Impact of PE Mapping on Cray T3E Message-Passing Performance

Eduardo Huedo, Manuel Prieto, Ignacio M. Llorente, and Francisco Tirado

Departamento de Arquitectura de Computadores y Automática

Facultad de Ciencias Físicas

Universidad Complutense

28040 Madrid, Spain

Phone: +34-91 394 46 25

Fax +34-91 394 46 87

{ehuedo, mpmatias, llorente, ptirado}@dacya.ucm.es

Abstract. The aim of this paper is to study the influence of processor mapping on message passing performance of two different parallel computers: the Cray T3E and the SGI Origin 2000. For this purpose, we have first designed an experiment where processors are paired off in a random manner and messages are exchanged between them. In view of the results of this experiment, it is obvious that the physical placement must be accounted for. Consequently, a mapping algorithm for the Cray T3E, suited cartesian topologies is studied. We conclude by making comparisons between our T3E algorithm, the MPI default mapping and another algorithm proposed by Müller and Resch in [9].

Keywords. MPI performance evaluation, network contention, mapping algorithm, Cray T3E, SGI Origin 2000.

1. Introduction

The belief has spread that processor mapping does not have any significant effect on modern multiprocessor performance. Consequently, this is no cause of concern to the programmer, who only has to distinguish between local and remote access. It is true that since most parallel systems today use wormhole or cut-through routing mechanisms, the message *first bit delay* is nearly independent of the number of hops that it must travel through. However, message latency depends also on the network state when the message is injected into it. Indeed, as we have shown in [1], *blocking times*, i.e. delays due to conflicts over the use of the hardware routers and communication links, are the major contributors to message latencies under heavy or unevenly distributed network traffic.

Consequently, as shown in section 2, a correct correspondence between logical and physical processor topologies could improve performance, since it could help to reduce network contention. In section 3 we propose a mapping algorithm that can be used to optimize the `MPI_Cart_create` function in the Cray T3E. Finally, some experimental results and conclusions are presented in sections 4 and 5 respectively.

2. Random Pairwise Exchanges

In order to illustrate the importance of processor mapping, we have performed an experiment in which all the available processors in the system pair off in a random manner and exchange messages using the following scheme:

```
MPI_Isend(buf1, size, MPI_DOUBLE, pair, tag1, Comm, &sreq);
MPI_Irecv(buf2, size, MPI_DOUBLE, pair, tag2, Comm, &rreq);
MPI_Wait(&sreq, &sstatus);
MPI_Wait(&rreq, &rstatus);
```

As we will show, for the parallel systems studied, an important degradation in performance is obtained when processors are poorly mapped.

2.1. Random Pairing in the Cray T3E

The Cray T3E in which the experiments of this paper have been carried out is made up of 40 Alpha 21264 processors running at 450MHz [3][4].

Figure 1 shows the results obtained using 32 processors. This system has two communication modes: messages can either be *buffered* by the system or be sent directly in a *synchronous* way. The best unidirectional bandwidth attainable with synchronous mode (using the typical ping-pong test) is around 300 MB/s, while using the buffered mode it is only half the maximum (around 160MB/s) [1][4][5].

Fig. 1. Transfers times (seconds) in the T3E obtained from different pairings. The measurement has been taken using buffered (right-hand chart) and synchronous (left-hand chart) mode.

For medium messages the difference between the best and worst pairing is significant. For example, the optimal mapping is approximately 2.7 times better than the worst for a 10MB message size in both communications modes.

It is also interesting to note that for the optimal mapping, the bandwidth of this experiment is better than the unidirectional one, due to the exploitation of the T3E bi-directional links. In the synchronous mode the improvement is approximately 40% (around 420MB/s) reaching almost 100% in the buffered one (around 300MB/s).

Figure 3 shows the usual correspondence between the physical and logical processors for a 4x4 topology and one of the possible optimal mapping, where every neighbour pair in the logical topology share a communication link in the physical one:

Fig. 2. Correspondence between physical and logical mapping in the Cray T3E.

2.2. Random Pairing in the SGI Origin 2000

Figure 2 shows the results obtained on the SGI Origin 2000. The system studied in this paper consists of 32 R10000 microprocessors running at 250 MHz [4][6].

Fig. 3. Transfer times (seconds) in the SGI Origin 2000 for different message sizes obtained from different pairings.

Here, the behavior is different: performance depends on the pairing but the difference between the best and the worst is only around 20% and does not grow with message size. In addition, when comparing to the unidirectional bandwidth there is no improvement. On the contrary, performance is worse for message sizes larger than 126KB. The effective bandwidth is only around 40 MB/s, while using a ping-pong test a maximum of around 120 MB/s can be obtained (for 2 MB messages)[1][2][7].

2.3. Preliminary Conclusions

In view of these results it is obvious that the physical placement must be accounted for. On the SGI Origin 2000 it is possible to perform the mapping control by combining MPI with an additional tool for data placement, called `dplace` [8]. Unfortunately, to the best of the author's knowledge, nothing similar exist on the T3E.

For n-dimensional cartesian topologies, the MPI standard provides for this need through the function `MPI_Cart_create`. It contains one option that allows the processor rank to be reordered with the aim of matching the logical and physical topologies to the highest possible degree. However, the MPI on the Cray does not carry out this optimization.

3. MPI_Cart_create Optimization on the Cray T3E

The only attempt, to our knowledge, to improve the implementation of the `MPI_Cart_create` function on the Cray T3E is a heuristic processor-reordering algorithm proposed by M. Müller and M. Resch in [9]. This algorithm sorts PE ranks according to physical coordinates using the 6 permutations of the (x, y, z) triplet and then selects the optimum with respect to the average hop count. Although this technique improves the MPI default mapping, we have in most cases obtained better results using an alternative approach based on a greedy algorithm.

3.1. Our Mapping Algorithm

The aim is to find a rank permutation that minimizes the distance between (logical) neighbouring processing elements (PEs). We shall start by making some definitions:

- *Physical distance* (d_p): the minimum number of hops one message must travel from the origin PE to the destination PE.
- *Logical distance* (d_l): the absolute value of the PE rank difference.

According to these definitions, the *distance* (d) used by our algorithm is:

$$d(PE, PE_1) < d(PE, PE_2) \Leftrightarrow \begin{cases} d_p(PE, PE_1) < d_p(PE, PE_2) \\ \vee \\ (d_p(PE, PE_1) = d_p(PE, PE_2)) \wedge (d_l(PE, PE_1) < d_l(PE, PE_2)) \end{cases} \quad (1)$$

The mapping evaluation function, which has to be minimized by the algorithm, consists in calculating the *average distance*:

$$d_{av} = \frac{\sum_{i \in \{PEs\}} \sum_{j \in neighbours(i)} d_p(i, j)}{\sum_{i \in \{PEs\}} \#neighbours(i)} \quad (2)$$

In the optimal case d_{av} is equal to 1. However, it may or may not be obtained depending on the cartesian topology chosen and the system configuration and state.

To find out the *physical distance* between PEs we use the physical co-ordinates of each PE (obtained with `sysconf(_SC_CRAY_PPE)`), which does not form part of MPI) and the knowledge of the Cray T3E bi-directional 3D-torus physical topology.

3.2. 1D Algorithm

The following scheme describes our 1D-mapping algorithm. Starting from a given processor, each step consists of two phases. First, the PE at minimum distance from the current one is chosen and then appended to the topology:

```
1D_algorithm(int dim, int current) {
    append(current);
    for (i= 1; i<dim; i++) {
        next= (not used) PE at minimum d from current;
        append(next);
        current= next;
    }
}
```

The algorithm is written in such way that it is possible to choose the first PE. For example, if the number of PEs is even it should start from the rank 0 PE. Otherwise, it should start from the highest rank PE. Figure 4 illustrates the algorithm in both cases:

Fig. 4. Mapping of our 1D algorithm for 16 (left-hand chart) and 15 (right-hand chart) PEs.

Grid	Non-cyclic			Cyclic		
	MPI	M&R	Greedy	MPI	M&R	Greedy
32x1x1	1.61	>1	1	1.625	>1	1
4x4x1	2	1	1	2.25	1	1
8x4x1	2.2	1	1	2.4	>1	1
2x2x2	1.3	1.3	1	2.0	>1	1
3x3x3	2.27	1.8	1.8	2.34	>1.8	1.9

Table 1. Average number of hops that a message has to travel. MPI and greedy refer to the default mapping and our algorithm respectively.

4. MPI_Cart_create Benchmark

Although the average number of hops can be used to estimate the quality of the mapping algorithm, the time required for exchanging data is the only definite metric. Therefore, we have used a synthetic benchmark to simulate the communication pattern that can be found in standard domain decomposition applications:

Fig. 6. Communication pattern in standard domain decomposition methods.

Results for communications in buffered mode are shown in the following figure:

Fig. 7. MPI_Cart_create benchmark results for different cartesian topologies, with (OP) and without reordering, using 32 PE (left-hand char) and 16 PE (right-hand chart).

Apart from 1D topologies, improvements are always significant. Table 2 presents the *asymptotic bandwidth* obtained with 32 PEs:

Topology	Non-cyclic communication		Cyclic communication	
	Buffered mode	Synchronous mode	Buffered mode	Synchronous mode
1D	219 → 221	300 → 300	219 → 219	300 → 300
2D	213 → 263	262 → 340	141 → 200	181 → 280
3D	219 → 262	267 → 351	194 → 230	255 → 300

Table 2. Improvements in the asymptotic bandwidths (MB/s) for 32 PEs.

For non-cyclic communications in buffered mode, improvements are around 20% for 2D and 3D topologies. As in the random pairwise exchange, performance is better using the synchronous mode. Obviously, the improvement obtained by the optimized mapping is greater in this mode, around 40%, since buffered mode helps to reduce the contention problem.

The behavior for 1D topologies agrees with some previous measurements of the network contention influence on message passing. As we have shown in [1], the impact of contention is only significant when processors involved in a message exchange are 2 or more hops apart. Although the MPI default mapping is sub-optimal for 1D topologies, the network distance between neighbours is almost always lower than 2 in this case, and hence, the optimal mapping improvements are not relevant.

For cyclic communications improvements are even greater than those obtained in the non-cyclic case, reaching around 50% for the 2D topology using synchronous mode.

5. Conclusions

We have first shown how PE mapping affects actual communication bandwidths on the T3E and the Origin 2000. The performance depends on the mapping on both systems, although the influence is more significant on the T3E where the difference between optimal and worst mappings grows with message size.

In view of these results, we conclude that the physical placement must be accounted for in both systems. The SGI Origin 2000 provides for this need through `dplace`. However, a similar tool does not exist in the T3E. For this reason we have proceeded to study a greedy mapping algorithm for n-dimensional cartesian topologies on the T3E, which can be used as an optimization for the `MPI_Cart_create` function. Compared to the MPI default mapping, our algorithm reduces the average number of hops that a message has to travel.

Finally, to measure the influence of the processor mapping on performance, we have used a synthetic benchmark to simulate the communication pattern found in standard domain decomposition applications. The improvements are significant in most cases, reaching around 40% in 2D and 3D topologies with synchronous mode.

Although our experiments have been focused on cartesian topologies, we believe that they open the possibility of optimizations in other topologies (e.g. graphs).

At the time of writing this paper we are applying this optimization to actual applications instead of synthetic benchmarks and we are probing it on a larger T3E system.

Acknowledgements

This work has been supported by the Spanish research grant TIC 1999-0474. We would like to thank Ciemat and CSC (Centro de Supercomputación Complutense) for providing access to their parallel computers.

References

- [1] M. Prieto, D. Espadas I. M. Llorente, F. Tirado. "Message Passing Evaluation and Analysis on Cray T3E and Origin 2000 systems", in Proceedings of Europar 99, pp. 173-182. Toulouse, France, 1999.
- [2] M. Prieto, I. M. Llorente, F. Tirado. "Partitioning of Regular Domains on Modern Parallel Computers", in Proceedings of VECPAR 98, pp. 305-318. Porto, Portugal, 1998.
- [3] E. Anderson, J. Brooks, C. Grass, S. Scott. "Performance of the CRAY T3E Multiprocessor", in Proceedings of SC97, November 1997.
- [4] David Culler, Jaswinder Pal Singh, Annap Gupta. "Parallel Computer Architecture. A hardware/software approach" Morgan-Kaufmann Publishers, 1998.
- [5] Michael Resch, Holger Berger, Rolf Rabenseifner, Tomas Bönish "Performance of MPI on the CRAY T3E-512", Third European CRAY-SGI MPP Workshop, PARIS (France), Sept. 11 and 12, 1997.
- [6] J. Laudon and D. Lenoski. "The SGI Origin: A ccNUMA Highly Scalable Server", in Proceedings of ISCA'97. May 1997.
- [7] Aad J. van der Steen and Ruud van der Pas "A performance analysis of the SGI Origin 2000", in Proceedings of VECPAR 98, pp. 319-332. Porto, Portugal, 1998.
- [8] Origin 2000 and Onyx2 Performance Tuning and Optimization Guide. Chapter 8. Available at <http://techpubs.sgi.com>.
- [9] Matthias Müller and Michael M. Resch, "PE mapping and the congestion problem on the T3E" Hermann Lederer and Friedrich Hertweck (Ed.), in Proceedings of the 4th European Cray-SGI MPP Workshop, IPP R/46, Garching/Germany 1998.