Evaluation of the Performance of the PlayStation 3 for Image Processing

Nicolás Narria and Per Bj. Bro

Universidad de Talca, Curicó, Chile,
pbro@utalca.cl

Abstract. The Cell/BE processor in a PlayStation 3 (PS3) was programmed with image processing routines of industrial interest, and the performance of this implementation was compared with a standard, Pentium Dual Core PC. The objective of this work was to evaluate the convenience of using a Cell/BE for high performance industrial image processing applications as opposed to the more conventional implementation on a PC. The methodology used was implementation of the algorithms in both sequential and parallel (threaded) fashion, and for the parallel case, vary the number of processors used. The results of this work indicate that despite the significant promise of the Cell/BE, the speedup obtained on the PS3 is slight as compared with the PC. One cause of this lack of performance enhancement is that the areas of interest of each image are relatively small compared to the cache size of the PC. The principal conclusion of this research is that the level of programming skills needed to take advantage of the Cell/BE architecture is quite high.

1 Introduction

The purpose of this research is to evaluate the performance of the PlayStation 3 (PS3) as compared with a Pentium based PC, for an industrial application of high performance computing. Optical sorting of fresh cherries has been limited by the computing power available for image processing. A typical sorting line will need to process between 50 - 100 fruit per second and the image processing algorithms for sizing the cherries are moderately complex. Consequently, high performance computing technology is neccesary for this industrial application.

The potential performance gain of the PS3 has been well publicized [1, 2, 4]. The power of the Cell/BE (the microprocessor of the PS3) is based on its heterogeneous multi-processor architecture [5]. A general purpose PowerPC unit (PPU) administers eight vector processors (Synergistic Processing Units, SPU's), joined by a very high speed bus. To develop programs for the Cell/BE one needs to compile for the PPU as well as for the SPU's, and to manage the interaction between these units.

One complication of cherry sorting using image processing is that the size is not just the largest or smallest diameter of the fruit, but is the diameter measured across the fruit on a line perpendicular to the line joining the fruit centroid and the peduncle base. Put more colloquially, when you hold the stem of the fruit between two fingers, the diameter is measured horizontally across the fruit. To calibrate fruit based on diameter, the image processing routines must detect the peduncle base and the fruit centroid, as illustrated in Figures 1, 2,3,4. Previous work on this topic [3] in which two algorithms for optical sorting were compared showed that the technique which searches around the border for the vertices of the peduncle base to be the fastest. This is the algorithm used in the current research.

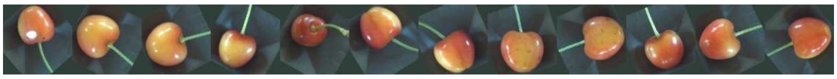

Fig. 1. Typical cherry images

Fig. 2. Cherry borders

2 Materials and Methods

Analysis was performed on 187 individual cherry images. Each image was rotated 12 times by an random angle to create a pool of 2244 total images. Groups of 12 images were taken from this pool and joined to form a strip similar to the view that an industrial camera would have of the processing line, as illustrated in Figure 1. The PC used for analysis was an AMD 64 bit x2 Turion processor, 1.6 GHz, 2.5 Gb RAM, 512 Kb cache, under 64 bit Debian, kernel 2.6.29.1. The PS3 was running Fedora 7.

Fig. 3. Peduncle bases

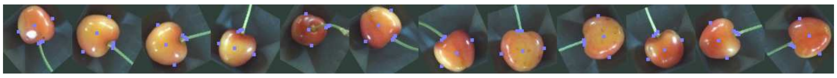

Fig. 4. Cherry diameters

3 Results

Figure 5 illustrates the effectiveness of increasing the number of SPU's in processing the images. As the number of SPU's are increased, the time to run 2500 images decreases. Note that the first point, at zero SPU's signifies processing with only the PPU, and not using any SPU's.

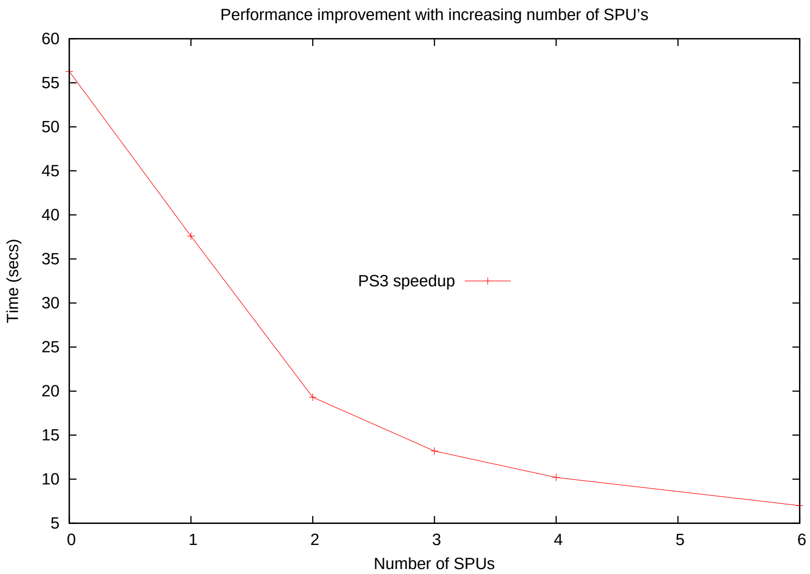

Fig. 5. Results: Processing speedup

Figure 6 illustrates in a nutshell the fundamental results of this work. The figure presents the time needed to process an increasing number of images, with different configurations. Note that the slowest configuration is the PS3 with only the PPU enabled. The fastest configuration is a toss-up with the bi-threaded PC being approximately as fast as the PS3 with 6 SPU's enabled.

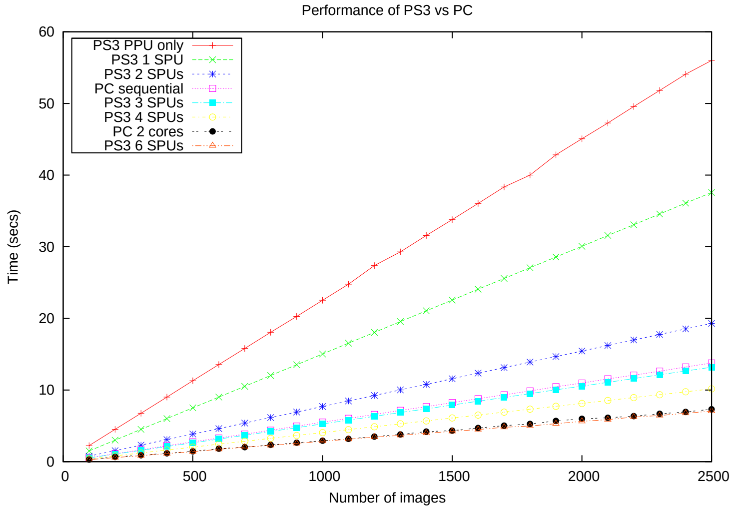

Fig. 6. Results: Cell/BE vs PC

4 Conclusions and Future Work

The results of this research are quite interesting, because they are not what was expected. The challenge is to explain why the Cell/BE didn't beat the pants off of the PC. Naturally, one very reasonable suggestion is that the coding on the Cell/BE was poorly done, and we can not reject this possibility. In and of itself, this is a notable result; the research group includes very experienced programmers as well as rising stars Another possible explanation is that when dealing with relatively small images, the administrative overhead of shuttling data back and forth between SPU's and PPU is high in comparison with the data crunching effort on the SPU's themselves.

The fundamental conclusion from this research is that the Cell/BE is a powerful machine, but requires significant expertise in obtaining useful results. A very concrete result of this research was that the industrial firm interested in using the Cell/BE decided to maintain their current technology in Pentium based systems, and to await future developments before investing resources in Cell/BE systems.

References

1. Michael D. McCool: Data-Parallel Programming on the Cell BE and the GPU using the RapidMind Development Platform. GSPx Multicore Applications Conference, Santa Clara, CA, USA (2006)
2. Pieter Bellens, Josep M. Perez, Rosa M. Badia and Jesus Labarta: CellSs: a programming model for the cell BE architecture. Proceedings of the 2006 ACM/IEEE conference on Supercomputing, Tampa, Florida (2006)

3. Per Bj. Bro and Nicolás Narria: Comparison of two algorithms for optical cherry sorting. American Society of Agricultural and Biological Engineers Annual Meeting, Reno, NV, USA (2009)
4. T. Chen, R. Raghavan, J. N. Dale and E. Iwata: Cell Broadband Engine Architecture and its first implementation: A performance view. Cell Broadband Engine Technology and Systems 51(5):559 (2007)
5. Michael Gschwind: The Cell Broadband Engine: Exploiting Multiple Levels of Parallelism in a Chip Multiprocessor. International Journal of Parallel Programming 35(3):233 (2007)