Comput Sci Res Dev (2010) 25: 135–140
DOI 10.1007/s00450-010-0123-3

SPECIAL ISSUE PAPER

Optimizing performance and energy of HPC applications on POWER7

Luigi Brochard · Raj Panda · Sid Vemuganti

Published online: 12 August 2010
© Springer-Verlag 2010

Abstract Power consumption is a critical consideration in high performance computing systems and it is becoming the limiting factor to build and operate Petascale and Exascale systems. When studying the power consumption of existing systems running HPC workloads, we find power, energy and performance are closely related leading to the possibility to optimize energy without sacrificing (much or at all) performance.

This paper presents the power features of the POWER7 and shows how innovative software can use these features to optimize the power and energy consumptions of large cluster running HPC workloads.

This paper starts by presenting the new features which have been introduced in POWER7 to manage power consumption and the tools available to manage and record the power consumption. We then analyze the power consumption and performance of different HPC workloads at various levels of the POWER7 server (processor, memory, io) for different frequencies. We propose a model to predict both the power and energy consumption of real workloads based on their performance characteristics measured by hardware performance counters (HPM). We show that the power estimation model can achieve less than 5% error versus actual measurements. In conclusion, we present how an innovative scheduler can help to optimize both power and energy consumptions of large HPC clusters.

Keywords Energy · Performance · Power consumption · POWER7

L. Brochard (✉)
Systems and Technology Group, IBM, Bois-Colombes, France
e-mail: luigi.brochard@fr.ibm.com

R. Panda · S. Vemuganti
Systems and Technology Group, IBM, Austin, TX, USA

1 Introduction

Power consumption has become a serious concern to HPC data centers. In many situations power consumption is becoming the determining factor of the system size. This is due to both the rising power consumption of the systems and the rising costs of energy. Managing power and energy is therefore a major issue for HPC. Realizing the importance of power management, hardware vendors are building more and more dynamic power management capabilities into microprocessors and server systems as well as providing software tools to obtain and view the power consumption data from the server systems. Some of the available tools can also be used to set limits on the power delivered to the server systems and thus help data center managers in the management of power and cooling costs. However, these software tools are not targeted to parallel applications and do not predict the impact of processor frequency scaling on total energy consumption, where energy is defined as the product of power consumption x elapsed time. For example, cases are described in this paper in which decreasing the processor frequency increases the energy consumed by the application while power consumption is decreasing.

Recognizing the HPC community's need, we present first the innovations which have been introduced in the POWER7 microprocessor and servers to better manage power consumption. We then study performance and power consumption on a selection of HPC applications. Our objective is to experimentally obtain generalized power-performance correlations for HPC applications that can be used to estimate the power consumption and energy of an application, on any platform, and at any frequency. Results are presented for IBM POWER7.

M. Broyles et al. [3] describe the system power management support in the POWER7 processors. Techniques such

as frequency scaling and power and temperature monitoring capabilities are discussed. Allarey et al. [4] describe idle and multi-core dynamic power reduction features in Intel's 65 nm cores, and they introduce a deep power-down idle state and power-performance tradeoffs for single threads, as well as enhanced sleep states. Rajamani et al. [5] propose real-time power and performance prediction capabilities that can be used for dynamic control of system resources such as DVFS and clock throttling to improve power-performance. They extend prior work related to average power prediction to predicting instantaneous processor power to enable applications such as operating system scheduling. Lee et al. [6] dynamically predict performance and power using regression models. Our work is distinguished from prior work by a generalized power consumption model which predicts the power consumption at any frequency based on the application characteristics at nominal frequency and by introducing energy as an additional metric to optimize. The rest of the paper is organized as follows.

Section 1 gives a brief description of the POWER7 and its power management features and the tools we used. Section 2 gives a description of the applications we studied, the performance data gathering process and the derived metrics used in the models. Section 3 presents the performance and power measurements gathered on the POWER7 server, and the impact of frequency scaling on power and energy for each application. Section 4 presents the model used for power projections.

2 POWER7 power management features and tools

The system used in our experiments is the IBM Power 750 server.

The IBM Power 750 is a 4 socket 8 cores per socket server based on the new POWER7 microprocessor. Each core is running at a nominal frequency of 3.55 GHz and is capable of 8 Flops per cycle when using the new VSX instruction unit [1].

POWER7 chip has the following characteristics:

- IBM's 45 nm SOI process
- 567 mm^2, 1.2 B transistors
- 8 out-of-order cores, 4-way SMT
- 32 KB L1 D/I, 256 KB L2 per core, 32 MB shared L3 in IBM's eDRAM process

The Power 750 server we used had 32×4 GB DDR3 DIMMs running at 1066 MHz. Regarding power management, all POWER7 severs have an integrated Thermal Processor Monitoring Device (TPMD) which manages the power consumption of the server.

There are also 44 digital sensors on chip (5 per core, 4 extra-chiplet): on-board ambient temperature sensor, memory buffer/DIMM thermal sensors and VRM thermal-trip logic, on-board measurement circuits and A/D channels to measure full system, processor socket, memory sub-system, I/O sub-system and fan power. In addition, POWER7 has also performance/activity sensors to monitor core-level usage with active cycle counts, instruction throughput counts core-level memory hierarchy usage and memory controller-level activity. TPMD gathers and analyzes all this information.

To better manage power consumption, POWER7 is capable of running at 8 P-states and different frequencies varying from 110% to 50% of nominal frequency.

POWER7 has also two to save energy when processor is idle:

Nap mode clocks off all execution units and L1 caches within core when unused. Sleep mode clocks off for the entire chiplet and caches are flushed prior to entry in sleep mode.

IBM has also introduced an EnergyScale architecture to manage power and energy on POWER6 with new features on POWER7 [3]. The main features are:

- Power/Thermal Trending
  - Collect and report power consumption, inlet and exhaust temp
- Power Capping
  - Guaranteed (Hard Cap)
    - Enforces a power cap via Dynamic Frequency and Voltage Slewing
  - Soft Power Cap
    - Attempted lower cap, but not guaranteed.
- Energy Management Modes—Enhanced for POWER7
  - Static Power Save (SPS)
    - Save power via a fixed voltage and frequency drop—as much as 30% down
    - Put processor in nap and sleep mode when cpu is idle
  - Dynamic Power Save (DPS)
    - Optimize power vs performance using Dynamic Voltage and Frequency Skewing
    - Will save power at most utilizations
  - Dynamic Power Save—Favor Performance (DPS-FP)
    - Will provide performance boost at most utilizations
    - Will save power only at very low utilization

Two system management tools were used to collect the power data used in our experiments. Amester is a tool that is internal to IBM that we have used to measure power at a component level in the blade server while AEM (Active Energy Manager) is a commercially available product that can be used to measure power at the server level as well as power at chassis level.

The Autonomic Management of Energy (AME) project was started at the IBM Research Lab in Austin in 2004 with the goal of controlling server or blade power-performance to within a specified power and temperature budget [2].

All POWER7 systems are provided with on-board power-measurement circuits and firmware additions to monitor the circuit outputs. An AME circuit places a very low impedance resistor in series with a power rail. Circuits are placed on the various rails feeding the VRMs (voltage regular modules) that power the system. The power management device (TPMD) then converts the voltage drop on each resistor to digital form.

Amester is executed from a remote machine and passes control commands to the service processor which executes the commands by interfacing to the TPMD and returns the requested data. Amester can sample the power dissipation at intervals from 1 ms on up. In the following experiments a sampling interval of 3 ms to collect power measurements of blade total power, core power, and DIMM power.

Active Energy Manager (AEM) provides management and control of the chassis and individual blade energy use [1]. It supports analysis and control such as power trending and capping, thermal trending, and CPU trending at the chassis or individual blade levels. It's part of IBM Tivoli Director.

3 Applications and performance metrics

For an effective analysis of power-performance, a set of floating point benchmarks was chosen that stress either the processor or the memory in the system, or both. A subset of 8 out of a possible 17 of the SPEC CPU2006 benchmarks was chosen in order to speed up the data collection and analysis tasks and to represent different benchmarks that are important to HPC as well as for their different performance characteristics in terms of CPI (cycles per instruction) and memory bandwidth. Although these are not true parallel applications, measurements show little difference in power consumption between true parallel workloads and the chosen set of benchmarks. Table 1 shows the selected benchmarks.

To carry out our experiments, performance counter data from the 750 was collected when running the workloads.

Table 1 List of benchmarks and HPC areas

Benchmark	Area
416.gamess	Quantum chemistry
433.milc	Physics
435.gromacs	Molecular dynamics
437.leslie3d	Fluid dynamics
444.namd	Molecular dynamics
454.calculix	Structural analysis
459.GemsFDTD	Electromagnetics
481.wrf	Weather forecasting

For gathering the counter data, we used *hpmcount* tool. Performance metrics like CPI (cycles per instruction) and memory bandwidth were computed for each of the SPEC benchmarks based on the hardware counter data collected using this tool. We ran the SPEC benchmarks in the throughput mode to assess the capability of each system. A number of copies of each of the benchmarks in Table 2 were on the platform for gathering the hardware counter data.

Although POWER7 allows 4 threads per physical core, we used ST (single thread) mode and ran the same number of copies as the number of cores in the system. Based on the elapsed time for the throughput benchmarks to complete on a system, one can also compute the throughput performance called "rate" according to SPEC benchmark rules. In Table 2, CPI (Cycle per Instruction) and memory bandwidth GBS (Giga Byte per second) metrics are shown for each of the benchmarks for the Power 750 system.

Significant differences between these applications are apparent. Some applications (highlighted in grey) are very core intensive and have very little memory bandwidth. The remaining applications (highlighted in white) have high memory bandwidth requirements. Wrf is both core intensive and high memory bound.

4 Power and performance measurements

Figure 1 shows how SPS mode caps frequency at 30% of nominal frequency (2.5 GHz vs 3.55 GHz) and lowers frequency at 1.65 GHz when the cpu is idle. This measurement is done with AMESTER collecting data every 3 ms.

Tables 3 and 4 below summarize the different components of power consumption per benchmark. The different components are the processor sockets, the memory DIMMS (labeled DIMM) and Static which includes the IO chip, off-chip cache, etc. Please note this power consumption does not include the AC/DC conversation and the associated power loss.

As expected, processor power consumption accounts for the majority of the total power consumption. Coming in sec-

Table 2 CPI and GBS for selected benchmarks on Power 750 at 3.55 GHz

Benchmark	CPI	GBS
416.gamess	0.59	0.06
433.milc	2.74	40.00
435.gromacs	0.75	0.98
437.leslie3d	0.91	39.42
444.namd	0.72	0.36
454.calculix	0.53	3.07
459.GemsFDTD	1.96	38.54
481.wrf	0.58	29.15

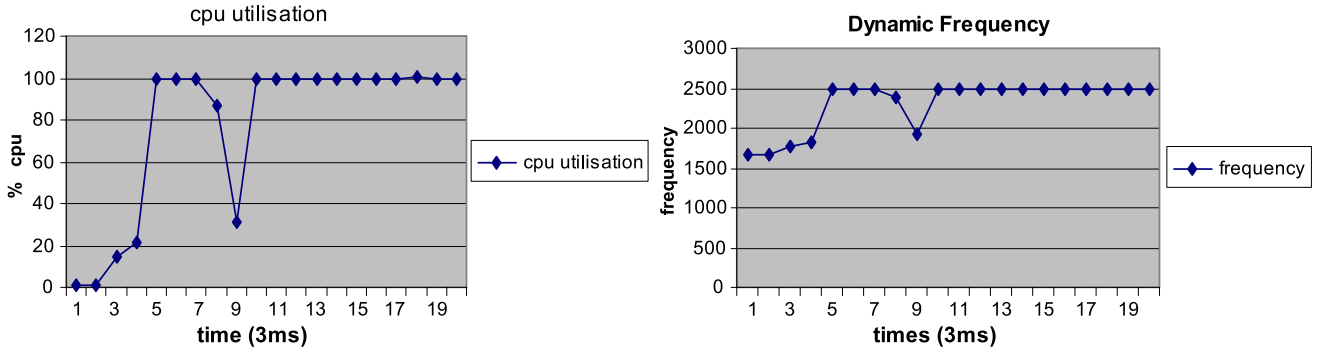

Fig. 1 Dynamic variation of core frequency on Power 750

Table 3 Performance and power consumption on Power 750 at 3.55 GHz

Benchmark	Measured average power (Watts)				CPI	GBS
	Socket	DIMM	Static	Total		
416.gamess	724	110	151	985	0.59	0.06
433.milc	646	268	171	1085	2.74	40.00
435.gromacs	705	144	153	1002	0.75	0.98
437.leslie3d	699	266	176	1142	0.91	39.42
444.namd	717	135	149	1001	0.72	0.36
454.calculix	747	142	151	1040	0.53	3.07
459.GemsFDTD	645	264	161	1070	1.96	38.54
481.wrf	750	234	163	1147	0.58	29.15
Idle power	517	98	153	769		

Table 4 Performance and power consumption on Power 750 at 2.5 GHz

Benchmark	Measured average power (Watts)				CPI	GBS
	Socket	DIMM	Static	Total		
416.gamess	423	103	138	665	0.61	0.06
433.milc	416	265	160	841	1.94	39.11
435.gromacs	417	143	144	704	0.75	0.69
437.leslie3d	453	264	157	874	0.65	38.78
444.namd	423	131	141	695	0.72	0.25
454.calculix	441	138	142	721	1.07	2.18
459.GemsFDTD	417	261	154	832	1.44	37.58
481.wrf	449	210	152	811	0.55	21.30
Idle power	304	98	139	541		

ond, off-chip cache and IO chips consume a large amount of power regardless of the application characteristics.

Regarding idle power, we should notice in this experiment the processor went into sleep mode. If the processor had been in nap mode, power consumption would have been around 600 Watt.

We now present the impact of frequency scaling on power and energy for the various benchmarks. Table 6 shows the power and performance (Elapsed time) effects of down clocking, or reducing frequency. In this table, Energy is in kWatt/Hour and defined by:

$$\text{Energy} = \text{Average Power} \times \text{Elapsed Time} \quad (1)$$

From Table 5, it can be seen that down clocking frequency though always saves power, it does not always save energy. This behaviour arises when power saving is less than the performance degradation. It happens for cpu bound application (highlighted in grey) as the performance of these benchmarks is directly affected by CPU frequency. On the other hand, down clocking frequency dos save energy for applications which are memory bound (highlighted in white).

5 Power consumption model

Based on these experiments,, we derive a model to predict the power consumption of a given benchmark at frequency f_n given its characteristics measured at frequency f_0:

$$\text{PWR}(f_n) = A_n * \text{GIPS}(f_0) + B_n * \text{GBS}(f_0) + C_n \quad (2)$$

where PWR, GIPS and GBS are respectively power consumption, Giga instructions per second, Giga bytes per second at a given frequency, with GIPS = Processor Frequency/(CPI × 10**9). GIPS(f_0) and GBS(f_0) are application characteristics measured at the nominal frequency (f_0). A_n, B_n and C_n are coefficients for a given platform at frequency f_n. They are calculated for each frequency with a multiple linear regression analysis using the method of least squares for determining the total power consumption over all workloads at a given frequency to fit equation (2). This model over the total power consumption provides a better fit than using separate models for the specific core power, memory power, and other power, and then adding them up. The physical meaning is less evident in the combined model,

Table 5 Down clocking from 3.55 to 2.5 GHz

p755, down clocking from 3.55 to 2.5 GHz				Delta	Saving	Saving
Workload	3.55 GHz runtime (s)	3.55 GHz power (W)	3.55 GHz energy (KWh)	perf	power	energy
416.gamess	956	984	0.3	−41.0%	32.5%	4.8%
433.milc	563	1084	0.2	−2.1%	22.4%	20.8%
435.gromacs	517	1002	0.1	−42.7%	29.7%	−0.3%
437.leslie3d	480	1141	0.2	−1.5%	23.3%	22.2%
444.namd	365	1001	0.1	−42.7%	30.5%	0.9%
454.calculix	442	1041	0.1	−41.4%	30.8%	2.1%
459.GemsFDTD	776	1070	0.2	−3.0%	22.2%	19.9%
481.wrf	474	1147	4.7	−35.2%	29.3%	4.4%

Table 6 Power coefficients

Frequency (GHz)	A_n	B_n	C_n	Aver. error
2.5	8.4	4.3	666.3	1.87%
3.55	22.3	4.3	877.1	1.05%

Table 7 Power consumption projection at 3.55 GHz

Benchmark	Measured power (W)	Projected power (W)	Error
416.gamess	984	1012	2.85%
433.milc	1084	1080	0.41%
435.gromacs	1002	987	1.52%
437.leslie3d	1141	1135	0.47%
444.namd	1001	989	1.19%
454.calculix	1041	1041	0.01%
459.GemsFDTD	1070	1085	1.43%
481.wrf	1147	1141	0.54%
		Avg.	1.05%

Table 8 Power consumption projection at 2.5 GHz

Benchmark	Measured power (W)	Projected power (W)	Error
416.gamess	665	701	5.41%
433.milc	841	847	0.67%
435.gromacs	704	697	1.01%
437.leslie3d	874	867	0.88%
444.namd	695	697	0.19%
454.calculix	721	695	3.53%
459.GemsFDTD	832	844	1.43%
481.wrf	811	796	1.80%
		Avg.	1.87%

but it is designed specifically to serve the purpose of projecting power at some frequency f_n based on the nominal frequency f_0, thereby hiding the dependency of GIPS and GBS on clock frequency for a given benchmark. In Table 6, we present the resulting values of the A, B and C coefficients for the power equation. The average error using this model on all benchmarks is less than 1.9%. In Tables 7 and 8, we present the projected power consumption for each application and the error between the projected and the measured power.

6 Conclusions

This paper presents the power management features POWER7 have introduced. Based on this ability to measure both power consumption and performance counters, we show it is possible to build a method based on very few performance counters which can manage both the power and energy consumed by an application while hurting minimally performance. Real examples of theses benefits are presented on POWER7. In future work, we'll investigate additional ways to optimize power consumption like memory throttling.

References

1. Kalla R, Sinharoy B (2009) POWER7: IBM Next generation server processor. Hot Chips Conference. HC21.25.826, Stanford University
2. Floyd MS, Ghiasi S, Keller TW, Rajamani K, Rawson FL, Rubio JC, Ware MS (2007) System power management support in the IBM POWER6 microprocessor. J Res Dev 51(6):733–746
3. Broyles M, Francois C, Geissler A, Hollinger M, Rosedahl T, Silva G, Van Heuklon J, Veale B. IBM EnergyScale for POWER7 processor based servers. http://www3.ibm.com/systems/power/hardware/whitepapers/energyscale7.html

4. Allarey J, George V, Jihagirdar S (2008) Power management enhancements in the 45 nm Intel core microarchitecture. Intel Technol J 12(3):169–178
5. Rajamani K, Hanson H, Rubio JC, Ghiasi S, Rawson FL (14 July 2006) Online power and performance estimation for dynamic power management. IBM Research Technical Report, RC 24007
6. Lee SJ, Lee HK, Yew PC (2007) Runtime performance projection model for dynamic power management. In: ACSAC 2007, pp 186–197