This is a post-peer-review, pre-copyedit version of an article published in Computer Science - Research and Development. The final authenticated version is available online at: https://doi.org/10.1007/s00450-018-0392-9

Power Measurement Techniques for Energy-Efficient Computing: Reconciling Scalability, Resolution, and Accuracy

Thomas Ilsche(✉)[1] · Robert Schöne[1] · Joseph Schuchart[2] · Daniel Hackenberg[1] · Marc Simon[3] · Yiannis Georgiou[3] · Wolfgang E. Nagel[1]

2018-04-23

Abstract The rising concern for power consumption of large-scale computer systems puts a research focus on the respective measurement methods. Varying workload patterns and energy efficiency optimizations cause highly dynamic power consumption on today's compute nodes – a challenge for every measurement infrastructure. We identify five partly contradictory requirements that characterize such infrastructures: temporal granularity, spatial granularity, well-defined accuracy, scalability, and cost. In two projects we push the boundaries for these criteria: a scalable measurement solution for hundreds of nodes at millisecond granularity that is tightly integrated into the HPC system, and a sophisticated single-node instrumentation to measure the power consumption of application events in the microsecond range. Both measurement solutions are calibrated and their accuracy is carefully studied. We discuss scalable processing of the measurements for global monitoring in large-scale systems and use this data for energy efficiency analyses in combination with contextual information such as application performance trace data.

Keywords HAEC · HDEEM · power measurement · verification · energy efficiency

[1]Technische Universität Dresden, Center for Information Services and High Performance Computing (ZIH), 01062 Dresden, Germany
E-mail: {thomas.ilsche, robert.schoene, daniel.hackenberg, wolfgang.nagel}@tu-dresden.de
[2] High Performance Computing Center Stuttgart (HLRS), University of Stuttgart, 70569 Stuttgart, Germany
E-mail: schuchart@hlrs.de
[3] BULL SAS, Rue Jean Jaurès, 78340 Les Clayes, France
E-mail: {marc.simon, yiannis.georgiou}@atos.net

1 Introduction

Measuring the power consumption of components in a high performance computing (HPC) system poses unique challenges due to the dynamic power consumption and scale of those systems. For discussing measurement approaches, we use five key criteria:

(i) A high *temporal granularity* (sampling rate) is needed to understand the power draw of short phases in an application, e.g., a synchronization that only takes a fraction of a millisecond.
(ii) A good *spatial granularity* means that individual components can be measured separately, e.g., to distinguish between CPU and memory power consumption.
(iii) Without a well-defined *accuracy*, it can be impossible to interpret the measurement data. Is this 5 % reduction in energy consumption really an improvement or just an effect of the measurement with 10 % or even unknown uncertainty?
(iv) HPC requires measurement approaches that are *scalable* to thousands of nodes, especially to understand effects that only occur at scale.
(v) *Cost*, of course, is also an important factor.

In this paper, we describe the results of our efforts to reconcile these sometimes contradictory requirements. One of our proposed solutions pushes the limits regarding temporal and spatial granularity while retaining a good accuracy at the cost of scalability. The second approach represents a highly scalable solution that still achieves good temporal and spatial granularity and a well-understood accuracy.

2 Power measurement in High Performance Computing

For a long time, power measurement of compute components was focused on mobile devices, for which battery life has always been crucial. Since power limitation and energy efficiency are now also a hot topic for HPC and data centers, a number of measurement tools and approaches are available for this context today. System vendors provide scalable solutions with a fine spatial granularity. The accuracy of these solutions is driven by the focus on energy efficiency and the need for precise power capping. The tight integration with the system architecture also improves convenience and cost for customers. The solutions differ in their purpose and sophistication.

2.1 Integrated system solutions

Vendors have started to provide integrated solutions designed for easy use and low cost. Notable examples are IBM with the Amester [1] and Cray, who offer a Power Management Database (PMDB) [2]. On the chip level, both Intel and AMD provide measurements of consumed energy through RAPL and APM, respectively. However, all of these approaches exhibit characteristics that limit their use, especially with regards to temporal and spatial resolution, as has been described in our previous work [3, 4].

2.2 Add-on solutions

Even though some vendors support power measurement interfaces, there are still reasons to enhance the power measurement capabilities, e.g., to achieve higher temporal or spatial resolution or to validate the existing measurement framework.

Power analyzers can often easily be connected to the AC input of a system or a group of systems. In this paper, we use a ZES ZIMMER LMG450 power meter, which provides accurate power measurements at 20 Sa/s.

A more detailed measurement can be conducted with fine grained instrumentation frameworks, which instrument individual voltage lanes and devices and provide a sampling rate of 1 kSa/s. PowerPack [5] is a hardware and software framework to access various types of power sensors. In a typical implementation, it uses resistors added to several DC pins and a National Instruments input module. PowerInsight [6] is a solution from Penguin Computing and Sandia, which uses sensor modules as Molex adapters and riser cards that are equipped with small Hall effect sensors. PowerMon2 [7] is a low-cost power monitoring device for commodity computer systems with a measurement rate of up to 1024 Sa/s. Another low-cost solution is ARDUpower [8] which uses an Arduino board and an Allegro ACS713 Hall-effect sensor circuit. Depending on the number of used channels, the sampling rate can reach up to 5,880 Sa/s.

2.3 Energy and power measurement APIs

The programming interfaces to access the measurement infrastructures described above are fragmented and in most cases proprietary. The PowerAPI [9] tries to establish a standardized interface that cuts through all system layers to provide measurement and control of energy and power consumption. A survey on existing power and energy measurement APIs is provided in [10].

3 Instrumentation

This section describes how systems can be instrumented, discussing the choices of measurement domains, instrumentation points and measurement sensors as well as the required analog processing. Alongside that discussion, we follow two distinct measurement approaches:

- The **HAEC** measurement infrastructure strives to push the limits regarding temporal resolution with a good spatial resolution and to maintain a well understood accuracy. The approach has been initially described in [11].
- In the **HDEEM** project, a vendor collaboration, we have built a highly scalable, integrated HPC measurement solution that is also verified and provides good temporal and spatial resolution. The approach and a prototype implementation are described in [12]. It is now deployed in a production HPC system with 1456 nodes.

3.1 Measurement domains

The first step in measuring power is the selection of the set of components under investigation. This is limited by the utilized instrumentation point, i.e., the point where the sensor is added. The closer an instrumentation point is to the individual component, the better the spatial and temporal resolution can be.

Power distribution can be considered as a tree with the actual components as leaves. Power conversion usually happens at intermediate edges and each power conversion introduces some low-pass effect. Simply speaking, energy for executing a single instruction is not converted on-demand, but stored within intermediate capacitances. Therefore, it is impossible to measure this energy in isolation. Similarly, the power consumption of a single core cannot be measured individually because the closest accessible instrumentation point covers multiple cores.

For the HAEC system, we selected the individual direct current (DC) power connectors as instrumentation points plus an additional PCIe riser card to measure power from the mainboard to a PCIe card. Furthermore, the entire AC power input is measured using a reference power meter. This results in the following measurement domains:

- 2 × Sockets, each covering the sum of CPU and Memory
- Mainboard via separate 12 V, 5 V, 3.3 V ATX connectors.
- GPU via PCIe riser and 8+6 pin power. The power supplied via PCIe is part of the mainboard ATX power.
- SSD separate for 12 V and 5 V.
- Total system fans consumption.

The HDEEM measurement covers:

- Total node power
- 2 × CPUs
- 4 × DRAM groups

3.2 Measurement sensors

The next step for instrumentation is the selection of a suitable measurement sensor. One common option are measurement shunts: well-defined resistors that cause a voltage drop that is proportional to the current going into a component. Since the voltage drop is very small, this signal needs to be amplified as described later.

Hall effect sensors represent an alternative that measure the current indirectly through the magnetic field. This is less intrusive and can even be applied in the form of a current clamp. In contrast to shunts, they are active sensors, requiring a dedicated power supply. Thus, they provide a higher voltage output that may not require amplification.

Both of these current sensors provide an output voltage that needs to be measured alongside with the voltage supplied to the component that is measured. For HAEC, we use precise measurement shunts that are plugged in-between the Molex DC power connectors.

3.3 Analog signal processing

The measurement sensors themselves produce an analog voltage signal that represents the current consumption of the component. In many cases, this signal needs to be amplified into a common, easily measurable range and a low-pass filter should be applied to remove high frequencies that cannot be sampled correctly by the data acquisition. If the signal changes faster than the digital sampling is performed, aliasing effects can introduce significant errors. Even though the instantaneous power values could technically be correct, integrating power over time might yield inaccurate energy values. In HAEC, we use custom amplifier boards that can be calibrated and contain configurable low-pass filters. The HDEEM node measurement signal is filtered by a second order low-pass filter at 600 Hz before the amplification and digitalization. Moreover, integrated circuits for energy metering exist that combine different functions, e.g., amplification, filtering, A/D conversion, and energy summation.

4 Measurement data processing

4.1 Fine granular measurement processing

For the HAEC sensors, we aim for very high sampling rates. The data acquisition is performed using two National Instruments DAQ cards: one PCI-6255 that samples all sensors at ≈7 kSa/s and one PCI-6123 that samples four selected sensors at 500 kSa/s. The data acquisition is performed by a separate system to avoid any perturbation of the system under test. The monitoring system runs a daemon that stores measurement data in memory and makes it available for post-mortem application analysis and live monitoring.

4.2 Scalable integrated measurement processing

The HDEEM infrastructure was designed with scalability, low-cost, and low-overhead in mind. The analog node measurement signal is sampled with 8 kSa/s, while the VRs directly provide digital values at 1 kSa/s. The raw digital signal is continuously read by a field-programmable gate array (FPGA) on the system board, which applies a digital low-pass filter to prevent aliasing artefacts in the digital signal. The result is a data stream with 1 kSa/s for the node measurements and 100 Sa/s for each of the VR measurements.

The data from the FPGA is continuously read by the Baseboard Management Controller (BMC), from where it can be accessed by applications. The BMC's memory is capable of storing up to eight hours of measurement data of all sensors.

Applications start and stop the recording of the high-resolution power measurement data through a low-latency GPIO signal, which allows for easy clock synchronization between the BMC and the host. Measurement data can be received by the application at any time after the start signal. The energy counters can be read by the application at any time. An API has been designed for access to power and energy measurements that supports both in-band access through GPIO and PCIe as well as out-of-band access through IPMI [13].

4.3 Holistic measurement data collection and storage

The Dataheap infrastructure was designed as a scalable infrastructure for storing and processing continuous measurement data [14]. It is therefore suitable as a storage infrastructure for the power measurements that are online regardless of running jobs. This allows users and administrators to access the data for online and post-mortem analyses without manual intervention to enable the recording.

The HDEEM power measurement data is collected from the BMCs and pushed to the Dataheap by a management node. For both HDEEM and HAEC measurements, the data is sampled down to 1 Sa/s to ensure a unified sampling rate and to reduce the storage requirements for long-term archival.

Dataheap is accessible through a number of interfaces. Users can freely browse through the history of measurements using a web interface and combine the data from multiple measurement sources in one display. This way, the system behavior can be analyzed and compared over a long time period and the effects of changes to the system can be tracked based on historical data.

4.4 Integration with performance analysis tools

Going beyond system monitoring, application energy efficiency analysis requires an integration of the energy measurements with performance analysis tools, such as Score-P [15]. The main focus of these tools is to capture the behavior of parallel applications in order to determine performance bottlenecks and to help the user understand the source of performance anomalies. Moreover, with the help of metric plugins, the measurements can be arbitrarily extended by application external and internal metrics to support the analyst. Plugins for HDEEM and HAEC can integrate the respective power measurement data into application performance traces. This provides an easy way of correlating the behavior of a parallel application with these measurements.

For HDEEM, the plugin initially starts the power measurement on the BMC and integrates it into the application trace after the application execution to avoid any perturbation of the application and system. This is done independently once per node, following the scalable approach of the parallel trace file. Similarly at smaller scale, the HAEC monitoring system is controlled from the plugin via the measurement daemon. A challenge for measurements at sub-millisecond time scales is the time synchronization of application events and measurement samples, which were generated on separate systems. Traditional synchronization schemes such as NTP do not provide sufficient accuracy. Hence, the plugin automatically executes a specific load pattern before and after the application execution, which is identified in the measurement data and used to align power measurements with the application trace.

4.5 Improving the energy accounting and power profiling of a scalable resource manager

Resource and job management systems (RJMS) can also take advantage of the HDEEM infrastructure. In this context, we have extended the open-source RJMS SLURM [16], which is specifically designed for the scalability requirements of state-of-the-art supercomputers.

We have implemented a plugin called `ipmi_hdeem` that leverages the high-resolution HDEEM measurements for job accounting and profiling. It enables accurate energy accounting without the need for frequent polling of the current power consumption by using the accumulated energy values provided by the FPGA. The consolidated energy for a job is reported by commands such as `sstat` and `sacct` and the consumed energy is finally stored in the SLURM database.

Furthermore, we have extended the HDF5 profiling framework of SLURM for flexibility and scalability purposes. In particular, the power profiling now supports multiple sensors (node, CPU, DRAM) and timestamps in microseconds to enable a higher precision in the profiling. The new `ipmi_hdeem` plugin is planned to be included in a future release of SLURM.

5 Calibration and verification

In order to obtain accurate and reliable measurements, any setup should be calibrated and verified. Due to aging of the hardware and environmental changes, calibration should be performed in regular intervals – for many instruments, the required calibration interval is one year. There are calibration services for portable commodity instruments, such as the LMG450 power analyzer or the National Instruments data acquisition cards. However, in case of complex and extensible measurement systems – especially within a large HPC system – regular calibration seems infeasible. Establishing that in an HPC context requires new processes and a business case.

5.1 Calibration and verification of a small scale measurement infrastructure

The design of the HAEC measurement system allows to digitally calibrate the measurement amplifiers. We performed the calibration of the amplifiers and shunts by connecting the sensor harness to a large variable load resistor. The calibration factor was adjusted such that

two calibrated multimeters – one measuring the current in serial, one measuring the output of the amplifier – were in agreement. The voltage amplification was calibrated similarly.

We also performed an extensive verification on all measurement channels. For the most important sensors, the two sockets, we used the LMG450 as reference measurement connected via additional Molex adapters to the same DC measurement domain.

There is still a limitation with this verification: The reference measurement provides a much lower readout rate and less accurate time synchronization, so that we can only compare average power measurements running for a longer time.

Verifying the other measurement channels is even more difficult, because no reference measurement at the same power domain is available. Therefore, a model is required to compensate for power conversion losses between the different measurements. Further details on this verification are described in [11].

5.2 Calibration of a large-scale measurement infrastructure

High accuracy is an important goal for the HDEEM measurement. Therefore, verification and calibration are an integral part of the design. In fact, the initial verification (see Figure 1) of the deployed HPC system has revealed errors outside the specified margin. The consequence of this finding was an in situ calibration of the blade sensors on site.

Using special calibration chassis is impractical when calibrating hundreds of nodes in a deployed system. Instead, we connect multimeters to the chassis, each of which contains 18 compute nodes divided into five slices with individual power supplies. Hence, five multimeters are needed to measure the current of each slice and a sixth to measure the common input voltage. The calibration setup is depicted in Figure 2.

To determine the calibration factors of each node, the slices are first measured with all the nodes powered off, providing a base power that is subtracted from further measurements. Then one node in each slice is powered on and the power consumption of each slice is measured. This process is automated using a control laptop which collects the power measurements from the multimeters and controls the nodes via an Ethernet connection.

For the calibration, each node executes two different processor loads, which are measured separately by both the multimeters and the embedded sensors. The discrepancy between the multimeter and embedded measurement at the two operating points provide an offset and factor, which are stored in the flash memory of the

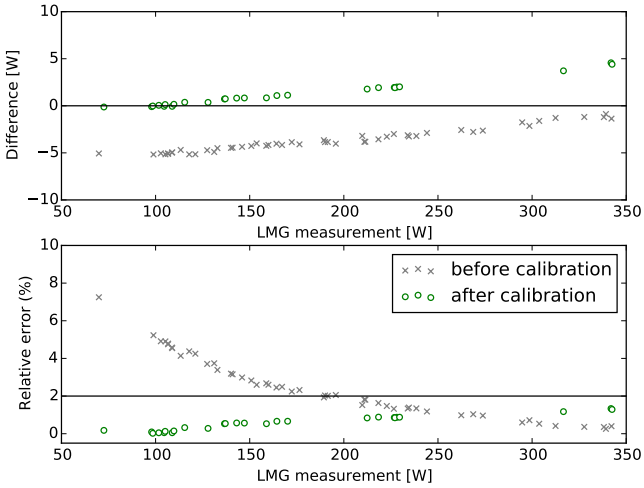

Fig. 1 Measurement error of the node power measurement of the same node before and after calibration

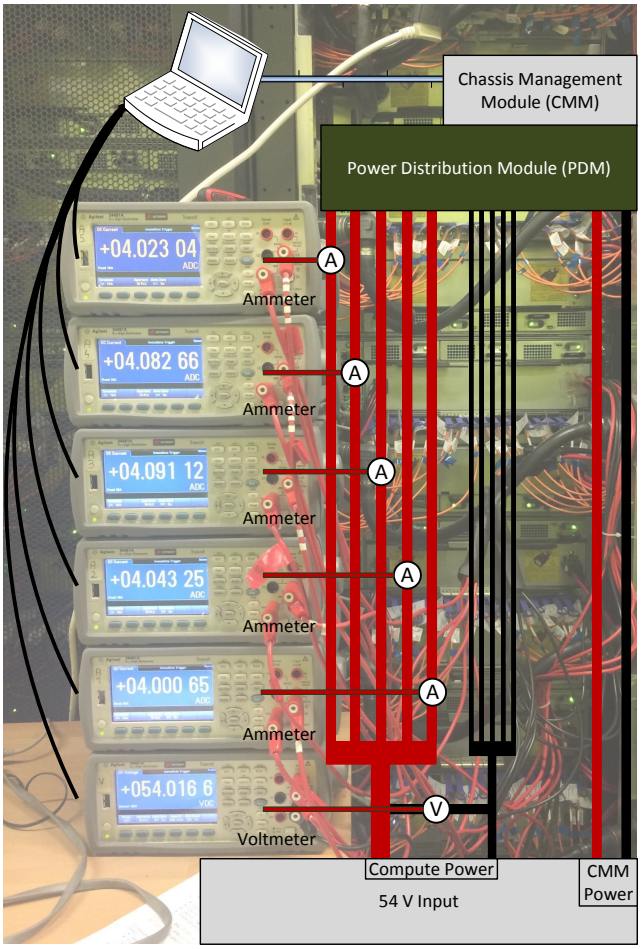

Fig. 2 Schema for the calibration of the HDEEM measurement infrastructure

corresponding node and are applied by the measurement FPGA to correct the raw sensor values.

5.3 Verification of a large-scale measurement infrastructure

Before and after the calibration, we subject the HDEEM measurements to extensive verification. For that, we use different measurement equipment and compute workloads that are specifically designed to exploit possible issues in the measurement. As a reference measurement, we use four channels of the LMG450 at the DC input to one chassis. The verification was performed on a sample of 36 nodes in two chassis. While there is no power conversion between the node-level HDEEM measurement and the reference measurement, the latter covers more components, e.g., the Infiniband switch in the chassis. Again, the base power is determined with all nodes powered down. We use FIRESTARTER [17] to generate workloads at different levels with different intervals of load changes by varying the number of active cores and core frequencies. Figure 1 shows the absolute and relative error before and after calibration for a node that showed particularly bad accuracy before the on-site calibration. The improvement from the calibration reduces the error below 2 % for this node. Over all tested nodes, all errors were below 3 % after calibration.

Furthermore, we created different alternating high-low load-patterns using FIRESTARTER. As described in Section 3.3, improper selection of sampling rates and analog filters can lead to aliasing effects. Over a wide range of workload frequencies, including the internal sampling rates, no aliasing from the HDEEM energy or power values could be observed.

6 Integrated application power and performance analysis

6.1 System description

The single-node fine-granular HAEC measurements were performed on a dual-socket system with Intel Xeon E5-2690 v3 processors running Ubuntu 16.04 Server. The system has been custom instrumented using the HAEC measurement system described in Section 3.

All scalable HDEEM measurements have been performed on a Bullx DLC B710/B720 based system installed at Technische Universität Dresden. The system contains different kinds of nodes, 1456 of which have two Intel Xeon E5-2680v3 CPUs running at 2.5 GHz and at least 64 GiB of main memory. The TDP of the Haswell processors is specified as 120 W.

6.2 Application description

The NAS parallel benchmarks are a set of parallel programs to evaluate the performance of HPC systems. We use a version that is parallelized using the thread-parallel OpenMP paradigm and the process parallel Message Passing Interface (MPI). To cover a wide range of systems, the benchmarks come with pre-defined working set classes to fit the tested system. In the following, we will show results from the parallel block tri-diagonal solver benchmark *bt* in version 3.3.

6.3 Application analysis

To illustrate the scalability of HDEEM, we run the benchmark on 1024 nodes with largest problem size

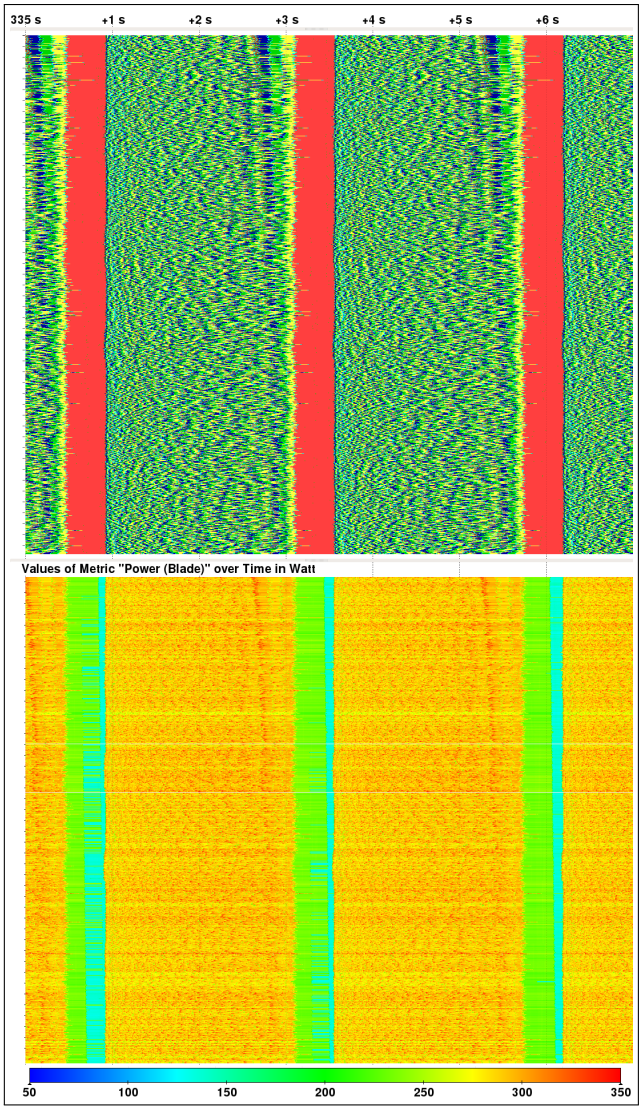

Fig. 3 HDEEM measurement of 1024 Bullx DLC B720 nodes executing a hybrid parallel program. The upper panel depicts the functions that are executed. Longest running functions are z_solve (yellow), y_solve (green), x_solve (dark blue), the parallel region spawned in `rhs.f` (brown), OpenMP synchronizations (cyan), and MPI synchronization (red). Zoomed into 7 seconds starting at second 335 that include two full iterations with the MPI-synchronizations in-between. Blade power consumption (displayed in the lower panel) drops when the processes synchronize and varies across nodes in the compute phases. Within the MPI iterations, the program executes iterations of rhs, x_solve,y_solve, z_solve, and add.

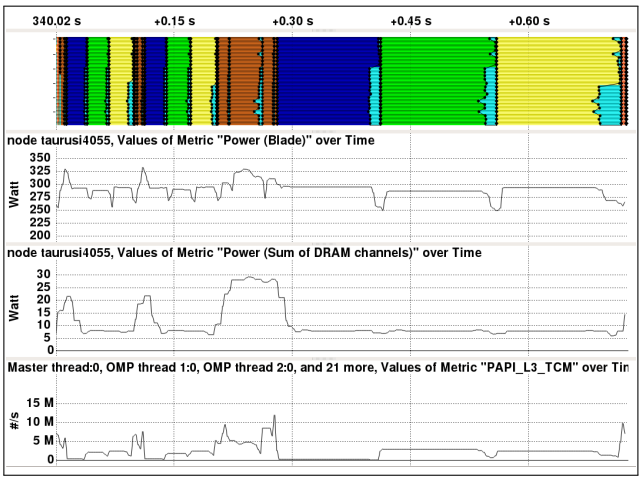

Fig. 4 Zoom into one node executing three inner iterations. Higher node power consumption in rhs regions relates to higher DRAM power consumption due to a higher last level cache miss rate (measured via PAPI_L3_TCM). The resulting power variation between regions with low cache miss rate and regions with high cache miss rate is up to 20 Watt per node (depending on the duration of the rhs region). The power consumption drops in OpenMP synchronization phases depending on the number of threads waiting for synchronization.

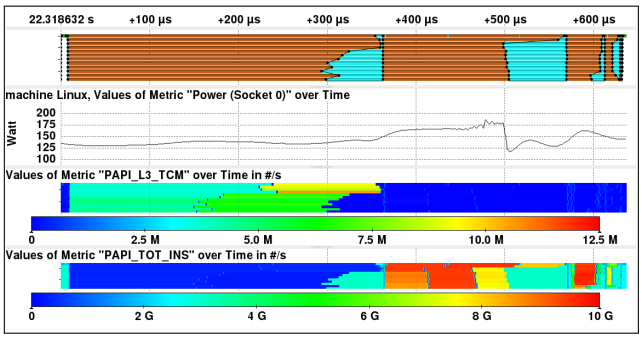

Fig. 5 Zoom into HAEC measurement of rhs region. Even though the threads synchronize at offset 300 µs, the power consumption does not drop as expected. This can be explained with an increased power consumption for DRAM accesses that compensate for possible processor power savings. After the first synchronization phase, the instruction throughput increases significantly, which results in a high processor power consumption, which drops when threads enter the second synchronization phase at 500 µs offset.

(class F). The resulting 360 GB trace is visualized with the performance analysis tool Vampir. Figure 3 shows seven seconds of the 677 seconds application runtime, illustrating two iterations of the solver with in-between MPI-synchronizations. This global view already shows some patterns, most notably a difference in power consumption between compute intensive and communication intensive regions (vertical) and a difference in power consumption between the nodes (horizontal).

Zooming in both vertically and horizontally, the trace reveals a difference between the executed functions inside the inner iteration. Figure 4 shows that rhs is the function with the highest power consumption. This is caused by increased DRAM activity triggered by last level cache (LLC) misses. Additionally, the power consumption drops in OpenMP synchronization phases depending on the number of threads that wait for synchronization. Finally, the regions x_solve, y_solve, and z_solve have different power demands related to their processor activity (not depicted in detail).

However, with an environment that supports a higher sampling rate for power measurement, even more details can be revealed. We measure the same benchmark with a smaller problem size (class C) on the HAEC test platform at a measurement rate of 500 kSa/s. We run a total of two MPI ranks, where each rank populates all cores of one socket with OpenMP threads. Figure 5 shows that the OpenMP loops within rhs have different power consumption characteristics and that the synchronization phases show diverging behaviors. We see two larger synchronization phases starting at 300 and 500 µs offset. For the first synchronization, the socket power consumption increases. This can be explained with an increased instruction throughput and DRAM activity caused by LLC misses. After the first synchronization, the socket power consumption increases significantly even though no memory accesses are visible. Thus, we can conclude that higher processor activity leads to increased processor power consumption. During the second synchronization, processor activity is reduced and DRAM activity remains low. The power consumption drops accordingly.

7 Conclusion

Correctly analyzing the power consumption characteristics of parallel applications requires highly specialized methodologies and hardware instrumentation. We have identified five key criteria to assess such solutions: temporal and spatial granularity, accuracy, scalability, and cost.

This work presents two distinct measurement solutions that push the boundaries of these criteria: The HAEC measurement infrastructure excels in terms of temporal resolution, whereas cost is rather high and scalability is naturally limited. The HDEEM approach maintains good temporal and spatial resolution at a level of scalability that can only be achieved with reasonable costs through tight integration into a volume class HPC server design. We address our choice of measurement domains and sensors, the analog signal processing, the digital data processing, collection and storage as well as the integration into resource managers and performance analysis tools.

Both solutions are designed to specifically address accuracy, which is often neglected. Careful design of the measurement hardware as well as extensive calibration procedures and rigorous verification are required to achieve this. Using specifically designed routines to provoke maximum errors in worst-case scenarios enables an end-to-end verification that ensures reliable measurement data for all real-world applications without making any additional assumptions.

Finally, we present possible usage scenarios for our two advanced power consumption measurement systems. We demonstrate the analysis of a 1024 node application run on the HDEEM system and its subsequent analysis using established and scalable performance analysis tools. We show that this solution can provide useful data at a resolution in the order of milliseconds and an isolation of CPUs and DRAM while maintaining full scalability. Moreover, we showcase that our HAEC solution pushes the temporal resolution even further to enable detailed analysis of individual program regions in the sub-millisecond range on a single compute node.

The HDEEM technology will be continued in the next generation of blades provided by Bull. It will embed more precise power sensors at the node and VR level, achieving 2 % accuracy without calibration after deployment. It remains for future work to build upon the presented measurement infrastructures to analyze and optimize the energy efficiency of the full software stack, including applications, frameworks, and programming models.

Acknowledgements This work is supported in parts by the German Research Foundation (DFG) in the Collaborative Research Center 912 "Highly Adaptive Energy-Efficient Computing", the Bundesministerium für Bildung und Forschung via the research project Score-E (BMBF 01IH13001), and Bull/Atos in the joint project "High Definition Energy Efficiency Monitoring" (HDEEM). The authors would like to thank Robin Geyer for his contribution on the HDEEM verification and Mario Bielert for improvements on the paper layout.

References

1. M. Knobloch, M. Foszczynski, W. Homberg, D. Pleiter, and H. Böttiger, "Mapping fine-grained power measurements to HPC application runtime characteristics on IBM POWER7," *Computer Science - Research and Development*, 2013.
2. G. Fourestey, B. Cumming, L. Gilly, and T. C. Schulthess, "First experiences with validating and using the cray power management database tool," *CoRR*, 2014.
3. D. Hackenberg, T. Ilsche, R. Schöne, D. Molka, M. Schmidt, and W. E. Nagel, "Power measurement techniques on standard compute nodes: A quantitative comparison," *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2013.
4. D. Hackenberg, R. Schöne, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer, "An Energy Efficiency Feature Survey of the Intel Haswell Processor," in *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International*, 2015.
5. R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. W. Cameron, "PowerPack: Energy profiling and analysis of high-performance systems and applications," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2010, DOI: 10.1109/TPDS.2009.76.
6. J. H. Laros III, P. Pokorny, and D. Debonis, "PowerInsight - A commodity power measurement capability," in *International Green Computing Conference (IGCC)*, 2013, DOI: 10.1109/IGCC.2013.6604485.
7. D. Bedard, M. Y. Lim, R. Fowler, and A. Porterfield, "PowerMon: Fine-grained and integrated power monitoring for commodity computer systems," in *IEEE SoutheastCon*, 2010, DOI: 10.1109/SECON.2010.5453824.
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," *Sixth International Green and Sustainable Computing Conference (IGSC)*, 2015, DOI: 10.1109/IGCC.2015.7393692.
9. Sandia National Laboratories, *Power API specification*, Std., Sep 2015. [Online]. Available: http://powerapi.sandia.gov/docs/PowerAPI_SAND.pdf
10. F. Almeida, J. Arteaga, V. Blanco, and A. Cabrera, "Energy measurement tools for ultrascale computing: A survey," *Supercomputing Frontiers and Innovations*, vol. 2, no. 2, 2015. [Online]. Available: http://superfri.org/superfri/article/view/45
11. T. Ilsche, D. Hackenberg, S. Graul, J. Schuchart, and R. Schöne, "Power measurements for compute nodes: Improving sampling rates, granularity and accuracy," in *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*, ser. Sixth International Green and Sustainable Computing Conference, IGSC, Dec. 2015.
12. D. Hackenberg, T. Ilsche, J. Schuchart, R. Schöne, W. E. Nagel, M. Simon, and Y. Georgiou, "Hdeem: High definition energy efficiency monitoring," in *International Workshop on Energy Efficient Supercomputing (E2SC)*. IEEE Press, 2014.
13. "HDEEM library reference guide," http://www.bull.com/download-hdeem-library-reference-guide, Bull Atos Technologies, Tech. Rep., 2016.
14. M. Kluge, D. Hackenberg, and W. E. Nagel, "Collecting distributed performance data with dataheap: Generating and exploiting a holistic system view," *Procedia Computer Science*, 2012.
15. A. Knüpfer, C. Rössel, D. Mey, S. Biersdorff, K. Diethelm, D. Eschweiler, M. Geimer, M. Gerndt, D. Lorenz, A. Malony, W. E. Nagel, Y. Oleynik, P. Philippen, P. Saviankou, D. Schmidl, S. Shende, R. Tschüter, M. Wagner, B. Wesarg, and F. Wolf, "Score-P: A joint performance measurement run-time infrastructure for Periscope, Scalasca, TAU, and Vampir," in *Tools for High Performance Computing 2011*, H. Brunst, M. S. Müller, W. E. Nagel, and M. M. Resch, Eds. Springer Berlin Heidelberg, 2012, pp. 79–91.
16. Y. Georgiou, T. Cadeau, D. Glesser, D. Auble, M. Jette, and M. Hautreux, "Energy accounting and control with slurm resource and job management system," in *Distributed Computing and Networking*, M. Chatterjee, J.-n. Cao, K. Kothapalli, and S. Rajsbaum, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 96–118.
17. D. Hackenberg, R. Oldenburg, D. Molka, and R. Schöne, "Introducing FIRESTARTER: A processor stress test utility," 2013.