Evaluation of performance vs power consumption using NAS Parallel Benchmarks

Vinícius Garcia Pinto, Nicolas Maillard
Grupo de Processamento Paralelo e Distribuído (GPPD)
Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 - 91501-970 - Porto Alegre - RS - Brasil
{vgpinto, nicolas}@inf.ufrgs.br

Abstract

Parallel systems are traditionally evaluated considering only the performance. However, the importance of power consumption in these systems is increasing. Benchmarks are programs commonly used for evaluating the performance of parallel systems. In this paper, we use benchmarks LU, FT, CG and EP from NAS Parallel Benchmarks to present a performance analysis considering the power consumption in a multi-core parallel system. We vary the number of threads used to run this benchmarks and identify how much this variation implies in the power consumption.

1. Introduction

Until a few years ago, most parallel systems focused only on performance without consider the increase in power consumption or the need to be energy efficient. However, in actual systems the importance of power consumption is growing because power costs have become so significant as the hardware costs. Therefore we have to consider the power consumption as a limiting factor in parallel systems [1].

The use of benchmarks are common way of assessing the performance of a system. Benchmarks can be categorized in three types [2]:

- Synthetic benchmarks: artificial programs, with functionality of real programs such as hyper pi or wPrime;
- Kernel benchmarks: code fragments from real programs like LINPACK;
- Real application benchmarks: actual applications like games or AutoCad;

The NAS Parallel Benchmarks (NPB), originally developed by NASA, is a set of eight programs derived from computational fluid dynamics codes [3]. NPB is well known and widely used to measure and compare the performance of parallel systems [4].

In this paper we use LU, FT, CG and EP benchmarks from NPB to evaluate the performance versus the power consumption in a multi-core parallel system. We execute the benchmarks varying the number of threads in use to observe how much this variation implies in the power consumption and what is the best relation between power consumption and number of threads used. The IPMI (Intelligent Platform Management Interface) is used to measure the power consumption.

The remainder of this paper is organized as follows: Section 2 presents the execution environment, the benchmarks used and the interface to measure power consumption. Section 3 shows the results. Finally, Section 4 presents the conclusions.

2. Execution Environment

The tests presented in this paper were executed in a multi-core system consisting of two Intel Xeon E5530 quad-core processors with Hyper-Threading technology enabled and clock frequency of 2.4GHz running Ubuntu Linux.

System power consumption was measured using FreeIPMI 0.8.11, a IPMI GNU library [5]. FreeIPMI provides a way to get the instantaneous power and the electric current in use in the system. We use a background script to get this measures with one second interval while a benchmark is running.

NAS Parallel Benchmarks (NPB) were implemented in different programming models. In this work, we used a OpenMP version written in Fortran. NPB defines different classes for the problems size [6]. We selected and executed four benchmarks from NPB (changing threads number from 1 to 16):

- **LU Lower-Upper solver** is a application benchmark for lower and upper triangular system solution. In this benchmark we use class C that corresponds to 162^3 problem size;

- **FT Fast Fourier Transform** is a kernel benchmark that perform a 3D partial differential equation solution using FFTs. In this benchmark we use class B that corresponds to $512x256^2$ problem size;
- **CG** this kernel benchmark use the **Conjugate Gradient** method to compute an approximation to the smallest eigenvalue of a large, sparse, symmetric positive definite matrix. In this benchmark we use class C that corresponds to 150000 problem size;
- **EP Embarrassingly Parallel** is a kernel benchmark which evaluates an integral by means of pseudo random trials, this benchmark requires virtually no interprocessor communication. In this program we use class C that corresponds to 2^{33} problem size.

We believe that benchmarks LU, FT, and CG represent a significative range of communication and computing patterns of parallel applications that can commonly be found in real world. EP benchmark is a different case from the other three because is an embarrassingly parallel problem where there exists no dependency or communication between parallel tasks.

3. Results

Results presented in this Section were obtained considering a average of thirty runs of each combination between benchmark program and number of threads. The instant power observed when the system is idle was around 150 Watts.

- LU Benchmark: The speed-up is presented in the graph of figure 1. The graph presented in figure 2 shows the instant power consumption while the benchmark was running, each line corresponds to executions with sixteen, eight, four, two or one thread(s). In this case, execution using only one thread spent 1303 seconds, consuming, during this time, around 185 Watts. Execution using two threads spent 659 seconds, consuming around 209 Watts, using four threads spend time was 387 seconds, consuming around 234 Watts. Through the use of eight threads spend time was 287 seconds, consuming around 264 Watts. Finally, execution using sixteen threads spend time was 271 seconds, consuming, during this time around 275 Watts;
- FT Benchmark: The speed-up observed when benchmark was executed with one, two, four, eight and sixteen threads is presented in the graph of figure 3. The graph of figure 4 shows the instantaneous power consumption while the benchmarks was running, each line in this graph corresponds to executions using sixteen, eight, four, two or one thread(s)

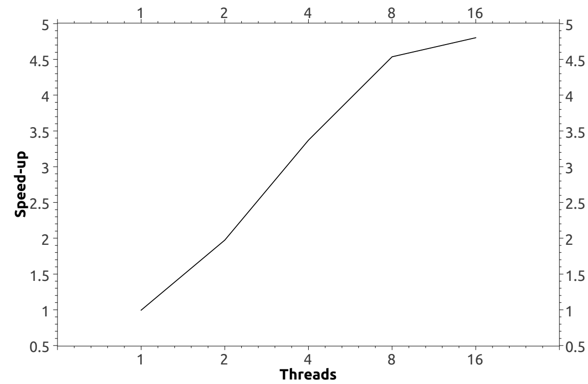

Figure 1. LU Speed-up

Execution of this benchmark using only one thread spent 71 seconds, consuming around 181 Watts. Through the use of two threads, spend time was 36 seconds consuming around 200 Watts. Execution using four threads spent 19 seconds consuming around 219 Watts, using eight threads spend time was 13 seconds consuming around 233 Watts. Finally, execution using sixteen threads spent 16 seconds consuming around 246 Watts;

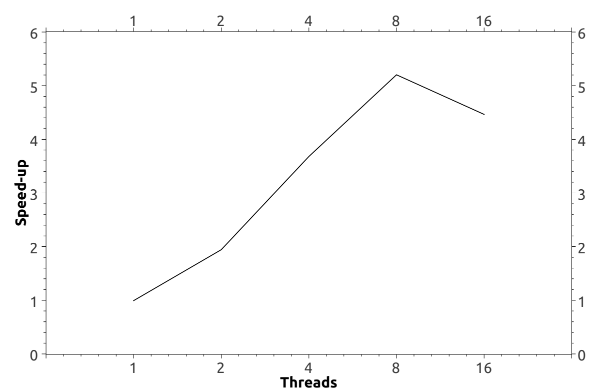

Figure 3. Speed-up FT

- CG Benchmark: Figure 5 presents the speed-up observed when benchmark was executed with one, two, four, eight and sixteen threads. The graph of figure 6 shows the instantaneous power consumption while the benchmarks was running, each line in this graph cor-

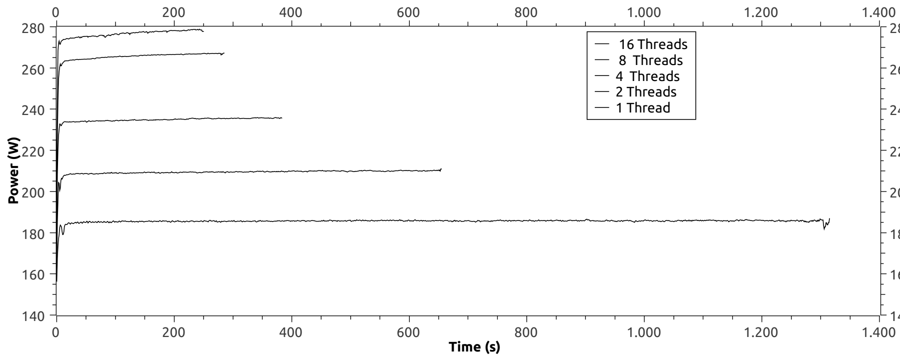

Figure 2. LU Power Consumption

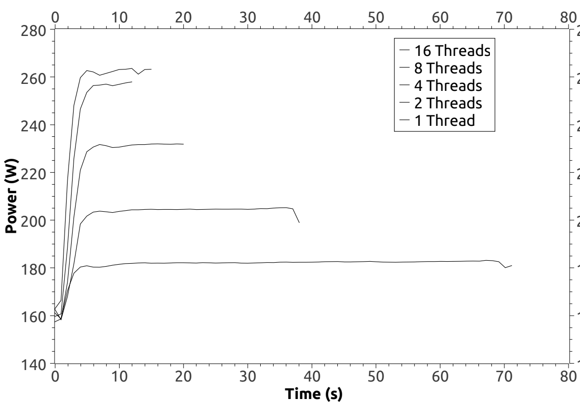

Figure 4. Power FT

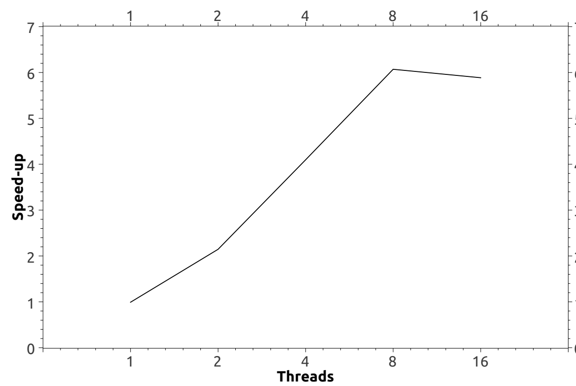

Figure 5. Speed-up CG

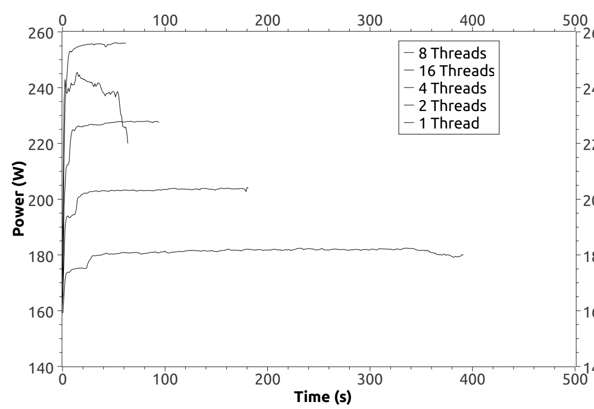

Figure 6. Power CG

responds to executions using sixteen, eight, four, two or one thread(s). Execution using a single thread spent 390 seconds consuming, during this time, around 180 Watts, using two threads, spent 181 seconds consuming around 202 Watts. Through the use of four threads spent 95 seconds consuming around 224 Watts. Execution using eight threads spend time was 64 seconds consuming around 251 Watts. Finally, execution using sixteen threads spend time was 66 seconds, consuming around 236 Watts;

- EP Benchmark: The speed-up obtained with one, two, four, eight and sixteen threads is presented in the graph of figure 7. The graph presented in figure 8 shows the instantaneous power consumption while this benchmark was running, each line corresponds to executions with sixteen, eight, four, two or one thread(s). Execution of this benchmark using only one thread spent 296 seconds, consuming around 176 Watts, using two threads, spend time was 148 seconds consuming around 195 Watts. Execution using four threads spent

75 seconds consuming around 211 Watts. Through the use of eight threads spend time was 40 seconds consuming around 243 Watts. Finally, execution using sixteen threads spent 28 seconds consuming around 261 Watts;

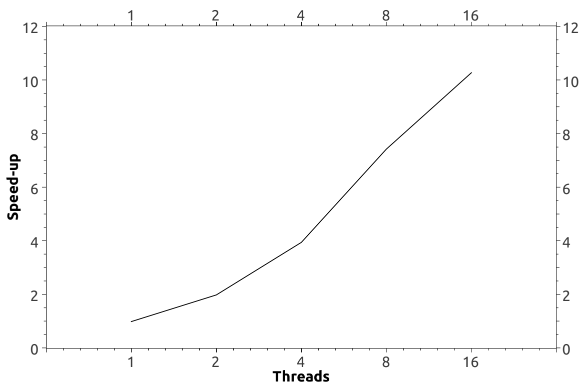

Figure 7. Speed-up EP

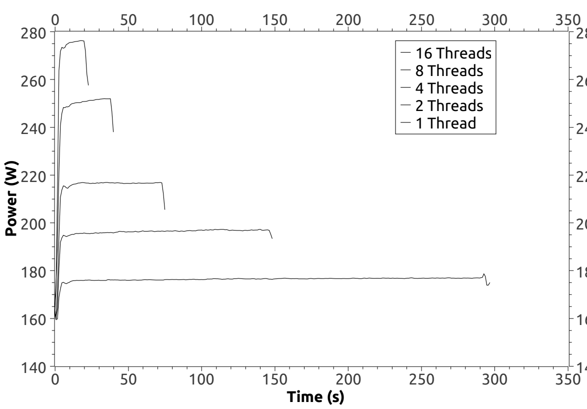

Figure 8. Power EP

4. Conclusions

This work presented an evaluation of the performance considering the power consumption using benchmarks LU, FT CG and EP from NAS Parallel Benchmarks in a multicore system.

Results showed that the increasing of power consumption followed, in a smaller proportion, the speed-up behavior. In other words, although the consumed power increases with the number of threads in use, the shorter time consuming this power keeps the advantages of using more threads. In the case of benchmarks FT, LU and EP the best speed-up and the best relation between increase of power consumption and increase of performance was obtained using the same number of threads, 8 threads for FT and 16 threads for LU and EP. However, with CG benchmark the best speed-up was obtained using 8 threads but the best relation between increase of power consumption and increase of performance was obtained using 16 threads.

For future work, we plan evaluate the relation between power consumption and performance decreasing the clock frequency to measure how the smallest frequency affects the power consumption and running more tests using different hardware configurations.

References

[1] L. A. Barroso, "The price of performance," *ACM Queue*, vol. 3, pp. 48–53, September 2005.

[2] T. Rauber and G. Rünger, *Parallel Programming: For Multicore and Cluster Systems*. Springer, 2010.

[3] "NAS Parallel Benchmarks," 2011. http://www.nas.nasa.gov/Software/NPB/.

[4] V. Kale, "Towards using and improving the nas parallel benchmarks: a parallel patterns approach," in *Proceedings of the 2010 Workshop on Parallel Programming Patterns*, ParaPLoP '10, (New York, NY, USA), pp. 12:1–12:6, ACM, 2010.

[5] "GNU FreeIPMI," 2011. http://www.gnu.org/software/freeipmi/.

[6] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga, "The nas parallel benchmarks summary and preliminary results," in *Proceedings of the 1991 ACM/IEEE conference on Supercomputing*, Supercomputing '91, (New York, NY, USA), pp. 158–165, ACM, 1991.