Comput Sci Res Dev (2014) 29:211–219
DOI 10.1007/s00450-013-0245-5

SPECIAL ISSUE PAPER

Mapping fine-grained power measurements to HPC application runtime characteristics on IBM POWER7

Michael Knobloch · Maciej Foszczynski · Willi Homberg · Dirk Pleiter · Hans Böttiger

Published online: 25 July 2013

Abstract Optimization of energy consumption is a key issue for future HPC. Evaluation of energy consumption requires a fine-grained power measurement. Additional useful information is obtained when performing these measurements at component level. In this paper we describe a setup which allows to perform fine-grained power measurements up to a 1 ms resolution at component level on IBM POWER (IBM and POWER are trademarks of IBM in USA and/or other countries.) machines. We further developed a plug-in for VampirTrace that allows us to correlate these power measurements with application performance characteristics, e.g. obtained by hardware performance counters. This environment enables us to generate both power and performance profiles. Such profiles provide valuable input to develop future strategies for improving workload-driven energy usage per performance. We show in comparison with power profiles of coarser granularity that these fine-grained measurements are necessary to capture the dynamics of power switching.

M. Knobloch (✉) · M. Foszczynski · W. Homberg · D. Pleiter
Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich, 52425 Jülich, Germany
e-mail: m.knobloch@fz-juelich.de

M. Foszczynski
e-mail: m.foszczynski@fz-juelich.de

W. Homberg
e-mail: w.homberg@fz-juelich.de

D. Pleiter
e-mail: d.pleiter@fz-juelich.de

H. Böttiger
IBM Deutschland Research & Development GmbH, Schönaicher Str. 220, 71032 Böblingen, Germany
e-mail: BOETTIG2@de.ibm.com

Keywords Energy · Performance · Power consumption · POWER7

1 Introduction

Rising IT spending on power has increased the awareness and need for monitoring, management, and optimization of data-center energy consumption. HPC centers have additional performance-based constraints and account for disproportionally large power/energy costs. Among the numerous challenges which exist in expanding systems to Exascale, the capping of the energy consumption at a reasonable power limit, say 20 MW, is probably the most important one.

Although computing systems offer an increasingly sophisticated set of power measurement and management capabilities, on most platforms fine-grained power measurements are difficult or impossible without modifying the hardware. In this paper, we focus on IBM systems based on the POWER7 processor. These systems are provided with on-board power measurement circuits to measure the power consumed by the full system, processor socket, memory subsystem, I/O sub-system and the fans [1]. Additionally, the power consumed in different parts of the POWER7 processor can be estimated using a hardware supported power proxy. This power proxy translates information from activity monitors into a power estimate using a programmable weight factor [2]. Information from various sensors is collected by a dedicated microcontroller, the Thermal Power Management Device (TPMD). This device is also used to implement a given power policy and management direction. It may, for instance, reduce the processor frequency to save power at the expense of system performance.

Applications in the HPC space tend to be designed and tuned to maximize performance with no consideration for

energy efficiency. Programming approaches can mask real utilization of resources like CPU or memory (e.g., wait loop in communication progress function), and a load balancing style of programming can interfere with autonomous hardware frequency scaling. Therefore, one of the challenges is to enhance methods and policies in this area to exploit energy management mechanisms. To identify optimization potential, one has to study the application's energy profiles as well as the utilization of the different system resources.

One of the challenges to obtain reliable energy profiles is to translate the power consumption $P(t)$ measured at time t into energy $E(t_0, t)$ consumed since time t_0, e.g. the time when program execution started. The energy is given by

$$E(t_0, t) = \int_{t_0}^{t} d\tau\, P(\tau). \tag{1}$$

Since power is measured at discrete points in time t_i, the integral has to be approximated by a sum

$$E(t_0, t) \simeq \sum_{k=1}^{N} \Delta t\, P(t_0 + k\Delta t), \tag{2}$$

where for simplicity we assumed $t_i = t_0 + k\Delta t$ with $\Delta t = (t - t_0)/N$. For this approximation to be good, Δt should be small compared to the time scale on which $P(t)$ changes.

To read the information from the TPMD, we use an IBM internal tool called Amester (Automated Measurement of Systems for Temperature and Energy Reporting), which provides an external service to collect the power consumption data. We used VampirTrace [7] to trace the performance of the application, and developed a plugin which queries Amester to add power measurement information to the performance traces.

After discussing related work, we describe our measurement setup in Sect. 3. Then we present the applications we used and the analysis results in Sect. 4. Finally we conclude the paper and give an outlook on future work in Sect. 5.

2 Related work

A large variety of papers have been published analysing the power consumption of individual components using synthetic workloads. However, far less information is available on the power consumption at component level for HPC production workloads.

In [6] the power consumption of a Cray XT4 system is studied at node and system level for a set of application based benchmarks like the NAS Parallel Benchmarks. A different approach was taken in [5] where the authors analysed rack level power measurement data collected for 12,500 jobs on a production Blue Gene/P system.

In [3] the power profiling infrastructure PowerPack is described. This infrastructure targets the analysis of parallel applications and also links power to performance measurements [10]. This setup however requires significant modifications of the hardware. Furthermore, a critical analysis of the conversion of power measurements into estimates of the energy consumption is lacking.

In recent years, the hardware support for power measurements has been driven by the need of monitoring power to ensure a certain power envelope not being exceeded. This may, e.g., be mandatory in case of high-density designs where components at high load may generate more heat than the cooling system is able to remove. In [9] a feedback control mechanism is described, which allows to operate the system in the highest performance state at a fixed power constraint. This requires precise power information to be periodically retrieved.

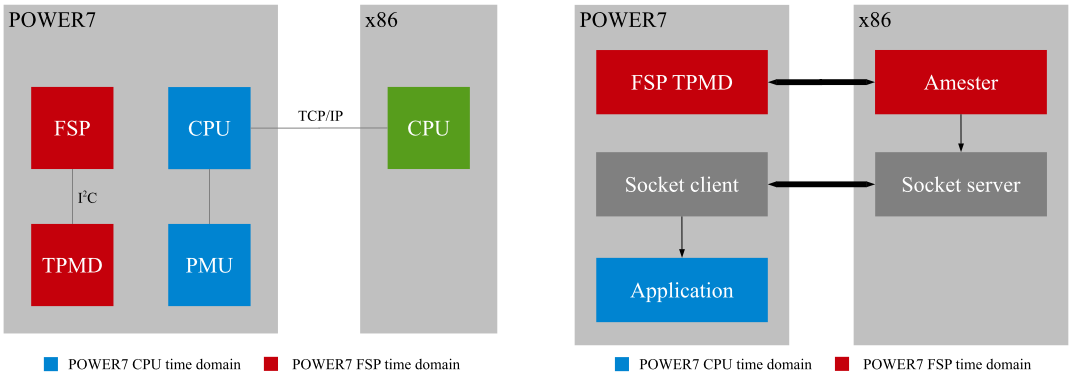
The authors of [1] present results of an investigation where they use the power monitoring and management features of POWER7-based systems in order to reduce the power consumption. The power measurements are used to fit the parameters of a heuristic model, which describes the power consumption of an application as a function of the frequency. The analysis is, however, restricted to a selection of SPEC CPU2006 workloads and does not include full applications.

Power consumption measurements often require a dedicated hardware setup. Even if power measurement capabilities are integrated into HPC production environments, then the data is either not accessible to the user or the user has to make significant efforts to analyse the data. Initial attempts to integrate power measurements into widely used performance analysis tools, e.g. Vampir, are reported by the eeClust project [11]. In this project, x86-based systems with external power meters were used.

3 Measurement setup

The heart of our measurement setup is a POWER7 processor-based server, an IBM Power 720 Express, on which the application is executed. The POWER7 processor is a recent generation server processor in the IBM POWER family. The main features of our machine are:

- Single 4-core 3.0 GHz processor (Pseries, 8202-EeB)
  - 96 GFLOPS peak
  - 4 SMT threads per core
  - Execution units per core
    - 2 fixed-point units
    - 2 load/store units
    - 4 double-precision floating-point units
    - 1 vector unit supporting VSX
  - 32 + 32 kB L1 instruction and data cache per core
  - 256 kB L2 cache per core
  - 16 MB shared L3 cache



(a) Hardware setup (b) Software setup

Fig. 1 Hardware and software setup for Amester measurements. It highlights the different time domains from which timestamps are obtained

– 16 GB memory
– Dual 300 GB 10 K RPM SAS disks
– TPMD (Thermal and Power Management Device)

An additional system is used to run the power measurement service (called Amester) without interfering with the workload execution on the POWER7-based server.

The POWER7-based server is connected to a Power Distribution Unit (PDU) which provides us with an estimate of the power consumed by the server at its power inlet. The PDU which we use, a Raritan DPXS 12A-16, only allows for relatively coarse power measurements with a granularity of 3 s and a precision of about 5 %. It should however be noted that power consumption at system inlet is expected to change at a much slower rate. The data is stored in an SQL database and is used as consistency check for power measurements with Amester. The PDU values are expected to be larger, e.g. due to inefficiencies of the power supplies in the POWER7 system. We found the difference to agree with the specified efficiency of the power supplies.

Fine grained power measurements on POWER7 are possible with a software tool called Amester, which communicates with the TPMD of the POWER7-based server via the Flexible Service Processor (FSP) of the POWER7. It sends commands to the FSP which returns the requested data. Amester can query counters with a sampling rate as low as 1 ms. Using this tool it is possible to retrieve, among others, data about power consumption of the full node, the processor, the memory, the I/O subsystem, and the fans. Amester is executed on a separate x86 server, i.e. it allows in principle for an intrusion free power measurement. The x86 server and the POWER7 system communicate via TCP/IP over a socket connection. Figure 1 shows the hardware (Fig. 1(a)) and software (Fig. 1(b)) setup we used for our experiments.

To match the Amester measurements and the performance data received on the POWER7 processor, a timestamp synchronization issue needs to be resolved. Timestamps of measurements taken by VampirTrace come from the POWER7 CPU time domain. On the other hand, samples gathered by Amester are marked with timestamps originating from the POWER7's FSP. The FSP's timer is a millisecond counter, incremented from the start-up of the system, contrary to the CPU clock. Therefore, in order to correlate Amester's fine-grained measurements with application performance characteristics, time offset calculation mechanism is required.

For this purpose, a simple micro-benchmark has been implemented, together with a suite of post-processing mechanisms. The purpose of the benchmark is to provide a number of IPS (Instructions Per Second) peaks, marked both by POWER7 CPU and FSP timestamps. CPU timestamps are taken during the micro-benchmark runtime, directly before and after each iteration, and printed out as the output of the application. In the same time, Amester is used to gather performance statistics from the TPMD, marked with FSP timestamps. Afterwards, Amester output is processed and beginnings and ends of peaks are recognized. Here a number of mechanisms to ensure the accuracy of data recognition has been implemented, i.e. filtering unwanted values by threshold, validating peak data series length, a number of cross-checks with values provided by the CPU, and more. Figure 2 shows a sample output of the offset detection. The final value for offset is given as an average of offsets for the start and end of each peak in the benchmark. This value is later used in the Amester plugin for VampirTrace.

In principle this approach should scale to multiple nodes as power data generation and communication do not interfere with the application run. However, the timestamp synchronization would have to be done for each node separately.

VampirTrace is a library to generate event-based trace files from instrumented applications. The VampirTrace workflow as we use it is shown in Fig. 3.

We developed a plugin for the VampirTrace plugin counter interface [12] to merge the counters provided by Amester into the OTF trace file generated by VampirTrace.

As the Amester measurement is out-of-band, we choose a post-mortem plugin where the values are merged into the trace file at the finalization of the measurement, i.e. after the application generated its results. This keeps the additional measurement overhead at a negligible level. Additional hardware performance counter values can be obtained with the PAPI library [14]. The resulting trace file can be visualized with the Vampir trace file visualizer [7].

The sensors that the VampirTrace plugin queries using Amester are listed in Table 1. The counter names we use for the Amester plugin are of the form P7_IPS and P7_POWER{_COMPONENT}_RESOLUTION.

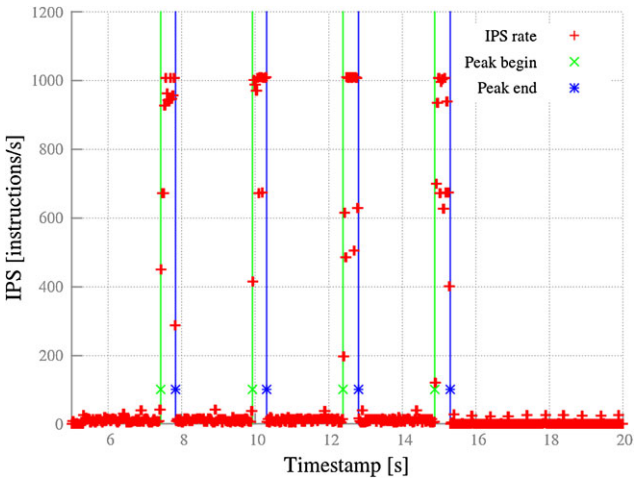


Fig. 2 Sample output of the POWER7 CPU—FSP offset detection. It shows the four peaks in IPS and the calculated beginning and end of the computation phases

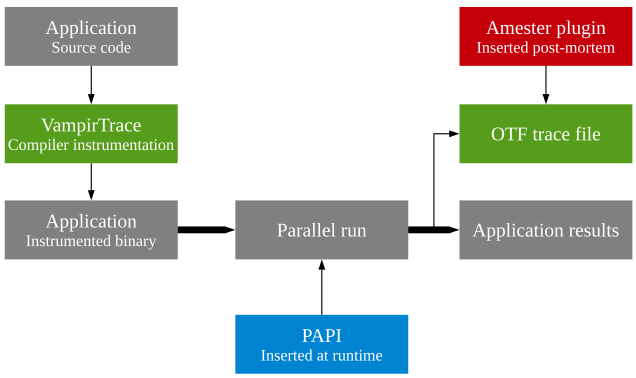


Fig. 3 VampirTrace Workflow for our experiments

4 Applications and analysis results

For our experiments we selected two codes developed at JSC, PEPC and MP2C, that also run—in different configurations from the ones we used here—on JSC's Blue Gene/Q supercomputer on several thousand cores.

PEPC (Pretty Efficient Parallel Coulomb solver) [15] is a mesh-free tree-code for computing long-range forces, e.g. Coulomb or gravitational forces, in N-body particle systems. The code was initially developed to study problems in plasma physics. It can, however, also be used for problems from other research areas like astrophysics and biophysics. By using successively larger multi-pole groups of distant particles, the computational complexity of the long-range force computations is reduced to $O(N \log N)$ which is a key requisite to achieve a very high scalability of the code. The code is written in Fortran90 and parallelized using MPI and Pthreads. In our test cases an MPI only version was used.

Figure 4 shows a Vampir screenshot of a PEPC run with 4 processes capturing power measurements of the total power consumption as well as CPU and memory power consumption. The top left 'master timeline' shows the program activity on a per-process base. Below are the 'counter timelines', showing the development of the different counters. Since there is more than one sample per pixel, Vampir shows for each counter the maximum value (the upper line), the minimum value (the lower line) and the average (the middle line). On the right side, some statistical information is displayed. The 10 iterations of the test run are clearly distinguishable.

A more detailed view of one iteration is shown in Fig. 5. The resolution is now fine enough for Vampir to show the

Table 1 Counters which are retrieved from Amester by the VampirTrace plugin

Sensor name	Units	Time scale	Description
PWR1MS	W	Instantaneous reading	Total power consumption for the node
PWR1MSP0	W	Instantaneous reading	Power consumption for processor #0
PWR1MSMEM0	W	Instantaneous reading	Power consumption for memory of processor #0
PWR32MS	W	avg. over last 32 ms	Total power consumption for the node
PWR32MSP0	W	avg. over last 32 ms	Power consumption for processor #0
PWR32MSMEM0	W	avg. over last 32 ms	Power consumption for memory of processor #0
IPS32MS	Mips	Every 32 ms	Average number of instructions per second

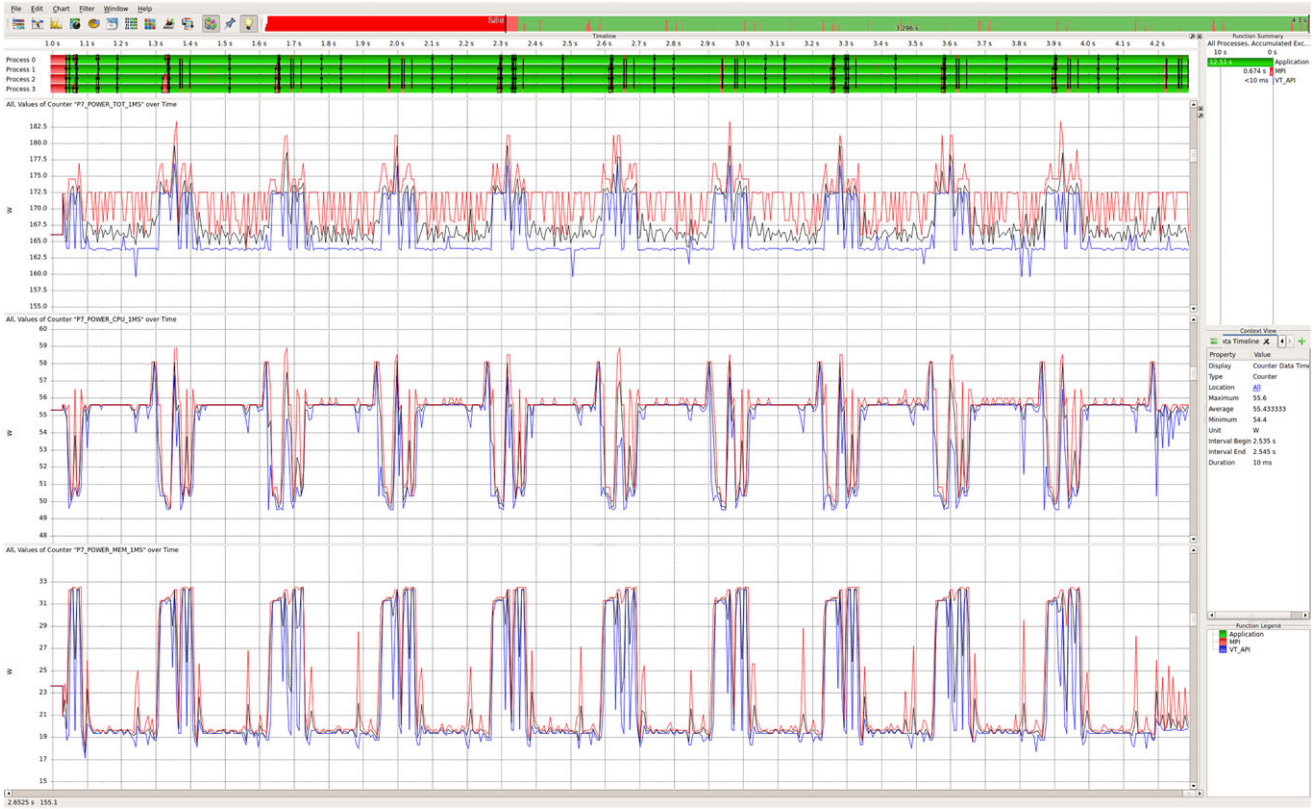


Fig. 4 Vampir screenshot of a PEPC run showing power measurements at system level (*top*) as well as for CPU (*middle*) and memory (*bottom*). For each measurement it shows maximum (*upper line*), average (*middle line*), and minimum (*lower line*)

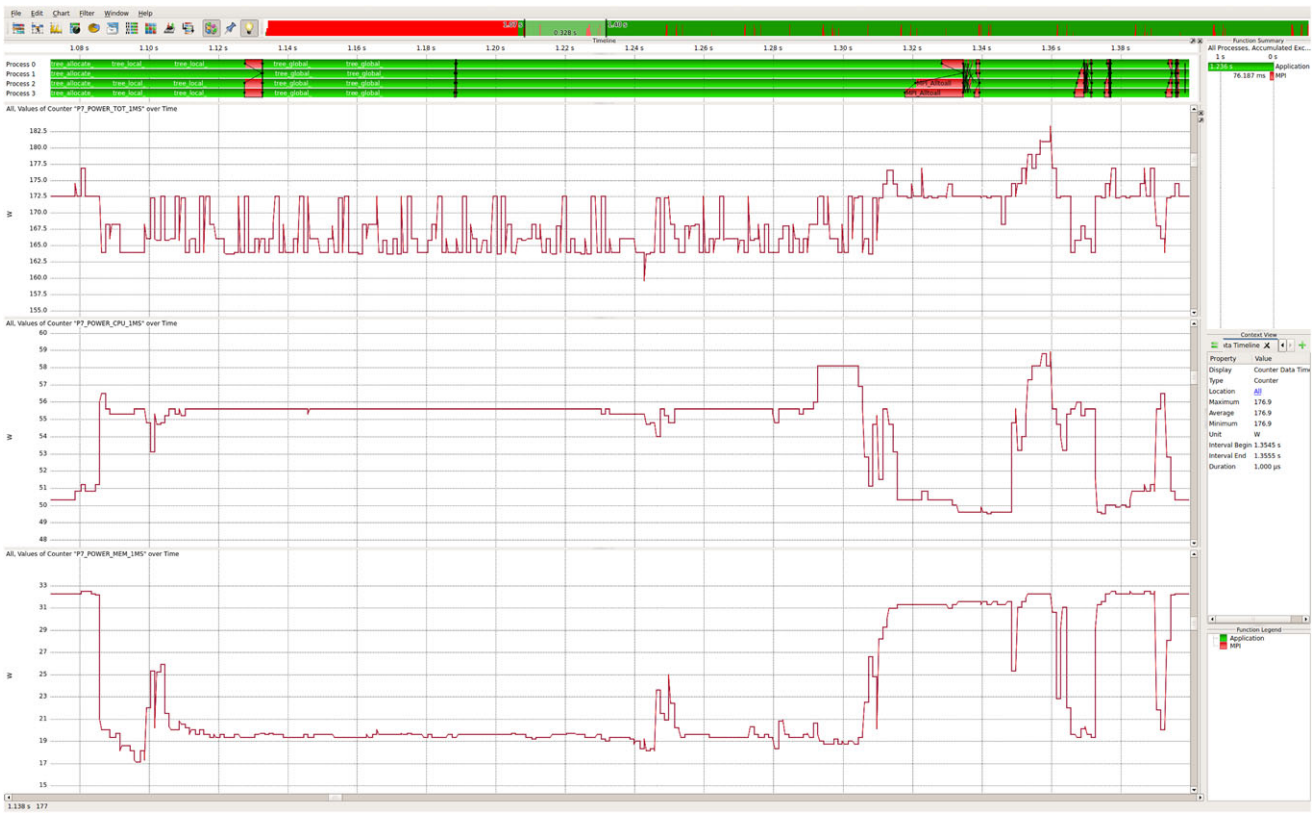


Fig. 5 Vampir screenshot of one iteration of a PEPC run showing that there are significant changes in power consumption at millisecond scale

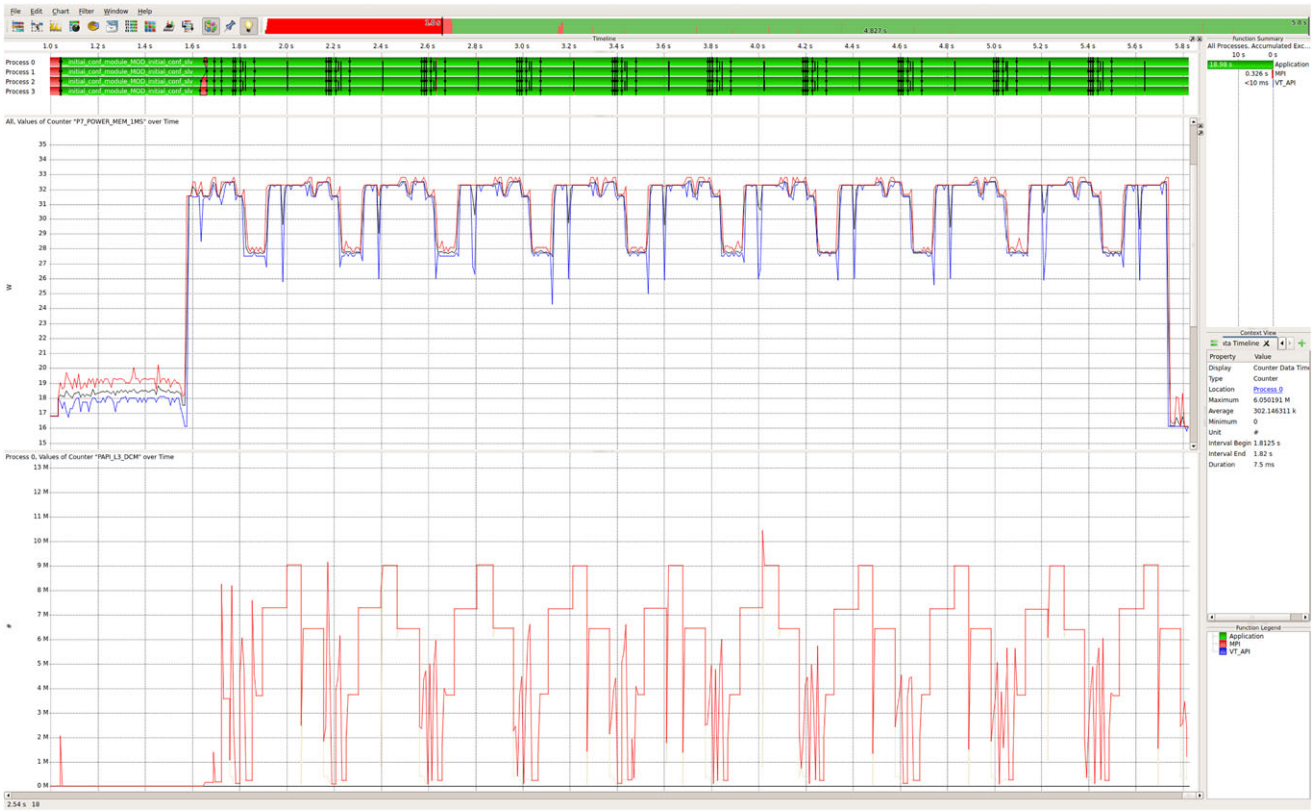


Fig. 6 Vampir screenshot of an MP2C run showing memory power measurements and L3 cache misses

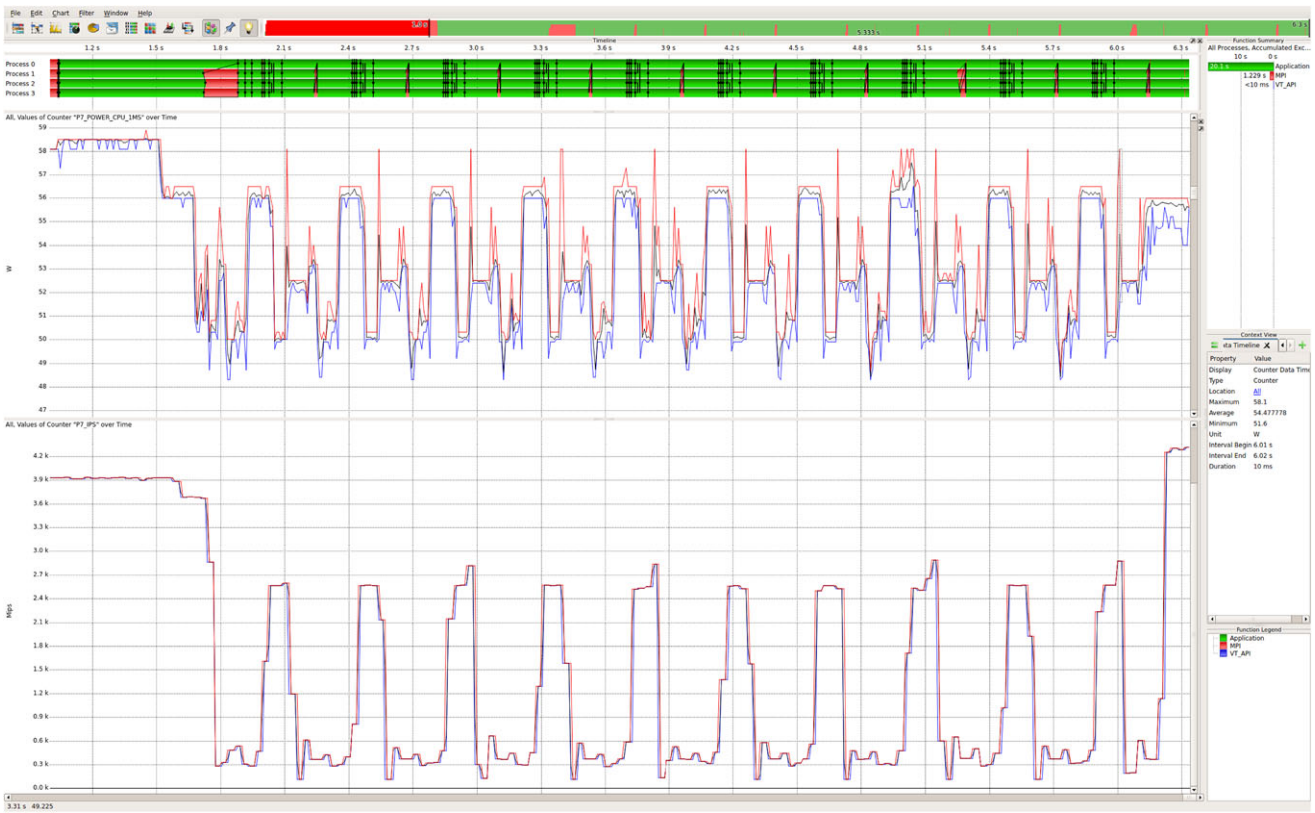
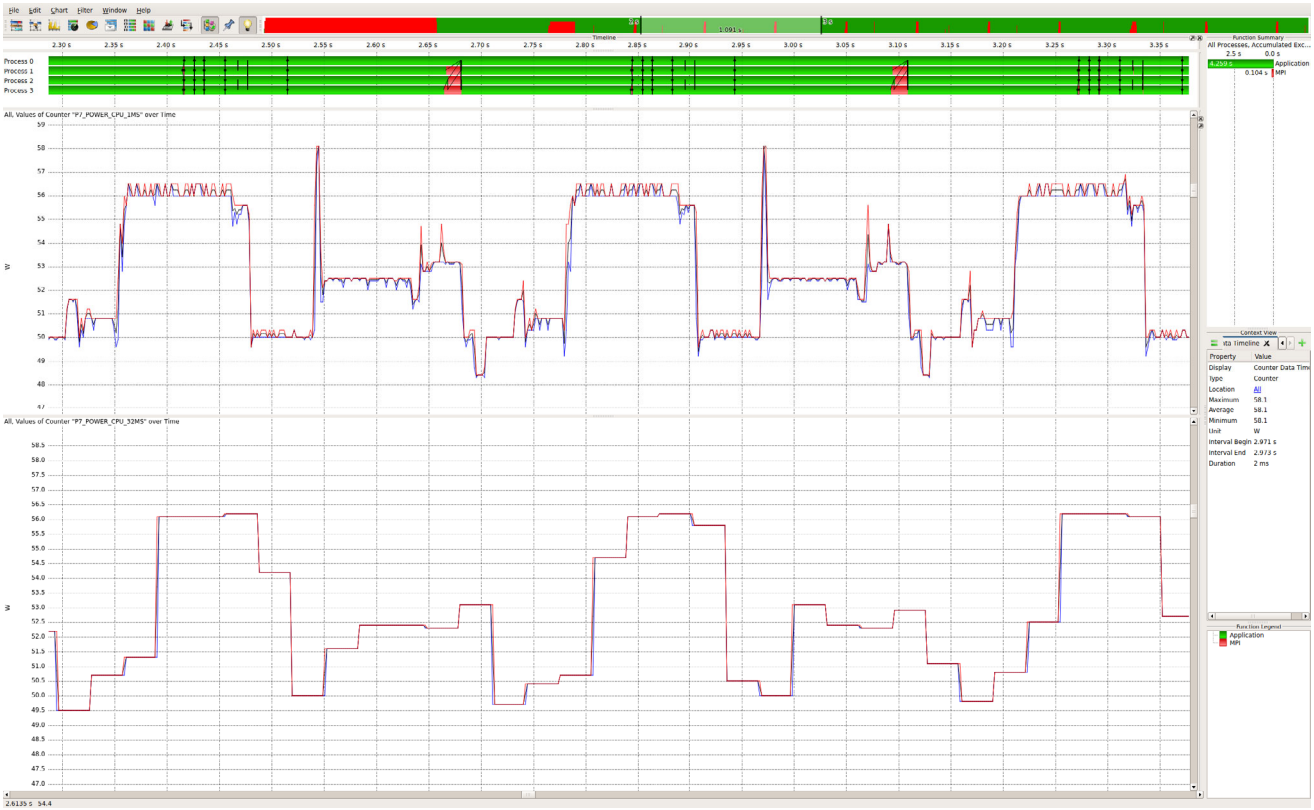


Fig. 7 Vampir screenshot of an MP2C run showing detailed CPU power consumption and IPS rate



(a) CPU

(b) MEMORY

Fig. 8 Comparison of power measurements with 1 ms resolution (*upper part*) and 32 ms resolution (*lower part*) for CPU (**a**) and memory (**b**)

measured values of the counters instead of the statistical information as in Fig. 4. We see that significant changes in power consumption occur at millisecond level for all components.

MP2C (Massively Parallel Multi-Particle Collision) [13] is a code for simulating fluids with solvated particles. It couples Multi-Particle Collision Dynamics (MPC) with Molecular Dynamics (MD). The former is a simulation technique where particles are simulated at mesoscale. By coupling MPC to MD, hydrodynamic interactions between solvated molecules can be taken into account. The code is written in Fortran90 and parallelized using MPI and OpenMP.

Figures 6 and 7 show Vampir screenshots of a MP2C run with 4 processes with component power measurements and runtime characteristics. The 10 iterations of the test case are easily detectable in both figures. Figure 6 plots the power consumption of the memory subsystem against the L3 data cache misses, and Fig. 7 shows the CPU power consumption and instructions per second (IPS). These values correlate quite nicely, although some peaks in the CPU power consumption can not be spotted in the IPS counter line. This might be related to the IPS being averaged over a time period of 32 ms, which might miss some details.

Figure 8 shows the comparison of power measurements with 1 ms resolution and with 32 ms resolution for CPU and memory. The 32 ms measurements internally accumulate 32 1 ms measurements and average them. Thus, they flatten some details that can be seen in the 1 ms measurements, yet result in the same integrated energy consumption for the whole application run. However, to calculate the energy consumption of shorter code parts, fine-grained measurements are beneficial.

5 Conclusion and outlook

In this paper we presented a setup that allows us to obtain fine-grained power measurements on the IBM POWER7 platform and correlate these values to application performance data. We showed that coarse-grained power measurements flatten the dynamics in power consumption on all components.

The next step is to adapt that workflow to work with the new Score-P measurement system [8], a unified measurement system used by multiple tools, e.g. Vampir and Scalasca [4].

Further, we are developing a model for the energy consumption on component level based on hardware performance counters, for which such fine-grained power measurements are beneficial. Such models can then be used by all kinds of tools, even profile based tools.

Acknowledgements These results were obtained using the IBM Automated Measurement of Systems for Temperature and Energy Reporting software. We gratefully acknowledge useful discussions with and support by Charles Lefurgy from IBM Research in Austin, TX. This work was funded by the state of North Rhine-Westfalia ("Anschubfinanzierung zum Aufbau des Exascale Innovation Center (EIC)")

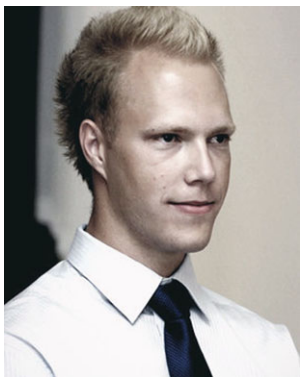
References

1. Brochard L, Panda R, Vemuganti S (2010) Optimizing performance and energy of hpc applications on power7. Comput Sci Res Dev 25(3–4):135–140. http://www.springerlink.com/index/10.1007/s00450-010-0123-3
2. Floyd M, Allen-Ware M, Rajamani K, Brock B, Lefurgy C, Drake A, Pesantez L, Gloekler T, Tierno J, Bose P, Buyuktosunoglu A (2011) Introducing the adaptive energy management features of the power7 chip. IEEE MICRO 31(2):60–75. doi:10.1109/MM.2011.29
3. Ge R, Feng X, Song S, Chang HC, Li D, Cameron K (2010) PowerPack: energy profiling and analysis of high-performance systems and applications. IEEE Trans Parallel Distrib Syst 21(5):658–671. doi:10.1109/TPDS.2009.76
4. Geimer M, Wolf F, Wylie BJN, Ábrahám E, Becker D, Mohr B (2010) The scalasca performance toolset architecture. Concurr Comput, Pract Exp 22(6):702–719. doi:10.1002/cpe.1556
5. Hennecke M, Frings W, Homberg W, Zitz A, Knobloch M, Böttiger H (2012) Measuring power consumption on ibm blue gene/p. Comput Sci Res Dev. doi:10.1007/s00450-011-0192-y
6. Kamil S, Shalf J, Strohmaier E (2008) Power efficiency in high performance computing. In: IEEE international symposium on parallel and distributed processing, pp 1–8
7. Knüpfer A, Brunst H, Doleschal J, Jurenz M, Lieber M, Mickler H, Müller MS, Nagel WE (2008) The vampir performance analysis tool-set. In: Tools for high performance computing. Proceedings of the 2nd international workshop on parallel tools. Springer, Berlin, pp 139–155
8. Knüpfer A, Rössel C, an Mey D, Biersdorff S, Diethelm K, Eschweiler D, Geimer M, Gerndt M, Lorenz D, Malony AD, Nagel WE, Oleynik Y, Philippen P, Saviankou P, Schmidl D, Shende SS, Tschüter R, Wagner M, Wesarg B, Wolf F (2012) Score-P—A joint performance measurement run-time infrastructure for periscope, scalasca, TAU, and vampir. In: Proc. of 5th parallel tools, Workshop, 2011, Dresden, Germany. Springer, Berlin, pp 79–91
9. Lefurgy C, Wang X, Ware M (2007) Server-level power control. In: Proceedings of the IEEE international conference on autonomic computing (ICAC)
10. Lively C, Wu X, Taylor V, Moore S, Chang HC, Cameron K (2011) Energy and performance characteristics of different parallel implementations of scientific applications on multicore systems. Int J High Perform Comput Appl 25(3):342–350. doi:10.1177/1094342011414749
11. Minartz T, Molka D, Knobloch M, Krempel S, Ludwig T, Nagel WE, Mohr B, Falter H (2012) Eeclust: energy-efficient cluster computing. In: Bischof C, Hegering HG, Nagel WE, Wittum G (eds) Competence in high performance computing 2010. Springer, Berlin, pp 111–124. doi:10.1007/978-3-642-24025-6_10
12. Schöne R, Tschüter R, Ilsche T, Hackenberg D (2011) The vampirtrace plugin counter interface: introduction and examples. In: Proceedings of the 2010 conference on parallel processing, Euro-Par 2010. Springer, Berlin, pp 501–511
13. Sutmann G, Westphal L, Bolten M (2010) Particle based simulations of complex systems with mp2c: hydrodynamics and electrostatics. AIP Conf Proc 1281(1):1768–1772. doi:10.1063/1.3498216

14. Terpstra D, Jagode H, You H, Dongarra J (2010) Collecting performance data with papi-c. In: Müller MS, Resch MM, Schulz A, Nagel WE (eds) Tools for high performance computing 2009. Springer, Berlin, pp 157–173. doi:10.1007/978-3-642-11261-4_11

15. Winkel M, Speck R, Hübner H, Arnold L, Krause R, Gibbon P (2012) A massively parallel, multi-disciplinary Barnes–hut tree code for extreme-scale n-body simulations. Comput Phys Commun 183(4):880–889. doi:10.1016/j.cpc.2011.12.013

Michael Knobloch received his diploma in Mathematics from Technische Universität Dresden, Germany in 2008. Since 2009 he holds a position as researcher at the Jülich Supercomputing Centre (JSC) of Forschungszentrum Jülich GmbH in the performance analysis group led by Dr. Bernd Mohr. His research interests include energy and performance analysis of HPC applications and the development of corresponding tools. He is part of several Exascale efforts at JSC.



Maciej Foszczynski received a M.S. degree in Computer Science from Wroclaw University of Technology, Poland, in 2011. Between 2011 and 2012, he was a part of IBM's Server System Operations group. In 2012, he has joined the Jülich Supercompuing Centre (JSC) of Forschungszentrum Jülich GmbH, as a researcher in the Exascale Innovation Centre. Since then, he has been involved in work on energy analysis of HPC systems and research of the active storage technology towards the future Exascale architectures.

Willi Homberg started in 1978 at the Central Institute for Applied Mathematics (ZAM) of Research Centre Jülich where he was primarily engaged in the education of mathematical-technical assistants. In 1990, he moved to data center operations. He was responsible for UNIX servers and Linux compute clusters. Since 2006 he works in the department Technology at Jülich Supercomputing Centre (JSC). Currently he is leader of the work package energy efficiency in "Exascale Innovation Center", a cooperation of IBM and JSC aiming at the development of future exascale supercomputer systems.

Dirk Pleiter is a Research Staff Member at the Jülich Supercomputing Centre and professor for theoretical physics at the University of Regensburg. He received his Ph.D. degree in physics from the Free University of Berlin in 2000, and then he joined the John-von-Neumann Institute for Computing as a postdoc and later became Research Staff Member at the particle accelerator laboratory Deutsches Elektronen-Synchrotron (DESY).

Hans Böttiger joined IBM in 1973 after receiving a B.S. degree in electrical engineering from the University of Esslingen, Germany. He has held various technical leadership positions in software, operating systems, and hardware development for mainframes, as well as in performance analysis for BI systems, compilers and blade computers. He currently works on research topics for next generation HPC systems.