Comput Sci Res Dev
DOI 10.1007/s00450-011-0192-y

SPECIAL ISSUE PAPER

Measuring power consumption on IBM Blue Gene/P

Michael Hennecke · Wolfgang Frings · Willi Homberg · Anke Zitz · Michael Knobloch · Hans Böttiger

Abstract Energy efficiency is a key design principle of the IBM Blue Gene series of supercomputers, and Blue Gene systems have consistently gained top GFlops/Watt rankings on the Green500 list. The Blue Gene hardware and management software provide built-in features to monitor power consumption at all levels of the machine's power distribution network. This paper presents the Blue Gene/P power measurement infrastructure and discusses the operational aspects of using this infrastructure on Petascale machines. We also describe the integration of Blue Gene power monitoring capabilities into system-level tools like `LLview`, and highlight some results of analyzing the production workload at Research Center Jülich (FZJ).

Keywords Blue Gene · Energy efficiency · Power consumption

1 Introduction and background

Power consumption of supercomputers is becoming increasingly important: Since 2007, the Green500 list publishes supercomputer rankings based on the *Flops/Watt* metric. The Top10 supercomputers on the November 2010 Top500 list [1] alone (which coincidentally are also the 10 systems with an R_{peak} of at least one PFlops) are consuming a total power of 33.4 MW [2]. These levels of power consumption are already a concern for today's Petascale supercomputers (with operational expenses becoming comparable to the capital expenses for procuring the machine), and addressing the energy challenge clearly is one of the key issues when approaching Exascale.

While the Flops/Watt metric is useful, its emphasis on LINPACK performance and thus computational load neglects the fact that the energy costs of memory references and the interconnect are becoming more and more important [3]. It has also been pointed out that a stronger focus on optimizing *time to solution* will likely result in a different ranking of competing algorithms to solve a given scientific problem than when solely optimizing for Flops/Watt [4]. It is therefore important to better understand the energy characteristics of current production workloads on Petascale systems. Those insights can then be used as input to future hardware design as well as for algorithmic optimizations with respect to overall energy efficiency.

In this work we focus on the IBM* Blue Gene* series of supercomputers. The guiding design principles for Blue Gene are *simplicity*, *efficiency*, and *familiarity* [5]. Regarding energy *efficiency*, the key feature of Blue Gene is its judiciously chosen low-frequency, low-voltage design which results in both high-performance and highly energy-efficient supercomputers. Blue Gene/L [6–8] and Blue Gene/P [9] systems have consistently gained top MFlops/Watt rankings on the Green500 list, and an early prototype of the next generation Blue Gene/Q system has recently set a new record at 2.1 GFlops/Watt [2]. One important aspect of the *familiarity* design principle is that the well established MPI parallel programming paradigm on homogeneous nodes is main-

* IBM, Blue Gene, DB2, POWER and PowerXCell are trademarks of IBM in USA and/or other countries.

M. Hennecke (✉)
IBM Deutschland GmbH, Karl-Arnold-Platz 1a,
40474 Düsseldorf, Germany
e-mail: hennecke@de.ibm.com

W. Frings · W. Homberg · A. Zitz · M. Knobloch
Forschungszentrum Jülich GmbH, Wilhelm-Johnen-Strasse,
52425 Jülich, Germany

H. Böttiger
IBM Deutschland Research & Development GmbH,
Schönaicher Str. 220, 71032 Böblingen, Germany

Published online: 31 August 2011

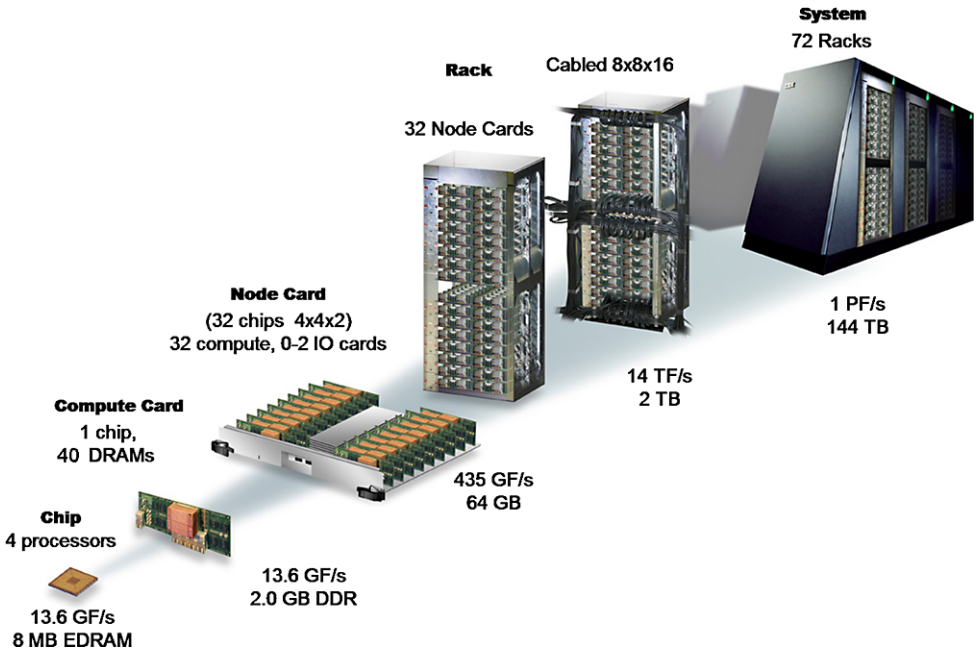

Fig. 1 Blue Gene/P system buildup

tained (augmented by OpenMP parallelism as the number of cores per node increases). This distinguishes Blue Gene from other current supercomputers, which often achieve high Flops/Watt efficiency by relying on accelerator technologies like the IBM PowerXCell* 8i [15] or GPGPUs. The *simplicity* principle includes packaging a large number of less powerful and less complex chips into a rack, and integrating most system functions including the interconnect into the compute chips [10].

A direct consequence of the Blue Gene system design is that additional energy *optimization* techniques like dynamic voltage and frequency scaling, which are typical for more complex processors operating at much higher frequencies [11–13], are both less feasible (because the Blue Gene chips do not include comparable infrastructure) and less important (as Blue Gene already operates at highly optimized voltage and frequency ranges).

On the other hand, a scalable environmental monitoring infrastructure is an integral part of the Blue Gene software environment [16]. While this is primarily used to satisfy the reliability, availability and serviceability (RAS) requirements of operating large Blue Gene systems with their huge number of components, it can also be used to *analyze* the machine's power consumption at scale while running production workloads.

The rest of this paper is organized as follows: In Sect. 2 we present the Blue Gene system architecture and power distribution network, followed by the environmental monitoring infrastructure in Sect. 3 and a detailed breakdown of Blue Gene/P power consumption in Sect. 4. Integration of Blue Gene energy information into system-level tools is described in Sect. 5. In Sect. 6, we present some initial results of analyzing job history data on the Petascale Blue Gene/P system operated by FZJ, before concluding the paper in Sect. 7.

2 Blue Gene/P architecture and power flow

The Blue Gene/P system architecture and packaging are described in detail in [9]. A Blue Gene/P *node* consists of the quad-core Blue Gene/P Compute Chip (BPC) and forty DDR3 DRAM chips, all soldered onto a printed circuit board for reliability. The BPC ASIC also includes a large L3 cache built from embedded DRAM [14], a 3D torus interconnect for MPI point-to-point operations, a tree network for MPI collectives, and a barrier network. A *node card* (NC) contains 32 compute nodes and up to two I/O nodes (IONs). Within a *midplane*, 16 node cards with 512 compute nodes are connected to form an $8 \times 8 \times 8$ torus (without any additional active components). Each midplane also contains a *service card* (SC) for bringup and management. To interconnect multiple midplanes, Blue Gene/P link chips (BPLs) are used which are packaged onto four *link cards* (LC) per midplane. The BPL ASICs can be programmed to either connect the surfaces of the $8 \times 8 \times 8$ cube to copper torus cables attached to other midplanes, or to close that torus dimension within the midplane. Figure 1 shows this system buildup.

On the top of each Blue Gene/P rack, bulk power modules (BPMs) convert AC power to 48 V DC power, which is then distributed to the cards through the two midplanes. Service cards, node cards and link cards include a number of DC/DC voltage regulator modules (VRMs) to provide

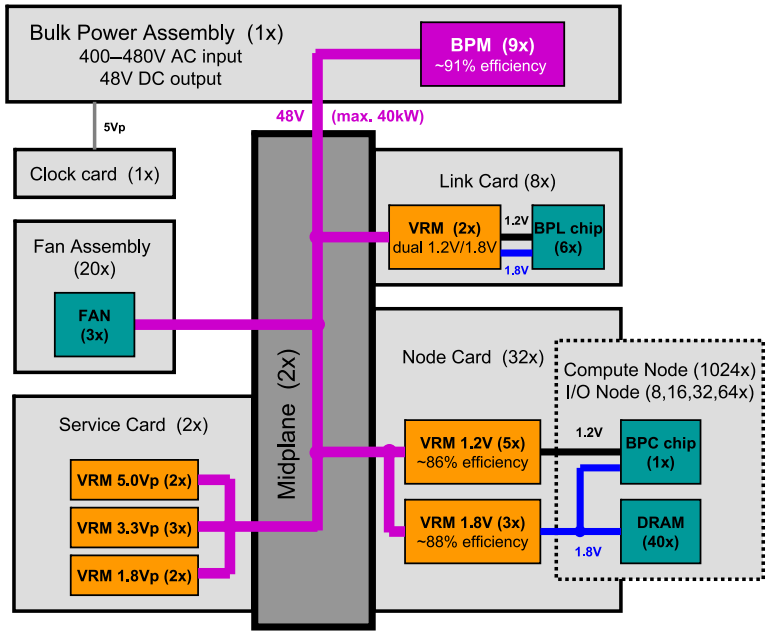

Fig. 2 Power flow within a Blue Gene/P rack

the different voltages required on the cards. All BPMs and VRMs are $N + 1$ redundant.

Heat is removed from the rack by side-to-side air cooling, with fan assemblies on the left side of the rack. The fan assemblies are powered directly from the BPMs at 48 V. Blue Gene/P also has an option for hydro-air cooling: Heat exchangers between the racks in a row are used to cool down the hot air exhausted from one rack before it enters the next rack in the row. This reduces sub-floor airflow requirements by up to 8×, and is also more efficient than using external computer room air conditioning (CRAC) units.

Figure 2 shows the power flow within a Blue Gene/P rack, including the main 48 V power distribution, the DC/DC voltage regulator modules, and the main energy consumers. Table 1 shows the part counts for the AC/DC and DC/DC voltage regulators, and Table 2 summarizes the part counts for the Blue Gene/P energy consumers.

Service cards, node cards and link cards can be accessed from an external Blue Gene service node through a 1 GbE hardware control network. Through this path, the DC/DC voltage regulators of the respective cards can be monitored as described in the next section. BPMs are monitored by the service card of the bottom midplane, and fans in a midplane are monitored by the service card of that midplane.

Table 1 Blue Gene/P voltage regulators

Blue Gene/P component	Count per card	Count per rack	Count for 1 PFlops (72 racks)
BPMs	–	9	648
SC VRMs	7	14	1,008
LC VRMs	2	16	1,152
NC VRMs	8	256	18,432

Table 2 Blue Gene/P consumers (using 8 IONs/rack)

Blue Gene/P component	Count per card	Count per rack	Count for 1 PFlops (72 racks)
Fans	3	60	4,320
BPL chips	6	48	3,456
BPC chips	1	1,032	74,304
DRAM chips	40	41,280	2,972,160

3 Blue Gene/P environmental monitoring

The Blue Gene service node uses an IBM DB2* relational database to store information about the Blue Gene machine [16]. This includes

– a *configuration* database containing a complete inventory of all hardware components;
– an *operational* database which records information about blocks (partitions), jobs, and their history;
– an *environmental* database which keeps current and past values for environmentals like temperature, voltages and currents;
– and a *RAS* database which collects hard errors, soft errors, machine checks, and software problems.

As described in Chap. 5.7 of [16], the Blue Gene Midplane Management and Control System (MMCS) is regularly polling the machine hardware to collect environmental data. All bulk power modules and voltage regulators are

```
-----------------------------------------------
export RACK="R00"

# 9 BPM LOCATIONs are $RACK-B-P0 to $RACK-B-P8

db2 "select substr(LOCATION,1,3) as RACK, TIME,
  (bigint(sum(OUTPUT0CURRENT*OUTPUT0VOLTAGE)))
  as WATT from BGPBULKPOWERENVIRONMENT
  where substr(LOCATION,1,3)='$RACK'
  group by substr(LOCATION,1,3), TIME"

RACK TIME                       WATT
---- -------------------------- --------
R00  2011-04-04-10.25.30.734577    26341
R00  2011-04-04-10.27.32.832208    24800
R00  2011-04-04-10.29.37.272993    23995
R00  2011-04-04-10.31.44.744716    22572
...
-----------------------------------------------
```

Fig. 3 SQL query to report rack power consumption

monitored, and their power environmental data (including output voltage and current) is stored in four DB2 tables:

- BGP*BULK*POWERENVIRONMENT
- BGP*SRVCCARD*POWERENVIRONMENT
- BGP*NODECARD*POWERENVIRONMENT
- BGP*LINKCARD*POWERENVIRONMENT

The polling intervals for the three card types (SC, LC and NC) can be individually controlled, BPMs are monitored by the service cards. The default polling interval is 300 seconds, and the valid range is 60–1800 seconds. With the environmental *monitoring* performed automatically by the MMCS subsystem, *reporting* power consumption is a simple SQL SELECT statement. For example, Fig. 3 shows how to report the per-rack power consumption (48 V DC output of the nine BPMs in a rack). Similar queries can be used to report the DC output power of the voltage regulators on the service cards, link cards, and node cards.

These SQL queries are an effective tool to obtain a high-level view of the machine's power consumption. Using information in the operational database, it is also possible to aggregate this data for individual hardware locations into per-partition or per-job data [17]. Section 5 shows the integration of this data into FZJ's `LLview` tool.

The main limitation of using standard MMCS environmental monitoring to measure power consumption is its relatively slow sampling frequency. On Blue Gene/P, environmental monitoring sequentially queries all cards of a given type, and this time has to be added to the configured polling interval for that card type (which is at least 60 seconds). While the time required for the actual queries may be neglectable on small machines, at Petascale it can become significant (with component counts as summarized in Table 2):

- For 72 racks, querying all service cards (and the BPMs monitored through them) takes roughly one minute. So the fastest achievable BPM sampling on a 1 PFlops system is one reading every 2 minutes.
- Querying all the link cards in a 72-rack system also takes roughly one minute. Link cards will typically be monitored at lower sampling frequency, because power consumption of the Blue Gene/P link chips does not vary much over time.
- Node cards draw the majority of a Blue Gene/P rack's power, and would be the most interesting component to monitor at high sampling frequency. However, they also contain the largest numbers of VRMs and consumers, and querying all 18,432 node card voltage regulators in FZJ's 72-rack system takes about 12 minutes.

In order to analyze application power consumption on Blue Gene/P at a higher sampling frequency than what is possible using standard MMCS environmental monitoring, we have developed a tool which directly interfaces to the lower level Blue Gene/P control system. It allows to read the environmentals of an individual node card at roughly four measurements per second. This tool has been used to generate the detailed timeline plots shown in Sects. 4 and 6.

4 Blue Gene/P power consumption details

The power consumption of a Blue Gene/P rack will vary depending on the current state of the hardware:

- The *peak* DC output power of the nine BPMs is 40 kW. With a BPM efficiency of 91%, this is 44 kW peak AC input power per rack. For FZJ's 1 PFlops machine this totals 2.9 MW peak DC output (3.2 MW peak AC input).
- Blue Gene/P power consumption when running the *LINPACK* benchmark is 31.5 kW AC (28.7 kW DC) per rack, or 2.3 MW AC (2.1 MW DC) for 72 racks.
- In Sect. 6, we present an analysis of the FZJ production workload, indicating an *average* of about 23 kW DC per rack (1.65 MW DC in total).

In addition to these operating states, two other states of the hardware are important: the *standby power* of a Blue Gene/P rack when both midplanes are shut down, and the *idle power* when the two midplanes are booted but no job is running. Monitoring the DC output of the rack BPMs indicates that the rack standby power is about 8.5–9 kW DC and the rack idle power (midplanes are booted) is in the 20–21 kW DC range.

A breakdown of the Blue Gene/P total rack power by system component can be derived as follows, referring to Fig. 2 which highlights the main consumers.

- Power consumption of the *fans* cannot be actively monitored. An upper bound is the "label plate" peak power of

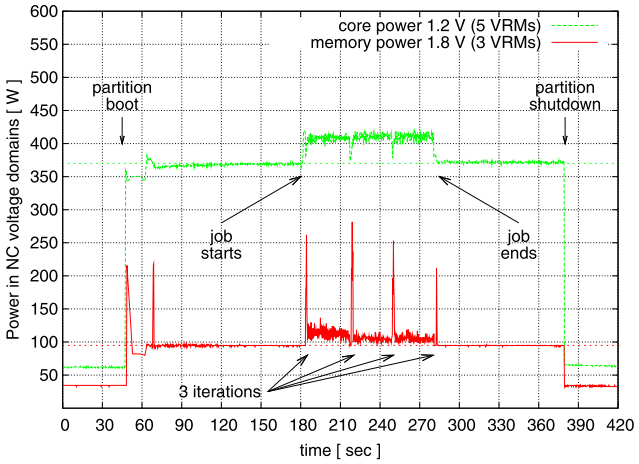

Fig. 4 Power consumption for a single-nodecard mpirun

1.9 kW (60 fans with 48 V 650 mA), with an expected operational power of about 1.6 kW.

- Monitoring the *link cards* provides an estimate of about 66 W of VRM output per link card, totaling about 520 W of BPL ASIC power per rack. This number varies only slightly (in the 10% range) across applications.
- Monitoring the *service cards* indicates about 250 W of SC VRM output per midplane (500 W per rack), with a 10–20% variation.

Factoring in a DC/DC VRM efficiency of about 87%, the above components total about 3 kW. So as expected, the majority of the BPM output power is consumed by the *node cards*.

Figure 4 shows the power consumption time line of a single node card (with 32 nodes) for a complete `mpirun` cycle, separated into the 1.2 V and 1.8 V VRM output domains. Those two voltage domains indicate *core power* (1.2 V) and *memory power* (1.8 V). The L3 cache on the BPC ASICs, which is implemented as eDRAM [14], is also powered at 1.8 V and included in the memory power.

Initially the node card is powered on but the compute nodes are not booted, resulting in a standby power consumption of roughly 65 W (1.2 V) + 35 W (1.8 V) = 100 W. When `mpirun` allocates the partition, the 32 compute nodes are booted which increases the node card power to about 365 W (1.2 V) + 100 W (1.8 V) = 465 W. Three iterations of a small test program are run (with only slightly higher power consumption than idle power), and finally the partition is shut down and nodes are powered off again.

When including the DC/DC VRM efficiency of 86% (1.2 V) and 88% (1.8 V), the node card idle power is 538 W which adds up to about 17.2 kW per rack. Together with the 3 kW for the remaining infracstructure, this matches the 20–21 kW of idle power as measured at the BPM level.

5 Integration of power data into LLview

The `LLview` tool [18, 19] developed by FZJ provides a graphical representation of the machine, together with information on the currently running and waiting batch jobs. It has now been expanded to include Blue Gene/P power environment data, in particular the per-rack BPM data. Figure 5 shows the `LLview` GUI for the 1 PFlops machine at FZJ, shortly after a full-system job has ended and smaller jobs are just being started. Above each rack, `LLview` displays the BPM output power of that rack as measured in Fig. 3. This is the BPM 48 V DC output power, not including the AC/DC conversion loss.

Within `LLview`, it is also possible to visually associate a running job with the subset of the machine hardware that it is running on. For example, hovering over a hardware component in the hardware view highlights the associated job in the job window, and hovering over a job in the job list highlights the hardware components (midplanes) allocated to that job. From the different rack states and jobs in the screen shot of Fig. 5, a number of observations can be made:

- Job 1 runs only on full racks in row 0 and row 1. The same is true for job 2 (violet) running on row 3.
- Job 3 runs on row 6, but the power display still shows $<$10 kW. This is an artifact: The `LLview` refresh rate for this plot was 60 sec, but as shown in Fig. 3 the BPM power data is only returned every 2 minutes.
- Job 4 uses only one of the two midplanes in a rack, on 16 racks in rows 2, 5 (top), 4 and 7 (bottom). In some of those racks the second midplane is idle, while in others the second midplane is allocated to a different job (job 5).
- Job 5 also runs in a partition which is using only one midplane per rack, on 12 racks in rows 2, 5 (top) and 4 (bottom).

When aggregating the power consumption of individual midplanes into a per-job power consumption, one complication arises: The nine BPMs in the top of the rack provide the power for *both* midplanes in the rack, but as shown in Fig. 5 partitions may be created which only use one of the two midplanes in a rack. Our tools distinguish three modes of splitting the per-rack power consumption of such racks into two per-midplane values:

- If the other midplane in the rack is not booted, 4.5 kW is subtracted from the total rack power (as a Blue Gene/P rack in which both midplanes are shut down has a DC power consumption of about 9 kW, see Sect. 4).
- If the other midplane in the rack is booted but no job is running on it, 10 kW is subtracted from the total rack power (as a Blue Gene/P rack in which both midplanes are booted but idle has a DC power consumption of about 20 kW).

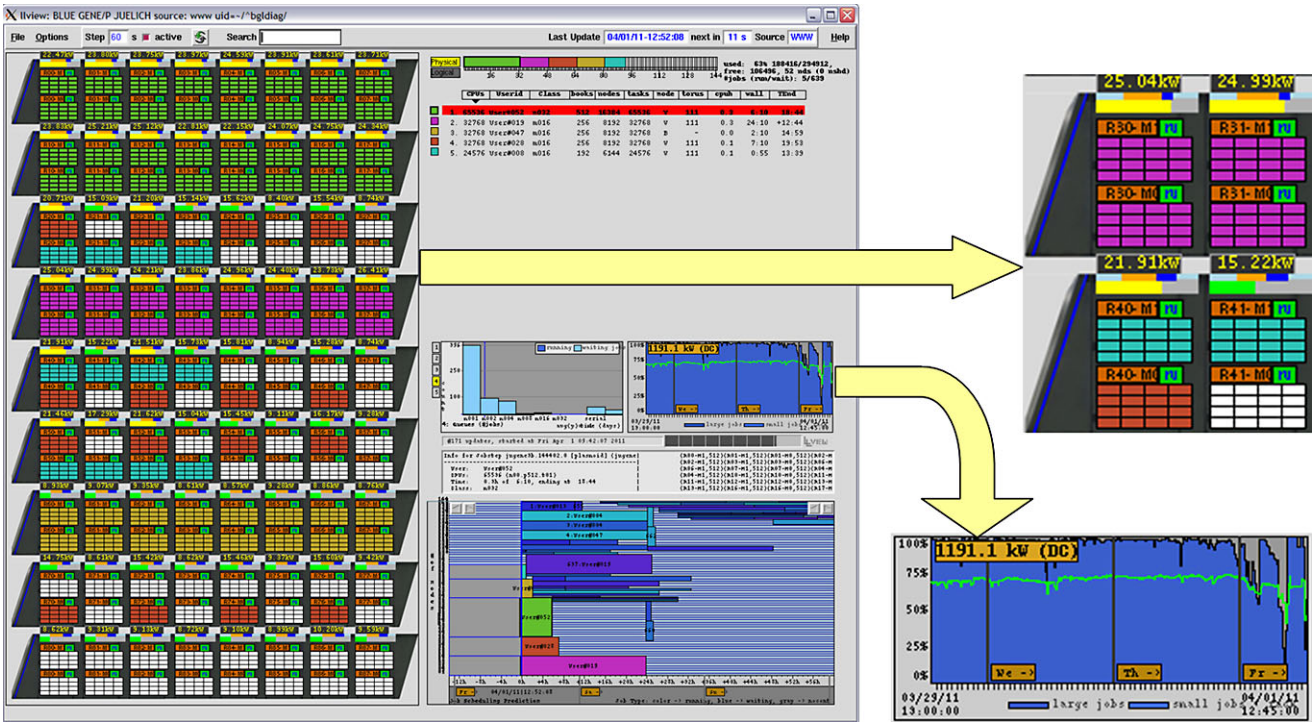

Fig. 5 Power consumption display in the LLview GUI of a 72-rack Blue Gene/P machine. The hardware view on the *left* shows the 9 rows of 8 racks, with row 0 at the *top* and row 8 at the *bottom*

– If the other midplane in the rack is booted and allocated to a different job, the total rack power is split equally between the two jobs. This is a simplification which may be improved in the future, e.g. by comparing the rack power of multiple racks.

Using this simple heuristic, `LLview` can now also report Blue Gene/P *per-job* power consumption.

The full system utilization display in the center right of the `LLview` GUI has been augmented to display total system power consumption, in addition to total system utilization. A history curve is plotted, and the current state is displayed in a box (here, 1191.1 kW). Note that like for the individual racks, this is the sum of the 48 V DC output of the BPMs so it does not include the AC/DC conversion loss within the BPMs. At a BPM efficiency of roughly 91%, AC input in this case would be roughly 1.3 MW.

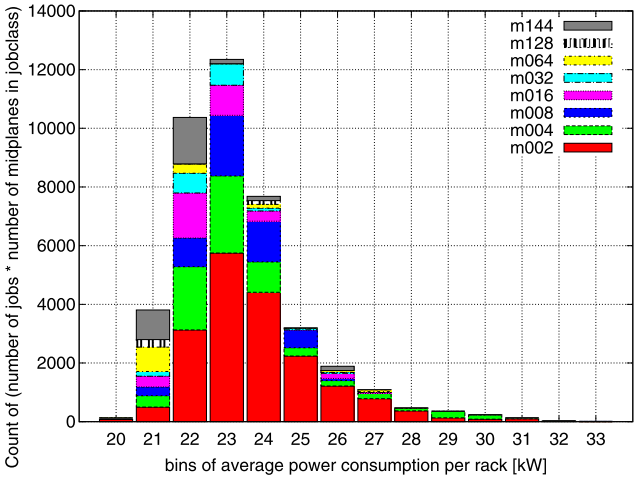

Fig. 6 Distribution of average per-rack power consumption

6 Analyzing the FZJ production workload

To understand the energy characteristics of FZJ's current production workload, a two-phase approach has been taken.

Firstly, BPM power monitoring is used 24 × 7 to record average per-rack power consumption for all jobs. Figure 6 shows the distribution of average kW/rack power consumption for over 12,500 jobs run in the first quarter of 2011. The bar chart displays job counts separately by partition size, from single rack jobs ("m002") to full system runs ("m144"). Jobs smaller than a full rack, or shorter than 30 minutes, are excluded.

Most jobs fall into the 22 to 24 kW per rack range, and very few jobs are at or above 28 kW (DC) per rack and thus comparable to or exceeding LINPACK power (31.5 kW AC × 91% = 28.7 kW DC). The bar graph also shows that very large jobs tend to be in the low kW/rack space. For example, almost all "m144" jobs are in the 21 and 22 kW bins. This is plausible, as larger jobs are generally more prone to

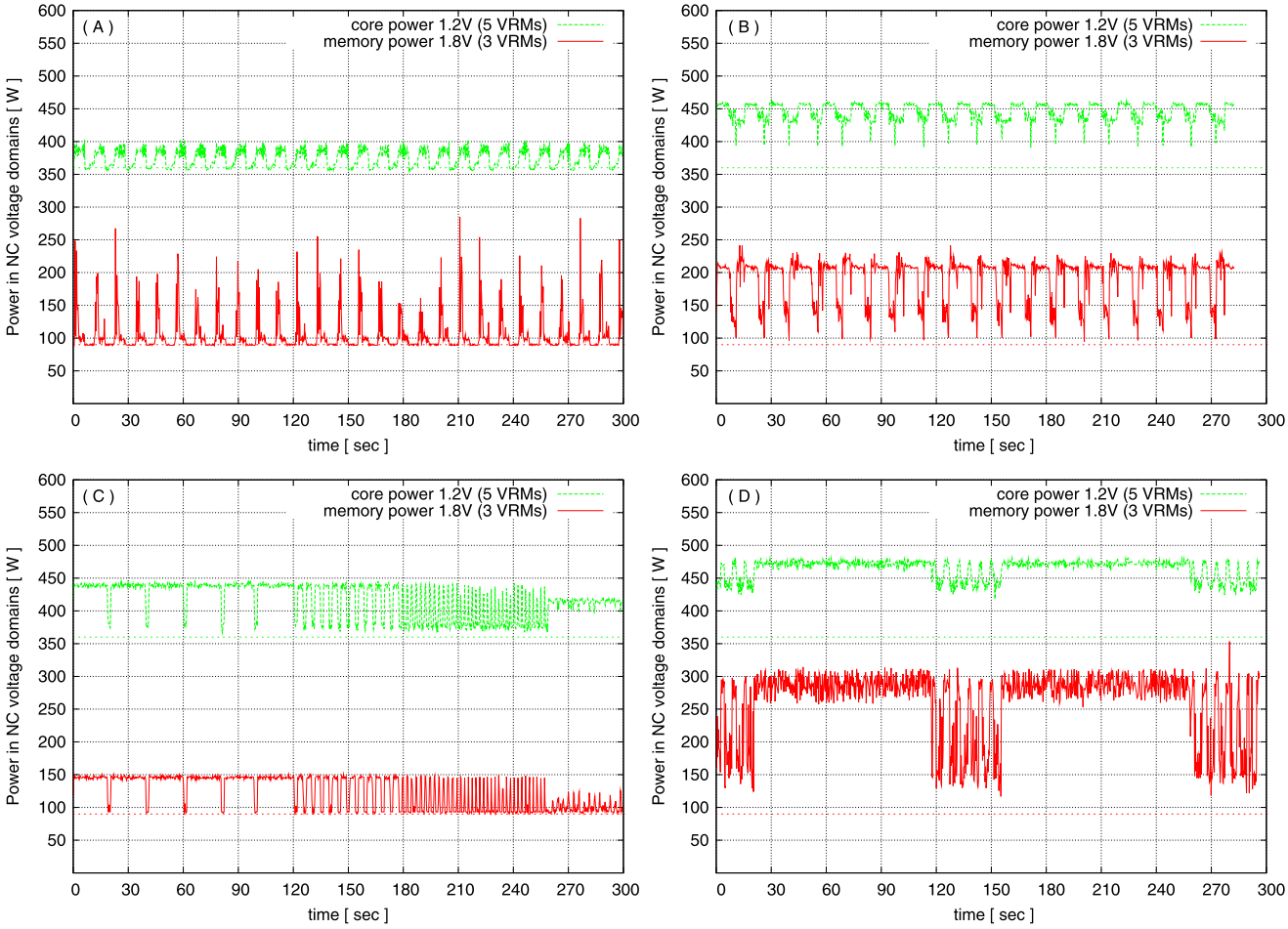

Fig. 7 Single node card power consumption timeline of four applications (300 seconds, VRM DC output, 4 Hz sampling)

load imbalance and communication wait times which tends to decrease power consumption.

Secondly, for jobs with above-average power consumption we have collected high resolution single nodecard traces to better understand their energy profiles. Note that nodecard power data is the aggregate power of the 32 nodes on the node card. Figure 7 shows four samples using typical applications in the FZJ workload. While core power is always larger than memory power, it is evident that variations of memory power are very pronounced: Core power seldomly varies by more than 50 W per node card (32 nodes), whereas for memory power variations of 100 W or even 200 W are not uncommon. Similarly to Fig. 4, the spikes in the timelines typically correspond to iterations in the applications. Several types of application characteristics can be observed:

- Figure 7 (A) is from an application with a lot of indirect addressing, so memory power is low during the computational phases and increases during messaging (which in this case transfers more contiguous memory regions).
- Figure 7 (C) shows an iterative refinement on the timeline.
- Figure 7 (D) has a very high power consumption (cores and memory) during the computational phases, and power consumption drops during communication. This is one of the applications with a per-rack power consumption exceeding 30 kW DC.

Work is in progress to integrate the high resolution power consumption timelines into application performance tools, to be able to correlate them with MPI traces, hardware performance counters, and other performance metrics. For example, writing this data in OTF format [21] would allow it to be visualized alongside with MPI traces [20].

7 Summary and outlook

We have described the Blue Gene/P power flow, have shown the variations of power consumption per rack depending on the state of the partitons, (shut down, booted but idling, booted and running user jobs), and have analyzed the FZJ production workload to obtain a better understanding of "typical" power consumption at Petascale.

At this technology level, more than 75% of the total rack power is consumed by the nodes (BPC ASICs and memory), roughly 20% is spent in AC/DC and DC/DC power conversion, and all other system functions including the interconnect consume less than 5%.

Traces taken from jobs running at high kW/rack ratings indicate that memory power currently is the most relevant contributor to power variations, and work is ongoing to integrate Blue Gene/P power consumption traces into application performance tools to better understand the power characteristics of those applications. This will enable a better understanding of the options to save power on current generation machines, and will also be useful when moving towards future system architectures where it is expected that more and more of the total rack power will be spent for accessing memory and the interconnect [3].

Acknowledgements The authors would like to thank the IBM Blue Gene Research and Development teams, in particular P. Coteus, R. Rand, M. Megerian and M. Woiwood, for valuable input to this study.

This work was supported by the FZJ/IBM Exascale Innovation Center, EIC cooperation agreement T/Z1213.02.09.

References

1. The Top500 list (November 2010). http://www.top500.org/lists/2010/11
2. The Green500 list (June 2011). http://www.green500.org/lists/2011/06/top/list.php
3. Kogge P (2009) Energy at exaflops. SC09 exascale panel. http://www.exascale.org/mediawiki/images/6/6e/Sc09-exa-panel-kogge.pdf
4. Bekas C, Curioni A (2010) A new energy aware performance metric. Comput Sci Res Dev 25:187–195. doi:10.1007/s00450-010-0119-z
5. Moreira J et al (2007) The Blue Gene/L supercomputer: a hardware and software story. Int J Parallel Program 35(3):181–206. doi:10.1007/s10766-007-0037-2
6. Gara A et al (2005) Overview of the Blue Gene/L system architecture. IBM J Res Dev 49(2/3):195–212. doi:10.1147/rd.492.0195
7. Coteus P et al (2005) Packaging the Blue Gene/L supercomputer. IBM J Res Dev 49(2/3):213–248. doi:10.1147/rd.492.0213
8. Moreira J et al (2005) Blue Gene/L programming and operating environment. IBM J Res Dev 49(2/3):367–376. doi:10.1147/rd.492.0367
9. IBM Blue Gene team (2008) Overview of the IBM Blue Gene/P project. IBM J Res Dev 52(1/2):199–220. doi:10.1147/rd.521.0199
10. Bright A, Ellavsky M, Gara A, Haring R, Kopcsay G, Lembach R, Marcella J, Ohmacht M, Salapura V (2005) Creating the BlueGene/L supercomputer from low-power SoC ASICs. In: Proceedings of IEEE international solid-state circuits conference. doi:10.1109/ISSCC.2005.1493932
11. Ware M, Rajamani K, Floyd M, Brock B, Rubio J, Rawson F, Carter J (2010) Architecting for power management: the IBM POWER7 approach. In: 2010 IEEE 16th international symposium on high performance computer architecture (HPCA). doi:10.1109/HPCA.2010.5416627
12. Floyd M, Ware M, Rajamani K, Gloekler T, Brock B, Bose P, Buyuktosunoglu A, Rubio J, Schubert B, Spruth B, Tierno J A, Pesantez L (2011) Adaptive energy-management features of the IBM POWER7 chip. IBM Journal of Research and Development 55(3). doi:10.1147/JRD.2011.2114250
13. Brochard L, Panda R, Vemuganti S (2010) Optimizing performance and energy of HPC applications on POWER7. Comput Sci Res Dev 25:135–140. doi:10.1007/s00450-010-0123-3
14. Iyer S, Barth J, Parries P, Norum J, Rice J, Logan L, Hoyniak D (2005) Embedded DRAM: technology platform for the Blue Gene/L chip. IBM J Res Dev 49(2/3):333–350. doi:10.1147/rd.492.0333
15. Baier H et al (2010) QPACE: power-efficient parallel architecture based on IBM PowerXCell 8i. Comput Sci Res Dev 25:49–154. doi:10.1007/s00450-010-0122-4
16. Lakner G (2010) IBM system Blue Gene solution: Blue Gene/P system administration. IBM redbook SG24-7417-03. http://www.redbooks.ibm.com/abstracts/sg247417.html
17. Hennecke M (2010) Saving block history data in the Blue Gene/P database. IBM techdocs whitepaper WP101678. http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101678
18. LLview: graphical monitoring of LoadLeveler controlled cluster. http://www.fz-juelich.de/jsc/llview/
19. Frings W (2007) New features of the batch system monitoring tool LLview. ScicomP13, Garching, 20. July 2007. http://www.spscicomp.org/ScicomP13/Presentations/User/WolfgangFrings-MON.pdf
20. Nagel W, Weber M, Hoppe H-C, Solchenbach K (1996) VAMPIR: visualization and analysis of MPI resources. Supercomputer 12(1):69–80. http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.1615
21. Knüpfer A, Brendel R, Brunst H, Mix H, Nagel W (2006) Introducing the open trace format (OTF). In: Computational science (ICCS 2006). Lecture notes in computer science, vol 3992/2006, pp 526–533. http://dx.doi.org/10.1007/11758525_71