A Correlation Study between MPP LS-DYNA Performance and Various Interconnection Networks — a Quantitative Approach for Determining the Communication and Computation Costs

Author:

Yih-Yih Lin

Correspondence:

Yih-Yih Lin
Hewlett-Packard Company
MR01-3
200 Forest Street
Marlboro, MA 01752
USA

Tel. +1-404-774-5278
Email: yih-yih.lin@hp.com

Keywords:
Communication cost, computation cost, speedup
accuracy, single precision, double precision, 64-bit computing

ABSTRACT

As MPP LS-DYNA uses the message-passing paradigm to obtain parallelism, the elapsed time of an MPP LS-DYNA simulation comprises of two parts: computation cost and communication cost. A quantitative approach for determining the communication cost and, hence, the computation cost and the speedup of an MPP LS-DYNA simulation is presented. Elapsed times, characteristic—latency and bandwidth—of interconnect networks, and message patterns are first measured, and then the method of least square errors is applied to estimate the two costs. This approach allows one to predict the performance, or the speedup, of MPP LS-DYNA simulations with any interconnect network whose characteristics is known.

Also, while conducting this performance study of MPP LS-DYNA, loss of accuracy in single-precision (32-bit) MPP LS-DYNA simulations has been found. This finding and the advantage of double-precision (64-bit) arithmetic are presented.

INTRODUCTION - Theory for Performance of MPP LS-DYNA

To run an N-processor MPP LS-DYNA simulation, or job, an interconnect network, or called simply as interconnect, must first be established to connect the N processors; the collection of the N processors and the interconnect is called an N-processor cluster. In this paper, we will consider only the case that the N processors are of the same kind. For such a job, MPP LS-DYNA starts by decomposing the geometrical configuration of the model into N sub-domains. Each of the N processors is assigned to perform computation on one of the sub-domains; meanwhile, messages are passed among all those processors so that necessary physical conditions, such as force conditions, can be enforced. Let $T^1_{comput}, T^2_{comput}, \ldots, T^N_{comput}$ be each processor's computation cost, and let $T^1_{comm}, T^2_{comm}, \ldots, T^N_{comm}$ be each processor's communication cost. Define the computation cost T_{comput} as max ($T^1_{comput}, T^2_{comput}, \ldots, T^N_{comput}$) and the communication cost T_{comm} as max($T^1_{comm}, T^2_{comm}, \ldots, T^N_{comm}$), respectively. Then the job's elapsed time can be described as:

$$T_{elapsed} = T_{comput} + T_{comm} \qquad (1)$$

For a given decomposition, the computation cost T_{comput} is fixed. In contrast, the communication cost T_{comm} varies with the characteristics of interconnects used. The term "speedup" is defined as the ratio $T_{elapsed, 1\text{-processor}} / T_{elapsed, N\text{-processor}}$. In general, speedups are smaller than N. Since for the 1-processor job the communication cost T_{comm} is zero, the perfect speedup of N folds can be realized only under the unrealistic conditions of zero communication cost, i.e., $T_{comm} = 0$, and perfectly balanced decomposition, which renders $T^1_{comput} = T^2_{comput} = \ldots = T^N_{comput}$.

Assuming that the N processors are of the same kind, the variation of $T^1_{comput}, T^2_{comput}, \ldots, T^N_{comput}$ arises out of the unbalanced decomposition of the N sub-domains. It is extremely difficult to find a universal algorithm to decompose a model with a balanced decomposition. MPP LS-DYNA does provide features, as documented in *pfile in parallel specific options*, for users to provide hints to get a more balanced decomposition than the default.

There are typically a large number of messages of various sizes transacting in an MPP LS-DYNA simulation. The communication cost T_{comm} is the sum of the communication costs of each message in the processor that obtains the maximal communication cost (called the "maximal" processor). The communication cost of a message depends solely on the two factors, latency and bandwidth, of the interconnect [1]:

Communication cost of a message = Latency + Message Size / Bandwidth

The latency is the sum of sender overhead, receiver overhead and time of flight; and the bandwidth refers to the maximum rate at which the interconnect can propagate information once the message enters the network. Messages of MPP LS-DYNA comprises of various different types, such as point-to-point communication and collective operations. In general, for a given interconnect, latency varies with message types, and bandwidth varies with message types and lengths. All the messages can be divided into m groups with the same latency, the same bandwidth and the same length. Considering messages of the "maximal" processor, let n_i, t^{lan}_i, t^{bw}_i and s_i be the i^{th} group's number of messages, latency, bandwidth and message size, respectively. Then the job's communication cost can be described as follows:

$$T_{comm} = S^m_{i=1}\ n_i (t^{lan}_i + s_i / t^{bw}_i) \qquad (2)$$

It is well known that the most basic operation for message passing is the point-to-point, or so called ping-pong, communication. Let t^{lan} and t^{bw} be the latency and bandwidth of the ping-pong communication, and let a_i be the ratio t^{lan}_i / t^{lan} and $ß_i$ be the ratio t^{bw} / t^{bw}_i, respectively. Then formula (2) becomes

$$T_{comm} = (S^m_{i=1}\ n_i a_i)\ t^{lan} + (S^m_{i=1}\ n_i ß_i s_i) / t^{bw} \qquad (3)$$

Further, let M be the number of messages and s be the average message size. Setting

$$Ma = S^m_{i=1}\ n_i a_i \quad \text{and} \quad Mßs = S^m_{i=1}\ n_i ß_i s_i \qquad (4)$$

we have the following formula

$$T_{comm} = M(at^{lan} + ßs / t^{bw}) \qquad (5)$$

Numbers a and ß are called as the latency constant and the bandwidth constant, respectively. For a given cluster, its ping-pong latency and bandwidth, t^{lan} and t^{bw}, can be measured. The number of messages M and the average message size s in each processor can also be measured. If the latency and bandwidth constants, a and ß, can be determined, then formula (5) will allow one to obtain the communication cost T_{comm}.

To determine them, assume all jobs are done on two different clusters, which comprise of the same number and the same kind of processors, but of two different interconnects, *a* and *b*. The two clusters are named as clusters *a* and *b*, respectively; their ping-pong latencies are denoted as t^{lan}_a and t^{lan}_b, respectively; and so are their ping-pong bandwidths as t^{bw}_a and t^{bw}_b. With such two clusters, then it can be conjectured that the two numbers, a and ß, in formula (4) remain the same, from runs to runs, of different numbers of processors and of clusters *a* and *b*. Such a conjecture should be a fair good one because of the fact that all decompositions and

hence message patterns are similar. Furthermore, for a relatively balanced N-processor job, the number of messages, M, and the average message size, s, in the "maximal processor" can be approximated as the average of numbers of messages and as the *average* of average message sizes among the N processors. With this conjecture on the property of α and β and with this approximation for the "maximal" processor's message number and average message size, the two numbers, a and ß, can then be determined by the method of least square errors.

Clearly, two jobs, with clusters *a* and *b*, of the same number of processors and precision have identical message patterns. Therefore, the two jobs have the same number of messages and the same average message size; let the number of messages and the average message size be denoted as M_n and s_n, respectively. To describe the method of least square errors, let the number of messages and the average message size, of a n-processor job and with cluster *a*, be denoted as M^a_n and s^a_n, respectively; and let M_n and s_n be similarly denoted for another n-processor job with cluster *b*. Since the decompositions of the two jobs are identical, their computation costs T_{comput} are equal. If the elapsed times with clusters *a* and *b* are, respectively, denoted as $T^a_{elapsed}$ and $T^b_{elapsed}$, it follows from formulas (1) and (5) that

$$M_n(t^{lan}_a - t^{lan}_b)a + M_n s(1/t^{bw}_a - 1/t^{bw}_b)ß = T^a_{elapsed} - T^b_{elapsed} \qquad (6)$$

When applying to measured data, formula (6) is only approximately correct and forms the base for obtaining the least square errors. In formula (6), let the two elapsed times on the right-hand side be substituted with the measured ones, and let the error be defined as the difference between the right-hand side and the left-hand side. Furthermore, let several pairs of same number-of-processor jobs, with the number of processors, n, varying, be measured. Each pair of such jobs produces an error. Clearly, the sum of squares of those errors is a quadratic function of the two variables, a and ß, and the solution that minimizes the quadratic function, which can be easily solved, is known to be the best approximation under the criterion of least square errors.

MODEL, MACHINE, INTERCONNECTS, MEASURED DATA

Model, Machine

In this paper, the well-known car crash model, refined Neon, of 535 thousands elements and with simulation time of 30 milliseconds, is used. Both single- and double-precision 960.1647 versions of MPP LS-DYNA are used. A 32-processor cluster, consisted of 16 machines of HP's 900MHz rx2600, is used. The rx2600 is a 2-CPU Itanium machine.

Interconnects and Their Characteristics

Two interconnects are used: the Gigabit Ethernet (GigE) and HP's Hyperfabric 2 (HF2). Its ping-pong latency and bandwidth have been measured and are shown in Table 1.

Elapsed times

Table 2 and Figure 1 show elapsed times, actually measured, for jobs with numbers of processors 1, 2, 4, 8, 16, and 32; and each with the four cases: single precision, GigE; single precision, HF2; double precision, GigE; double precision, HF2.

	GigE	HF2
Latency	43 μsec	22 μsec
Bandwidth	112 MB	216 MB

Table 1. Ping-pong latency and bandwidth of Gigabit Ethernet and HF2

No. of processors / Interconnect, Precision	1	2	4	8	16	32
GigE, SP	37010	21065	9913	5108	2963	2094
HF 2, SP	37010	21065	9926	4998	2800	1799
GigE, DP	41407	24484	11827	6215	3582	2441
HF 2, DP	41407	24484	11703	6024	3332	2119

Table 2. Elapsed times, in seconds, measured

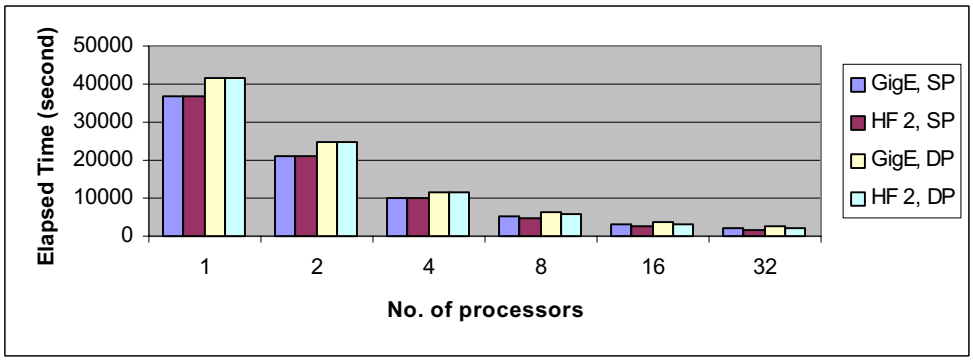

Figure 1. Graph for table 2

Message Patterns

Table 3 shows the measured average numbers of messages and average message sizes per processor, with numbers of processors 4, 8, 16, and 32; and with single and double precisions. Furthermore, it has been found that messages for all those jobs are concentrated within a small range of small message sizes. Figures 2 and 3 show such a concentration of small messages for the 32-processor, single-precision job. Such a concentration clearly implies that the use of average message size in formula (4) is a good approximation.

ESTIMATION OF COMMUNICATION COSTS

Latency Constant α and Bandwidth Constant β

To estimate α and β, call the cluster with GigE as cluster *a* and the one with HF2 as cluster *b*. Then, two jobs—one from cluster *a*, the other from cluster *b*—with the same number of processors and the same arithmetic precision form a pair of jobs, as described in the **INTRODUCTION** section. With numbers of processors being 4, 8, 16, and 32, and with arithmetic precisions being single and double, there are 8 such pairs of jobs. The 8 errors, as derived from formula (6), for these 8 pairs of jobs, can then be obtained with the ping-pong latency and bandwidth in Table 1, the elapsed time data in Table 2, and the message data in Table 3. The sum of squares of these 8 errors is a quadratic function of α and β. The minimum of the quadratic function

occurs when its partial derivatives with respect to α and β are equal to zero, which, in turn, forms two linear equations of the two unknowns α and β, whose solution can be easily obtained as:

$$\alpha = 3.6 \text{ and } \beta=1.6 \qquad (7)$$

This means that, for the Neon model, the effective latency of a given interconnect is 3.6 times its ping-pong latency, and its effective bandwidth is 0.625, or 1/1.6, times its ping-pong bandwidth.

No. of Processors	Ave. No. of Messages per Processor, SP	Ave. No. of Messages per Processor, DP	Ave. Message Size in Bytes, SP	Ave. Message Size in Bytes, DP
4	1232174	1231635	1707	3360
8	1760433	1760515	1044	2042
16	2419095	2419646	703	1368
32	3684285	3683544	445	866

Table 3. Average numbers of messages per processor and averages message sizes for single-precision and double-precision jobs with different numbers of processors.

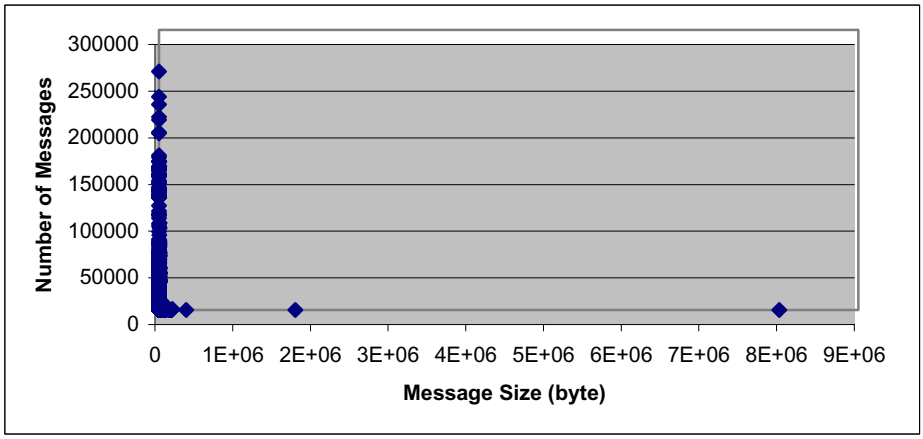

Figure 2. Distributions of all message sizes in the 32-processor, single-precision job

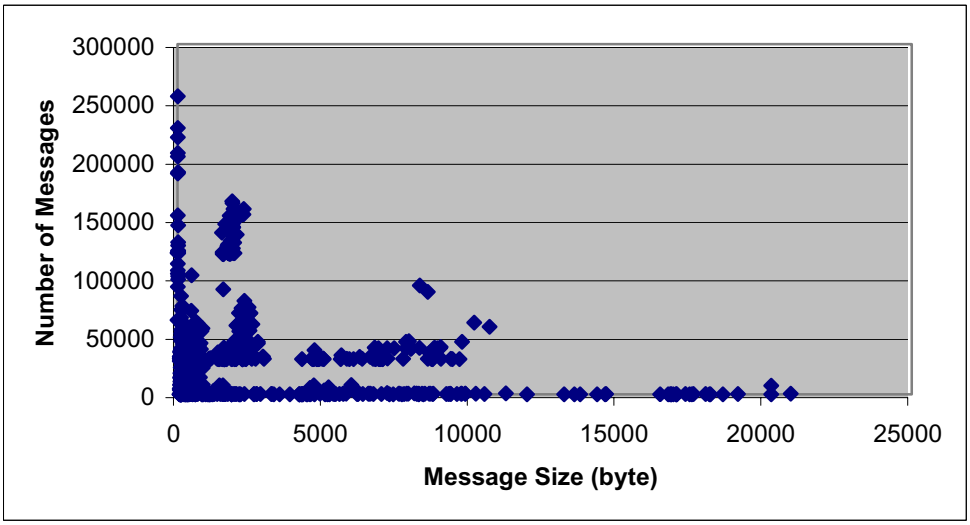

Figure 3. Distribution of message sizes, in the range of 0 to 25,000 bytes, in the same job as Figure 2

Estimates of Elapsed Times for Various Cases

With the latency constant α and the bandwidth constant β determined, we can then use formula (5) to estimate the communication cost $T_{comm;}$ and hence T_{comput}, using formula (1). Shown in Table 4 and Figure 4 are estimated elapsed times for the 5 double-precision cases:

1. An interconnect of infinite speed, i.e., zero latency and infinite bandwidth
2. An interconnect with the same latency as that of HF2 and with infinite bandwidth
3. An interconnect with the same latency as that of HF2 and with bandwidth doubled
4. An interconnect with the same bandwidth as that of the HF2 and zero latency
5. An interconnect with the same bandwidth as that of the HF2 and latency halved

Number of Processors	4	8	16	32
HF2, Measured	11703	6024	3332	2119
HF2, Infinite Speed, Estimated	11606	5885	3141	1829
HF2, Infinite Bandwidth, Estimated	11703	6024	3332	2119
HF2, Bandwidth Doubled, Estimated	11703	6024	3332	2119
HF2, Zero Latency, Estimated	11606	5885	3141	1829
HF2, Latency Halved, Estimated	11654	5954	3236	1974

Table 4. Measured elapsed times and estimated elapsed times for the 5 cases: infinite-speed interconnect, HF2 with infinite bandwidth, HF2 with bandwidth doubled, HF2 with zero latency, HF2 with latency halved.

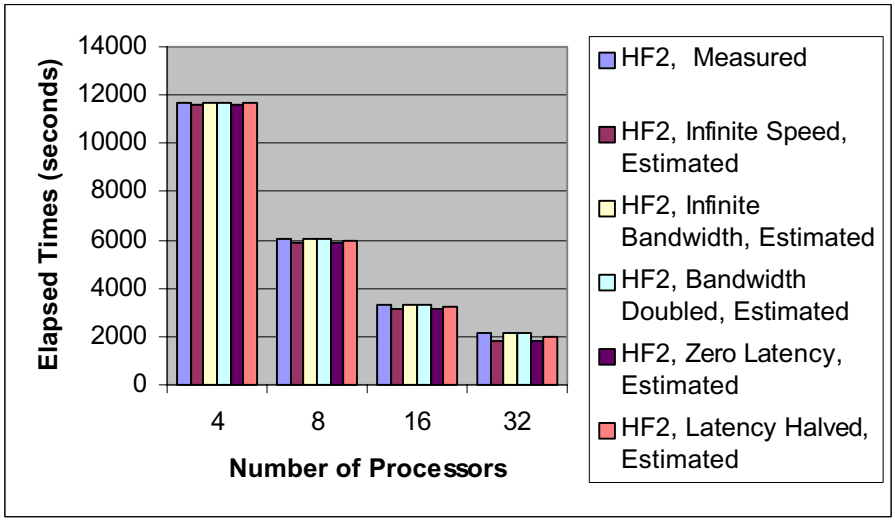

Figure 4. Graph for Table 4

Clearly, it shows that increasing the bandwidth of an interconnect has virtually no effect on the performance of MPP LS-DYNA, but decreasing the latency is effective in improving its performance. This is consistent with the observation that messages in the DYNA jobs are mostly small. Furthermore, the elapsed time of the 32-processor, double-precision job, with an interconnect of infinite speed, is calculated to be about 1/23th of the 1-processor job. So, for the Neon model with the default decomposition, the upper limit of speedup is about 23.

LOSS OF ACCURACY DUE TO SINGLE-PRECISION ARITHMETIC-- WHY 64-BIT COMPUTING?

Accuracy of MPP LS-DYNA

The aforementioned approach involved the use of both single-precision and double-precision MPP LS-DYNA jobs. As we examine the results of those jobs, described in the section, entitled **MODEL, MACHINE, INTERCONNECTS, MEASURED DATA**, we have found that results from single-precision jobs are not consistent. As the accuracy and consistency of jobs are very important to LS-DYNA users, this finding is presented here. Table 5 and Figures 5 and 6 depict that the total mass and the mass center, obtained from single-precision jobs, varies as the number of processors varies from 1 to 32. In contrast, the two quantities remain the same for double-precision jobs. Since the laws of conservation of mass and conservation of momentum dictate that the total mass and the mass center should remain the same under any deformations, this result shows losses of accuracy in single-precision MPP LS-DYNA simulations. The remedy for this loss of accuracy requires the use of double-precision MPP LS-DYNA.

Advantages for 64-bit Machines over 32-bit Machines

Traditionally, the main obstacle for MPP LS-DYNA users to adopt the double-precision simulation has been its relative cost to single precision: For example, it has been observed that, with the Neon model and with a cluster of 32-bit IA32 processors, elapsed times of double-precision jobs nearly *triples* those of single

precision jobs. In contrast, elapsed times, with the 64-bit Itanium machine, HP's rx2600, increase only by 20 percent, relative to those of single-precision jobs, as shown previously in Table 2. The 64-bit Itanium architecture offers not only higher performance in double-precision simulation but also a virtually limitless addressing space: A 64-bit machine offers addressing space up to 8 quintillion (10^{18}) bytes, in contrast to 2 gigabytes (10^{9}) bytes offered by a 32-bit machine.

No. of processors, Precision	Total Mass	X-Mass Center	Y-Mass Center	Z-Mass Center
1-32, Double	6.7645207E+01	5.2749981E+03	1.0303143E-01	8.3909895E+02
1, Single	6.7645180E+01	5.2588257E+03	1.0304040E-01	8.3548993E+02
2, Single	6.7645195E+01	5.2635117E+03	1.0304271E-01	8.3745575E+02
4, Single	6.7645233E+01	5.2730068E+03	1.0295519E-01	8.3794165E+02
8, Single	6.7645226E+01	5.2739995E+03	1.0299297E-01	8.3851489E+02
16, Single	6.7645241E+01	5.2746655E+03	1.0301991E-01	8.3878705E+02
32, Single	6.7645164E+01	5.2744912E+03	1.0302909E-01	8.3897211E+02

Table 5. Variation of the total mass and variation of X-coordinate, Y-coordinate, and Z-coordinate of the mass center for single-precision jobs with the number of processors varying from 1 to 32

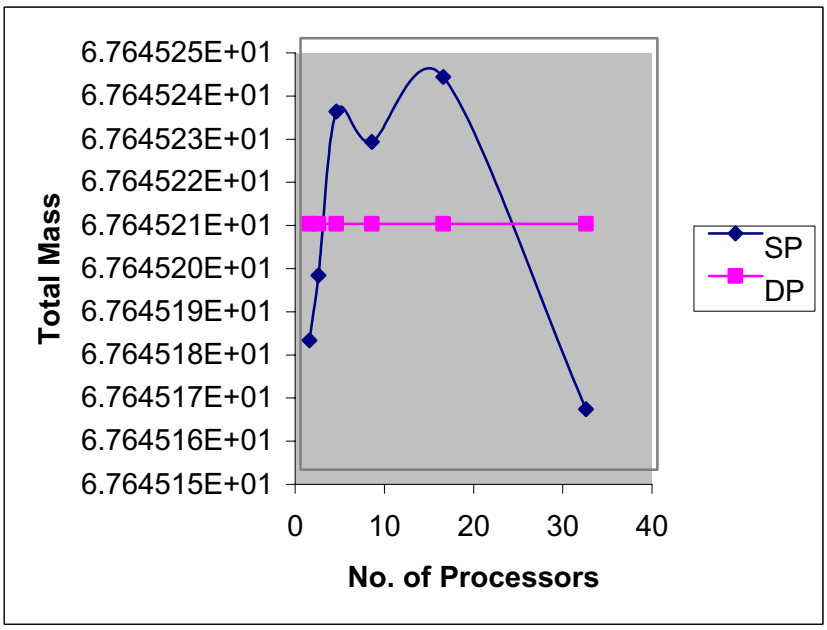

Figure 6. Graph for variation in the total mass as in Table 5

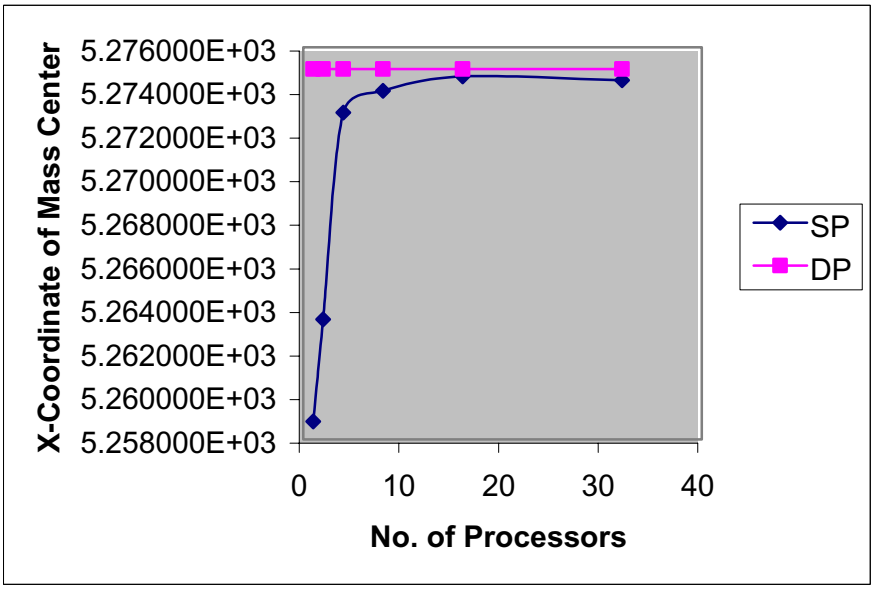

Figure 7. Graph for variation in the X-coordinate of the mass center in Table 5

Currently, the prevailing model size in crash simulation is about 0.5 million elements. A model of such size requires about 0.5 gigabytes of memory for the single-precision LS-DYNA and 1.0 gigabytes of memory for the double-precision. As the memory requirement goes roughly with the square of number of elements, should a user want to perform a crash simulation of 1 million elements, he has to use 64-bit machines.

SUMMARY AND CONCLUSIONS

In this paper, a quantitative approach to estimate the communication and the computation costs of an MPP LS-DYNA simulation is presented. The knowledge of the two costs will provide the MPP LS-DYNA user, the software developer and the hardware designer a deep insight into factors that affect the performance of MPP LS-DYNA. Additionally, the finding that there is loss of accuracy in single-precision MPP LS-DYNA simulations is presented.

REFERENCES

1. Hennessy, J. L., Patterson, D. A., *Computer Architecture: A Quantitative Approach*, 2nd Edition, 1996, Morgan Kaufmann Publishers, Inc., pp. 565-571.