | 0.83, 1.66, 2.49     |
|              | DiG    | 446.60               | 28.18                 | 0.83, 1.66, 2.49     |
| 100 Hz       | Scope  | 474.85               | 36.97                 | 0.1, 0.2, 0.3        |
|              | DiG    | 475.95               | 40.6                  | 0.1, 0.2, 0.3        |

Fig. 4. Pulse train @100 Hz: Comparison between oscilloscope (top) and DiG (bottom) monitoring on the current sensor output.

$$\sigma_P \approx \sqrt{I^2 \sigma_I^2 + V^2 \sigma_V^2}$$

Table III reports the resulting precision at 50 kS/s for three different server operating conditions: idle (~180 W), medium load (~600 W) and maximum load (~1200 W). The precision for ~68.3% of the power measurements (σ) is bounded between 1.73–3.96 W, for minimum (idle) and maximum load, respectively, and increases to 5.2–11.88 W when considering 99.7% of the samples (3σ). Of course the CV follows the opposite trend: it decreases when the power consumption increases (from 0.96%, in idle, to 0.33% for maximum workload, considering σ). Moreover, the table reports the results of precision when sampling at lower rates (by applying a software average), namely 25 kS/s, 1 kS/s and 1 S/s. As can be seen, the values of both σ and 3σ drastically improve to few watt precision already at 25 kS/s (even at the maximum load).

With the goal to dynamically increasing the DiG precision on the power measurements, we can set the monitoring daemon to use the Dynamic Software Average to dynamically switch to a lower sampling rate (e.g. to 25 kS/s by averaging in software the power samples), if it is monitoring low currents for a certain time period. Thanks to this trade-off we can always keep the monitoring precision below a pre-set threshold (up to sub-watt precision), which makes DiG suitable to be used in production environments as a high precision HPC energy monitoring solution.

B. DiG benchmarking

This section evaluates the capability of DiG in unveiling high-frequency power components directly related to the computation activity. We must stress the fact that due to

3The average value of ~206 mV in the first test corresponds to the HE sensor offset when is measuring 0 A, while the increase to ~446 mV correspond to the proportional consumption of the system in idle.
TABLE III

PRECISION BASED ON DYNAMIC SOFTWARE AVERAGE

|                | Idle -180 W [W] | Medium Load -600 W [W] | Max Load -1200 W [W] |
|----------------|----------------|------------------------|----------------------|
| 50 kS/s        | σ (CV) 1.73 (0.96 %) | 2.58 (0.43 %)           | 3.96 (0.33 %)        |
|                | 3σ (CV) 5.2 (2.89 %)         | 7.74 (1.29 %)           | 11.88 (0.99 %)       |
| 25 kS/s        | σ (CV) 0.5 (0.28 %)           | 1.26 (0.21 %)           | 2.28 (0.19 %)        |
|                | 3σ (CV) 1.5 (0.83 %)           | 3.72 (0.62 %)           | 6.84 (0.57 %)        |
| 1 kS/s         | σ (CV) 0.47 (0.26 %)           | 1.14 (0.19 %)           | 2.16 (0.18 %)        |
|                | 3σ (CV) 1.37 (0.76 %)           | 3.54 (0.59 %)           | 6.6 (0.55 %)         |
| 1 S/s          | σ (CV) 0.32 (0.18 %)           | 1.02 (0.17 %)           | 2.04 (0.17 %)        |
|                | 3σ (CV) 0.97 (0.54 %)           | 3.06 (0.51 %)           | 6.12 (0.51 %)        |

the limitations of the power monitoring support for standard computing nodes, up until now this kind of analysis can not be performed.

1) Pulse train of instructions: we start our analysis by applying a synthetic benchmark on the compute node to generate pulse trains of instructions with a controlled frequency and duty cycle. We report five configurations: a pulse train with fundamental at 100 Hz, where we compare two different duty cycles at 20 % and 50 %, and three pulse trains at -6.5 kHz, ~9 kHz and ~11 kHz, with 50 % duty cycle. Figure 5 reports the Power Spectral Density (PSD) for the first case, where we want to show that DiG can capture both fundamental and the different harmonics composition obtained by varying the duty cycle (100 Hz with 20 % duty cycle, top figure, and 100 Hz with 50 % duty cycle, bottom figure). The PSD is computed in a time window of 0.5 s, and the x and y axes report frequency (Hz) and power over frequency (dB/Hz), respectively. From the Fourier series theory [33], any pulse train \( x(t) \), with fundamental frequency \( f \), amplitude \( A \) and duty cycle \( d = k/T \) (where \( T \) is the period and \( k \) is the fraction of time that the pulse is high), can be reconstructed by the following synthesis equation:

\[
x(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(2\pi ft n)
\]

where

\[
a_0 = Ad \quad a_n = \frac{2A}{n\pi} \sin(n\pi d)
\]

Figure 6.6 and 6.7 highlight that DiG can capture the activity of set of instructions up to 45 µs. In particular, the peaks around 11 kHz are less pronounced due to the cut-off frequency of the low-pass filter.

2) Real workloads: this set of tests wants to prove that DiG can identify real system conditions exploiting a high frequency power sampling. Figure 6.1 represents the system in idle, where the dynamic tick is enabled by default. This means the Linux kernel runs without regular timer interrupts used to awake the scheduler process. Figure 6.2 shows instead the system in idle where we disabled the dynamic tick: its well visible the fundamental at 1 kHz and all its harmonics (at multiples of the fundamental) till 11 kHz. This prove that DiG can catch Operating System (OS) kernel activities such as timer interrupts and system ticks.

Figure 6.6 and 6.7 highlight that DiG can capture the bottlenecks of various components within the HPC node: they report the difference between a CPU bound and a memory bound synthetic benchmark, respectively. More in depth, the former is bound in the CPU ‘front-end’ process, which is the phase where instructions are fetched and decoded into operations that constitute them (it differs from the CPU ‘back-end’ process, where the required computation is performed), and the latter is stuck in the SDRAM. While measuring only the average value of the power consumption would not be enough to understand the difference between the two bottlenecks (~340 W for the memory bound and ~300 W for the CPU front-end bound), DiG can detect different spectral components in the two benchmarks, due to the different usage of architectural resources. Indeed, the CPU front-end bound benchmark shows different peaks than the memory bound in the range 0–6 kHz, activity that is clearly different also from the idle one.

Finally, we report in Figure 6.8 the PSD of a real application benchmark, Quantum Espresso (QE) [34], which is an open-source integrated suite for material modeling at the nanoscale and electronic structures calculation. Comparing the PSD of this benchmark with the one of the system in idle, different peaks that define the spectral footprint of QE rise up in the entire bandwidth between 0–12 kHz. These final tests prove that exploiting the fine-grain measurement support enabled by...
DiG, it possible to unveil application and operating system activities which are not visible otherwise.

3) Time-frequency analysis: real benchmarks activities are characterized by a varying workload. In order to capture it we can exploit the Continuous Wavelet Transform (CWT). Unlike the PSD, the CWT allows to construct a frequency representation of the signal over the time, which gives the possibilities to catch localized spectral components that would not be possible to appreciate otherwise. This is visible in Figure 7, where we run the QE benchmark and show a zoom of 2 s of both the time-domain acquisition of the related power consumption (top figure) and its CWT (bottom figure). In particular, the CWT reports time (seconds) and frequency (kHz) in the x and y axes, respectively, while the color of the pixels represents the magnitude. During the alternation between high and low load phases various frequencies rise up with a different intensity. For instance, in this case lower frequencies (around few Hz) seem to be more excited during the high computation load, while higher frequencies (between 1–2 kHz) seems to be unrelated by the degree of computation as they rise up with various intensities during the all acquisition’s window. DiG can probe all these activities, creating now novel opportunities and challenges for the use of signal processing algorithms together with the power and activity monitoring of computing nodes.

C. DiG as Intrusion Detection System (IDS)

Network security is a crucial challenge in HPC infrastructures and datacenters, to prevent attackers getting access into the system and steal sensible data or illegally use computing resources. Before intruding into the system attackers need to gather information about the target machine and its running services, and thus about vulnerabilities that can be exploited. This is called scanning phase, and one of the most popular tools for port scanning is NMAP [35]. In this section we show how DiG can detect the scanning phase of network attacks, capability that can be used in future to improve the quality of existing SoA intrusion detection systems (e.g. SNORT [36]).

The use case scenario is an attacker that try to collect information about the HPC/datacenter front-end node, to get access into the local network. Thus we run NMAP from a remote computer (outside the local network) with the OS detection mode enabled to understand open ports, running services and OS of the front-end target machine. The scanning attack requires around 10 seconds and Figure 8 shows the PSD of the DiG measurements when monitoring the front-end node. PSD are computed in a time window of 0.5 s and the top plot refers to the front-end in idle, while the second and third plot to the front-end under attack, at the second 2 and 8, respectively. Results show that the first phase of the NMAP scanning (e.g. plot at second 2) excited more low frequencies, such as the peak at 1 kHz, while the last phase excited also higher frequencies, like peaks at 1 kHz, 4 kHz, 8 kHz and 9 kHz.

Thanks to the DiG framework, this pattern recognition analysis can be used in future works to exploit ML classifiers which run on the edge (directly on the embedded computer) and correlate this information with SoA signature-based IDS (e.g. SNORT).

D. DiG cost analysis

To evaluate the absolute cost increase of adding DiG in a HPC system, we consider the most expensive case of using the PCB solution that we manufactured for prototyping purposes (of course an integration of DiG within the compute node, with a dedicated form factor design, would drastically decrease the deployment cost). Considering an average cost of ~45 $ per PCB (cost for manufacturing a small number of boards), and the cost of ~45 $ per BBB, we can realize a DiG monitoring point for each HPC node with just 90 $. Considering a price of 20 k$/ per HPC server (this is a conservative estimation with respect to SoA HPC nodes), this means an increase of only 0.45 % on the total cost.

V. Conclusion

This work proposes the use of a novel enabler framework for future green computing, predictive maintenance and security of HPC systems. The framework, named DiG, is low cost and provides high resolution monitoring (best in class in terms of time resolution, synchronization and precision), along with edge analysis, of HPC nodes power and energy consumption. DiG can be easily interfaced with built-in monitoring tools to correlate data coming from other sources [5], and allows scalability to a large number of sampling points (crucial for future exascale HPC installations). It provides precision below 1 % (σ), 20 µs time resolution (50x improvement with respect to SoA [1], [2], [4], [19], [20]), and sub-microsecond measurements synchronization (below the sampling period). We perform run-time FFT analysis of the measurements and show the insights of the monitoring through an extensive test campaign. DiG can capture useful characteristics on system conditions, computing applications (e.g. it can discriminate between different workloads) and network attacks, opening new opportunities for predictive maintenance, power management and security of supercomputers. Finally, it can also help vendors for EMI/EMC per-compliance measurements, saving cost and time for doing tests in industrial facilities.

Acknowledgment

The authors would like to acknowledge funding from the EU H2020 FET-HPC project ANTAREX (g.a. 671623), and from the ERC MultiTherman project (ERC-AdG-291125).

References

[1] J. H. Laros et al., “Powerinsight - a commodity power measurement capability,” in Green Computing Conference (IGCC), 2013 International, June 2013, pp. 1–6.

[2] D. Hackenberg et al., “Hdeem: High definition energy efficiency monitoring,” in Energy Efficient Supercomputing Workshop (E2SC), 2014, Nov 2014, pp. 1–10.

[3] 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.

[4] D. Hackenberg et al., “An energy efficiency feature survey of the intel haswell processor,” in 2015 IEEE International Parallel and Distributed Processing Symposium Workshop, May 2015, pp. 896–904.

[5] 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.
Fig. 6. Power Spectral Density of various synthetic benchmarks with the goal to show what is possible to appreciate at this fine time granularity.

[6] T. Ilsche et al., “Power measurements for compute nodes: Improving sampling rates, granularity and accuracy,” in Green Computing Conference and Sustainable Computing Conference (IGSC), 2015 Sixth International, Dec 2015, pp. 1–8.

[7] H. M. Hashemian and W. C. Bean, “State-of-the-art predictive maintenance techniques,” IEEE Transactions on Instrumentation and Measurement, vol. 60, no. 10, pp. 3480–3492, Oct 2011.

[8] S. Saponara et al., “Predictive diagnosis of high-power transformer faults by networking vibration measuring nodes with integrated signal processing,” IEEE Transactions on Instrumentation and Measurement, vol. 65, no. 8, pp. 297–306, Feb 2012.

[9] C. Chen et al., “Prediction of machine health condition using neuro-fuzzy and bayesian algorithms,” IEEE Transactions on Instrumentation and Measurement, vol. 57, no. 3, pp. 490–497, March 2008.

[10] G. A. Jacoby et al., “Detecting software attacks by monitoring electric power consumption patterns,” Jan. 25 2011, uS Patent 7,877,621.

[11] D. He and H. Leung, “Network intrusion detection using cfar abrupt-change detectors,” IEEE Transactions on Instrumentation and Measurement, vol. 57, no. 3, pp. 566–582, Sept 2009.

[12] F. Salvadori et al., “Monitoring in industrial systems using wireless sensor network with dynamic power management,” IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 9, pp. 3104–3111, March 2008.

[13] J. Pontt et al., “Developing a simple, modern and cost effective system for emc pre-compliance measurements of conducted emissions,” in 2007 European Conference on Power Electronics and Applications, Sept 2007, pp. 1–7.

[14] J. Pontt et al., “Multipurpose platform for power system monitoring and analysis with sample grid applications,” IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 3, pp. 566–582, March 2014.

[15] G. Da Costa and J. M. Pierson, “Characterizing applications from power consumption: A case study for hpc benchmarks,” in Information and Communication on Technology for the Fight against Global Warming, D. Kranzlmüller and A. M. Toja, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 10–17.

[16] J. L. Berral et al., “Towards energy-aware scheduling in data centers using machine learning,” in Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, ser. e-Energy ’10. New York, NY, USA: ACM, 2010, pp. 215–224.
Fig. 7. Time-domain acquisition (top) and CWT (bottom) when monitoring the activity of Quantum Espresso with DiG.

Fig. 8. DiG detecting the port scanning phase during a network attack.

Antonio Libri received the M.Sc. degree in electrical engineering and information technology from University of Genova, Italy, in 2013. After two years working as embedded software engineer in Socowave Ltd, Cork, Ireland, he joined in 2015 the Integrated Systems Laboratory, ETH Zurich, Switzerland, pursuing a Ph.D. degree. His current research interests include energy efficiency of HPC systems, with special emphasis on data monitoring and synchronization.

Andrea Bartolini (M’13) received the Ph.D. degree in electrical engineering from the University of Bologna, Bologna, Italy, in 2011. He is currently a Post-Doctoral Researcher in the Integrated Systems Laboratory, ETH Zurich, Zurich, Switzerland. He also holds a Post-Doctorate Position in the Department of Electrical, Electronic and Information Engineering Guglielmo Marconi, University of Bologna. His current research interests include green computing and dynamic resource management ranging from embedded to large-scale high performance computing systems with special emphasis on thermal and power-aware HW/SW codesign techniques.

Luca Benini is professor of Digital Circuits and Systems at ETH Zurich, Switzerland, and is also professor at University of Bologna, Italy. His research interests are in energy-efficient multicore SoC and system design, smart sensors and sensor networks. He has published more than 800 papers in peer-reviewed international journals and conferences, four books and several book chapters. He is a fellow of the ACM and Member of the Academia Europea, and is the recipient of the IEEE CAS Mac Van Valkenburg Award 2016.