A Communication model for determining optimal affinity on the Cell BE Processor

C.D. Sudheer*, Sanaka Sriram*, Baruah P.K
Dept. of Mathematics and Computer Science, Sri Sathya Sai University, Prasanthi Nilayam, India.
sudheer@sssu.edu.in, sanaka.sriram@gmail.com, baruahpk@sssu.edu.in

Abstract

In the Cell BE, the SPEs communicate over Element Interconnect Bus (EIB). The bandwidth utilization on EIB is reduced due to the congestion created by the simultaneous communications. We observed that the actual bandwidth obtained for inter-SPE communication is strongly influenced by the assignment of threads to SPEs (Thread-SPE affinity). The major contributions of this work are to help understanding the reasons of reduction in bandwidth utilization and develop strategies to build an effective thread SPE mapping schemes in order to optimize the applications that have the inherent inter thread communication. By default, the assignment scheme provided is somewhat random, which sometimes leads to poor affinities and sometimes to good ones. We studied some common communication patterns, for which we could identify a particular affinity that yields performance that is close to twice the average performance of the default affinity. We have observed a performance growth of around 10%-12% by using the above mentioned study in a communication intensive Monte Carlo particle simulation application. We expect that Image and Signal processing applications which follow a pipelined model of operation will be greatly benefited by the optimal Thread-SPE affinity. We also discuss the optimization of affinity on a Cell Blade. We then describe a communication model tool created based on the observations from [3], which aids in choosing a good affinity, given the communication pattern of the application.

1. Introduction

The SPEs are connected to each other and to main memory by a high speed bus called the EIB, which has a bandwidth of 204.8 GB/s. The latency between each pair of SPEs is identical for short messages and so affinity does not matter in this case. In the absence of contention for the EIB, the bandwidth between each of pair of SPEs is identical for long messages too, and reaches the theoretical limit. However, we observed that in the presence of contention, the bandwidth can fall well short of the theoretical limit, even when the EIB's bandwidth is not saturated. This happens when the message size is greater than 16 KB. It is, therefore, important to assign threads to SPEs to avoid contention, in order to maximize the bandwidth for the communication pattern of the application.

We first identify causes for the loss in performance, and use this information to develop good thread-SPE affinity schemes for common communication patterns, such as ring, binomial-tree, and recursive doubling. We show that our schemes can improve performance by over a factor of two over a poor choice of assignments. By default, the assignment scheme provided is somewhat random, which sometimes leads to poor affinities and sometimes to good ones. With many communication patterns, our schemes yield performance that is close to twice as good as the average performance of the default scheme. Our schemes also lead to more predictable performance, in the sense that the standard deviation of the bandwidth obtained is lower. We also discuss optimization of affinity on a Cell blade consisting of two Cell processors. We observed that the affinity within each processor is often less important than the assignment of threads to processors. Based on the knowledge gained on this topic from [3], we created a communication model that determines the

* Student Author

theoretically best possible mapping for any given communication pattern. The outline of the rest of the paper is as follows. In Section 2, we summarize important architectural features of the Cell processor relevant to this paper. In Section 3, we describe that thread-SPE affinity can have significant influence on inter-SPE communication. Here, we also depict factors responsible for reduced performance. We can use these results to suggest good affinities for common communication patterns. We next discuss optimizing affinity on the Cell blade. In Section 4, we then describe our communication model and the evaluation of the model by using it for common communication patterns and a practical application. We finally present our conclusions and future work in Section 5.

2. Cell Communication Architecture

We summarize below the architectural features of the Cell of relevance to this work, concentrating on the communication architecture. Further details can be found in [2].

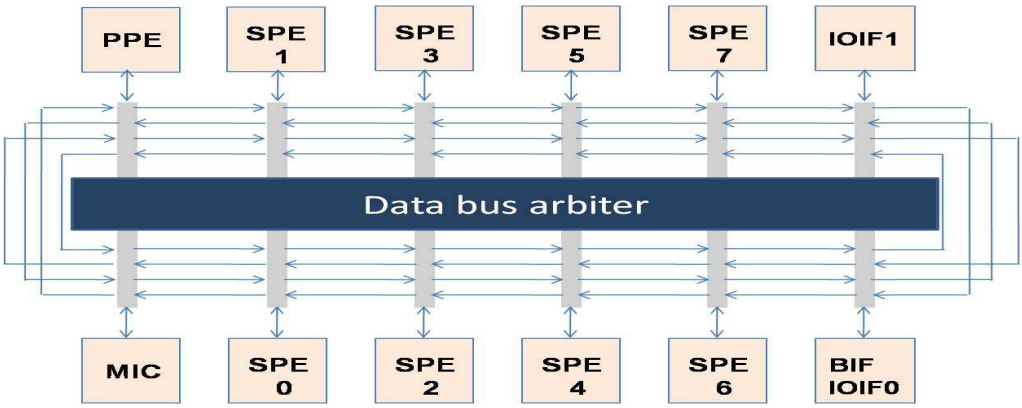

Figure 1: Overview of Cell communication architecture

Here, we reproduce the description of Cell we provided in [3]. Figure 1 provides an overview of the Cell processor. It contains a cache-coherent PowerPC core called the PPE, and eight co-processors, called SPEs, running at 3.2 GHz each. An XDR memory controller provides access to main memory at 25.6 GB/s total, in both directions combined. The PPE, SPE, and memory controller are connected via the EIB. The maximum bandwidth of the EIB is 204.8 GB/s. In a Cell blade, two Cell processors communicate over a BIF bus. The numbering of SPEs on processor 1 is similar, except that we add 8 to the rank for each SPE. The data can be transferred much faster between SPEs than between SPE and main memory [2]. It is, therefore, advantageous for the algorithms to be structured such that SPEs communicate directly between themselves over the EIB, and make less use of memory.

The data transfer time between each pair of SPEs is independent of the positions of the SPEs, if there is no other communication taking place simultaneously [3]. However, when many simultaneous messages are being transferred, transfers to certain SPEs may not yield optimal bandwidth, even when the EIB has sufficient bandwidth available to accommodate all messages. In order to explain this phenomenon, we now present further details on the EIB.

The EIB contains four rings, two running clockwise and two running counter-clockwise. All rings have identical bandwidths. Each ring can simultaneously support three data transfers, provided that the paths for these transfers don't overlap.

The EIB data bus arbiter handles a data transfer request and assigns a suitable ring to a request. When a message is transferred between two SPEs, the arbiter provides it a ring in the direction of the shortest path. For example, transfer of data from SPE 1 to SPE 5 would take a ring that goes clockwise, while a transfer from SPE 4 to SPE 5 would use a ring that goes counter-clockwise. If the distances in clockwise and anti-clockwise directions are identical, then the message can take either direction, which may not necessarily be the best direction to take, in the presence of contention. From these details of the EIB, we can expect that certain combinations of affinity and communication patterns can cause non-optimal utilization of the EIB.

3. Influence of Affinity on Inter-SPE Communication Performance

As mentioned in [3], affinity significantly influences the communication performance when there is contention. We identified factors that lead to loss in performance, which in turn enables us to develop good affinity schemes for a specified communication pattern.

Experimental Setup

The experiments were performed on the CellBuzz cluster at the Georgia Tech STI Center for Competence for the Cell BE. It consists of Cell BE QS20 dual-Cell blades and a few Cell BE QS22 blades. The QS22 blades have a newer version of Cell processor called PoweXCell8i. The codes were compiled with the ppuxlc and spuxlc compilers, using the -O3 -qstrict flags and SDK 3.1 was used. Further details regarding Timing etc. are provided in [3]. The results of the experiments performed on QS22 match with the same on QS20.

Summary of Experimental Results:

Several experiments using various affinities on different communication patterns were performed [3]. We noticed that the communication patterns where all the messages go in a single direction can use only half of the EIB bandwidth, as they do not use two of the rings of the EIB. And also, messages with overlapping paths create congestion on the EIB, which leads to performance degradation. We observed that messages that travel half-way across the ring can go in either direction.

Affinity on a Cell Blade

Communication on a Cell blade is asymmetric, with around 30 GB/s theoretically possible from Cell 0 to Cell 1, and around 20 GB/s from Cell 1 to Cell 0. However, we observed that communication between a single pair of SPEs on different processors of a blade yields bandwidth much below this theoretical limit [3] and [1]. In fact, this limit is not reached even when multiple SPEs communicate, for messages of size up to 64 KB each. The bandwidth attained by messages between SPEs on different processors is much lower than that between SPEs on the same processor [3]. So, these messages become the bottleneck in the communication. In this case, the affinity within each SPE is not as important as the partitioning of threads amongst the two processors.

4. Communication Model

Based on the understanding from the above mentioned experimental analysis on specific communication patterns, we created a communication model that determines a mapping with less communication volume (cost). The main purpose of creating a quantitative model is that we can find an optimal affinity for an arbitrary communication pattern, without being ingenious. We evaluate all possible affinities, and use the model to give a measure of how good each affinity is. We choose the best one. The communication model was developed based on the following guiding principles.

1. For good communication performance, the communication load should be distributed equally across all the four EIB rings.
2. Each ring can simultaneously support three data transfers, provided that the paths for these transfers don't overlap. Model should look for a mapping, where the paths taken by the messages do not overlap often.
3. The Model takes into account the asymmetry in the bandwidth between two processors, and so a partition sending more data will be placed on processor 0.

Evaluation of the model

We now evaluate the effectiveness of the model by using it for determining affinities for some communication patterns and a real application. Model's affinity in the figures denotes the affinity determined by the communication model, for the given communication pattern. Figure 2 shows the performance of different affinities with the Ring communication pattern. Figure 3 shows the results with the first phase of recursive doubling.

We next consider the performance of the communication model on a Monte Carlo application for particle transport, which tracks a number of random walkers on each SPE [4]. We used the diffusion scheme to balance the load between the SPEs.

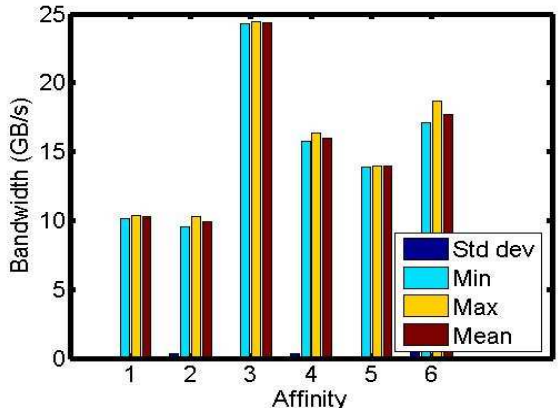

Figure 2: Performance of the following affinities: 1. Overlap 2. Default 3. EvenOdd 4. Identity 5. Ring 6. Model's Affinity

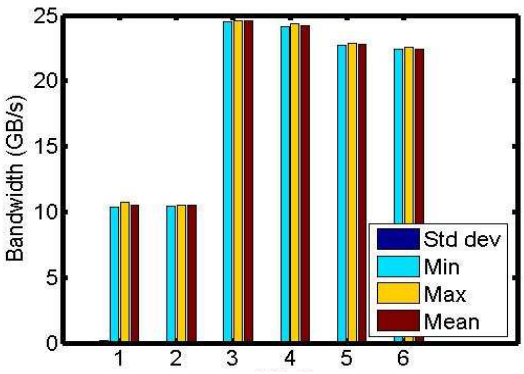

Figure 3: Performance of the following affinities: 1. Overlap 2. Default 3. EvenOdd 4. Identity 5. Ring 6. Model's Affinity

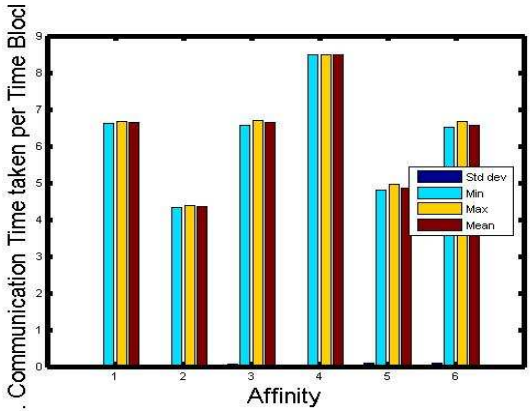

Figure 4: Performance of the following affinities: 1. Identity 2. EvenOdd 3. Ring 4. Overlap 5. Leap2 6. Model's Affinity

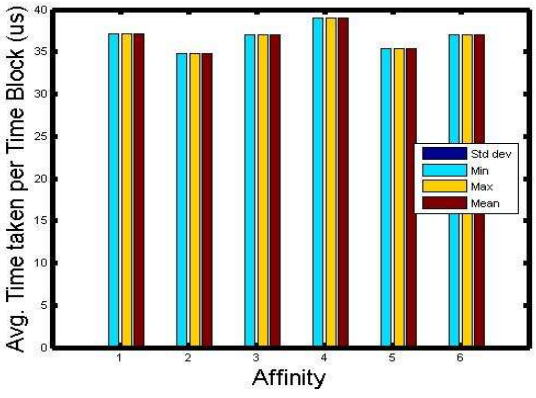

Figure 5: Performance of the following affinities: 1. Identity 2. EvenOdd 3. Ring 4. Overlap 5. Leap2 6. Model's Affinity

We can see a factor of two difference between the communication time cost for the best and worst affinities in fig 4. Figure 5 shows a difference in total application performance of over 10% between the best and worst affinities. We

can observe from the above figures that the affinity given by the model obtains predictable and average performance. We realized that the suboptimal behavior of the model is because of the reason that the three guiding principles used in creating our communication model, may not be including all the aspect of the Cell communication network. For example, we observed a poor bandwidth performance even with three non overlapping messages between SPEs going in the same direction. This occurs when at least two of them are of path length two or more, and go across the sides of the ring (i.e., through PPE-MIC or BIE-IOIF1 in fig. 1). This peculiar behavior of the non overlapping messages is not included in the model. We also understand that traffic to the Main Memory which also go through the same EIB, significantly affects inter-SPE communication performance.

Another issue is that in all the experiments, we considered only messages with equal size. We observed that messages with unequal sizes results in different behavior in some cases. For example, we observed that the above noted peculiar behavior of non overlapping messages does not hold true if the messages are of different sizes. We also assumed symmetry in rotating the affinity, to reduce the number of affinities tested from 8! to 7!, however, our experiments with some communication patterns indicated that such symmetry does not exist. We intend to modify our model by taking into account all the aspects of the communication network mentioned above, to make it complete.

5. Conclusions and Future Work

We observed that the SPE-thread affinity has a significant effect on inter-SPE communication performance, for common communication patterns and also for real applications. The performance obtained by specifying a good affinity can be predictable and is also a factor of two more over using the default assignment for many communication patterns. On a Cell Blade, the assignment of threads to processors is more important than the issues of affinity on each processor. We created a communication model that theoretically determines the best affinity for a given communication pattern. The communication model determines the mapping, where communication load is well balanced across the four rings and has minimum possible overlapping paths. It also considers the asymmetry between the communications between two processors on a Cell Blade. We understand that not including the above mentioned peculiar patterns, main memory traffic and uneven sized messages etc. are the reasons for the suboptimal behavior of the model. We intend to modify our model by considering all the aspects of the communication network mentioned above, to make it complete. The existing model works only for communication patterns with a single phase. We wish to extend our model so that it can also be used for communication patterns with multiple phases. We wish to develop few applications such as the pipelined image processing applications [5] etc. which have inherent inter-SPE communication, to evaluate our communication model.

References

[1] P. Altevogt, H. Boettiger, T. Kiss, and Z. Krnjajic. IBM blade center QS21 hardware performance. TR, WP101245, IBM, 2008.

[2] M. Kistler, M. Perrone, and F. Petrini. Cell multiprocessor communication network: Build for speed. *IEEE Micro*, 26:10 23, 2006.

[3] C.D. Sudheer, T. Nagaraju, P.K. Baruah, and A. Srinivasan, “Optimizing Assignment of Threads to SPEs of the Cell BE Processor”, 10th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC), Proceedings of the 23rd International Parallel and Distributed Processing Symposium, IEEE, (2009).

[4] G.. Okten and A. Srinivasan. Parallel quasi-Monte Carlo methods on a heterogeneous cluster. *In Proceedings of the MCQMC*,Springer-Verlag.

[5] Pipeline Pattern, modified by Yunsup Lee, Ver 1.0 (March 11,2009), based on the pattern “Pipeline Pattern” described PPP, by Tim Mattson et.al.