Task Migration in 3D Wormhole-Routed Mesh Multicomputers

Gwo-Jong Yu
Department of
Computer Science,
National Central
University,
Chung-Li, Taiwan, R.O.C.
yu@axp2.csie.ncu.edu.tw

Chih-Yung Chan
Department of
Information Science,
Tamsui Oxford
University College,
Taipei, Taiwan, R.O.C.
changcy@jupiter.touc.edu.tw

Tzung-Shi Chen
Department of
Information Management,
Chang Jung
University,
Tainan, Taiwan, R.O.C.
chents@mail.cju.edu.tw

Abstract

In a mesh multicomputer, to perform jobs needs to schedule submeshes according to some processor allocation scheme. In order to assign the incoming jobs, a task compaction scheme is needed to generate a larger contiguous free region. The overhead of compaction depends on the efficiency of the task migration scheme. In this paper, two simple task migration schemes are first proposed in 3D mesh multicomputers with supporting dimension-ordered wormhole routing in one port communication models. Then, a hybrid scheme which combine advantages of the two task migration schemes is discussed. Finally, we compare the performance of all of these proposed approachs.

Keywords: Dimension-ordered routing, mesh multicomputers, parallel processing, task migration, wormhole routing.

1 Introduction

In a mesh multicomputer, to perform a sequence of jobs needs to schedule submeshes according to some sepecific processor allocation strategy, where each job allocates processors in a submesh with appropriate size. Number of research aimed at processor allocation and job scheduling schemes in hypercubes [3] [4] and mesh multicomputers[2] [5] [6] [9] [11].

Given a job, the mesh system will first allocate processors, then execute job, and finally free processors. After a lot of allocation and deallocation process, the mesh system became fragment. In such condition, a new job can not be scheduled to execute on this mesh due to lack of large enough contiguous area of processors, even if the number of free processors is sufficient. So, a task migration which moves running jobs from source processors to target processors is needed to increase the system utilization. Efficiency of task compaction is measured by the transmission latency of task migration. A lot of researchers design fast task migration schemes in hypercube by exploring disjoint paths between two subcubes. In hypercube, the larger the degree of the node the more the parallel pathes exist. However, the degree of each node is fixed to be 6 in 3D mesh system. In most case, it is impossible to find out the parallel paths between two submeshes. The objective of this paper is to minimize the tramission latency of task migration by means of transferring the job in a way of several phases.

In recent parallel computer machines, wormhole routing is the most important switching technique. In this paper, we use 3D mesh multicomputers supporting one port communication with dimension-ordered wormhole routing as target machines. We first present two task migration schemes in 3D mesh multicomputers. Then a hybrid task migration scheme is presented. Finally, we use performance analysis to compare all of our proposed task migration schemes.

In the next section, we first introduce the system model of mesh multicomputer and describe the problem to be solved. In Section 3, two task migration schemes are presented in 3D mesh multicomputers, and then a hybrid approach which integrates these two schemes is proposed. Performance analysis of various cases are demonstrated in Section 4. Section 5 is the conclusion.

2 Preliminaries

In this section, we introduce our system model for designing the task migration schemes on a 3-dimensional mesh multicomputer. The mesh multicomputer system is composed of nodes, each node is a computer with its own processor, local memory, and communication link, each directed link connects to two neighboring nodes through network [7] [8]. A common component of nodes in a new generation multicomputer is a router. It can handle and control the message communication entering, leaving, and passing through the node. The architecture of the $3D$ mesh network system used in this paper is to provide dimension-ordered wormhole routing with one-port communication, in which one node can only send and, simultaneously, receive a worm from the other respective nodes at the same time. The dimension-ordered routing used in this paper is assumed to route messages to destination nodes first along X-direction, then Y-direction, and then Z-direction on the mesh.

A three-dimensional mesh system $M(D, W, H)$ consists of $N = D \times W \times H$ number of processors arranged in an $D \times W \times H$ three-dimensional grid, where D, W, H represent as the depth, width, and height respectively. A processor in the grid is denoted by the coordinate (x, y, z). Let $M(D, W, H)$ denote the set of processors $\{(0,0,0), \ldots, (D-1, W-1, H-1)\}$. We define the submesh $SM(d, w, h)$ with depth d, width w, and height h in the mesh $M(D, W, H)$ that is a rectangular grid of processors embedded into $M(D, W, H)$. $SM(d, w, h)$ represents $\{(x_1, y_1, z_1), \ldots, (x_2, y_2, z_2)\}$, where (x_1, y_1, z_1) and (x_2, y_2, z_2) are the coordinates of the bottom-left-front and top-right-back corners respectively, where $d = x_2 - x_1 + 1$, $w = y_2 - y_1 + 1$, and $h = z_2 - z_1 + 1$. Here used in this paper we assume that the source submesh is $SM(d, w, h) = \{(x_1, y_1, z_1), \ldots, (x_2, y_2, z_2)\}$ and the destination submesh is $SM'(d, w, h) = \{(x_3, y_3, z_3), \ldots, (x_4, y_4, z_4)\}$. These two submeshes with both of the same shape and size are located in different locations with allowing them to be partially overlapped. One node (x, y, z) in $SM(d, w, h)$ is needed to route its assigned subtask to the corresponding destination node $(x\prime, y\prime, z\prime)$ in $SM'(d, w, h)$, where

$$\begin{cases} x' = x_3 + (x - x_1) \\ y' = y_3 + (y - y_1) \\ z' = z_3 + (z - z_1) \end{cases} \quad (1)$$

3 Task Migration

In this section, two task migration schemes are first proposed in $3D$ mesh multicomputers based on dimension-ordered wormhole routing. Then, we propose a hybrid scheme in order to minimize the total routing latency.

3.1 Diagonal Scheme

In this subsection, a task migration approach which explores disjoint paths in mesh multicomputer with dimension-ordered wormhole routing is proposed. First, we use the following example to illustrate the main idea of the developed task migration scheme. A task with several subtasks allocated to a $5 \times 4 \times 3$ submesh is needed to be migrated to another $5 \times 4 \times 3$ submesh on a mesh $\{(0,0,0), \ldots, (11,11,11)\}$ as depicted in Figure 1. The task in the source submesh $\{(1,1,1), \ldots, (5,4,3)\}$ is needed to be migrated to the destination submesh $\{(6,7,8), \ldots, (10,10,10)\}$ in a $12 \times 12 \times 12$ mesh systems. We use a scheme to migrate the tasks distributed in nodes located in various X, Y, and Z axis, labeled by phases 1, 2, 3, 4, and 5 to the corresponding destination nodes. As shown in Figure 2, 12 nodes migrate their subtasks to corresponding destinations respectively in parallel. Note that each node migrate its subtasks first along X-direction, then Y-direction, and finally Z-direction. We can check out no common links are used or shared in each phases. For example, in Phase 1 if we project those selected nodes to the Y-Z plane, then all nodes will fall in different position. We can conclude that no two nodes will share or use the same link along X-direction. Similarly, it is easy to verify that no two nodes will share or use the same link along Y-direction and Z-direction too. Therefore, in each phase, the routing is congestion-free based on dimension ordered routing in one-port communication, in which one node can simultaneously send out and receive from one worm at a time. There are totally five phases needed to complete the task migration tasks in this example.

Next, we generalize the above descriptions of the migration routing scheme in more detail as follows. In order to avoid congestion during performing dimension-ordered routing, the nodes in the source submesh needed to route their subtasks to the corresponding nodes are in different rows and columns when we project on XY-plane, YZ-plane, and XZ-plane respectively. Therefore, we

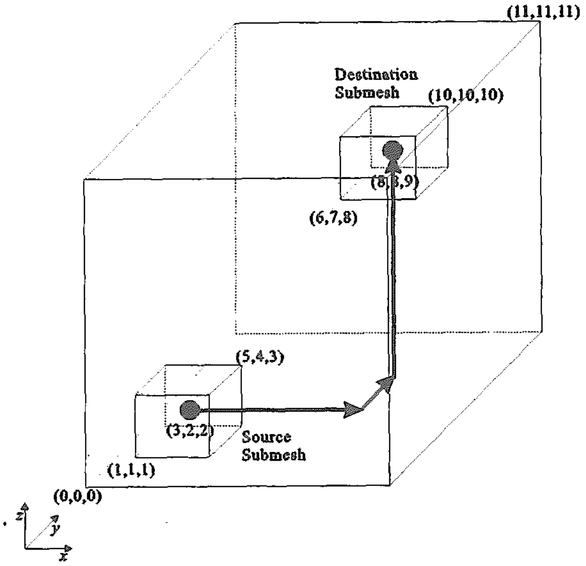

Figure 1: Task migration between two $5 \times 4 \times 3$ submeshes in a 3D mesh with dimension-ordered wormhole routing.

take into account using the diagonal scheme to migrate these subtasks. Here we only state how to initialize the task migration from the source submesh because it is easy for each pair of nodes (x, y, z) and (x', y', z') to identify the routing from source to destination.

Here the source submesh $SM(d, w, h)$ is assumed to be $\{(x_1, y_1, z_1), \ldots, (x_2, y_2, z_2)\}$ where $d = x_2 - x_1 + 1$, $w = y_2 - y_1 + 1$, and $h = z_2 - z_1 + 1$. We state how to arrange the routing phases P_k for $1 \leq k \leq \max(d, w, h)$. The routing during migration is one-by-one from phase P_1 to $P_{max(d,w,h)}$. For each phase P_k, all of nodes (x, y, z), which are lying on planes

$$x + y + z = c \cdot \max(d, w, h) + k, \qquad (2)$$

are scheduled to simultaneously migrate their subtasks to their corresponding destinations, where c is intergers, $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$, $z_1 \leq z \leq z_2$, and $1 \leq k \leq \max(d, w, h)$. Clearly, we have a lot of parallel planes $x + y + z = H'$, where $(x_1 + y_1 + z_1) \leq H' \leq (x_2 + y_2 + z_2)$. From the above description, the entire task distributed to all nodes in the source submesh is migrated to destination submesh in $\max(d, w, h)$ phases. We prove that the routing is congestion-free in each phase below. In addition, we prove that the number of routing phases is minimum with congestion-free routing in one-port communication model.

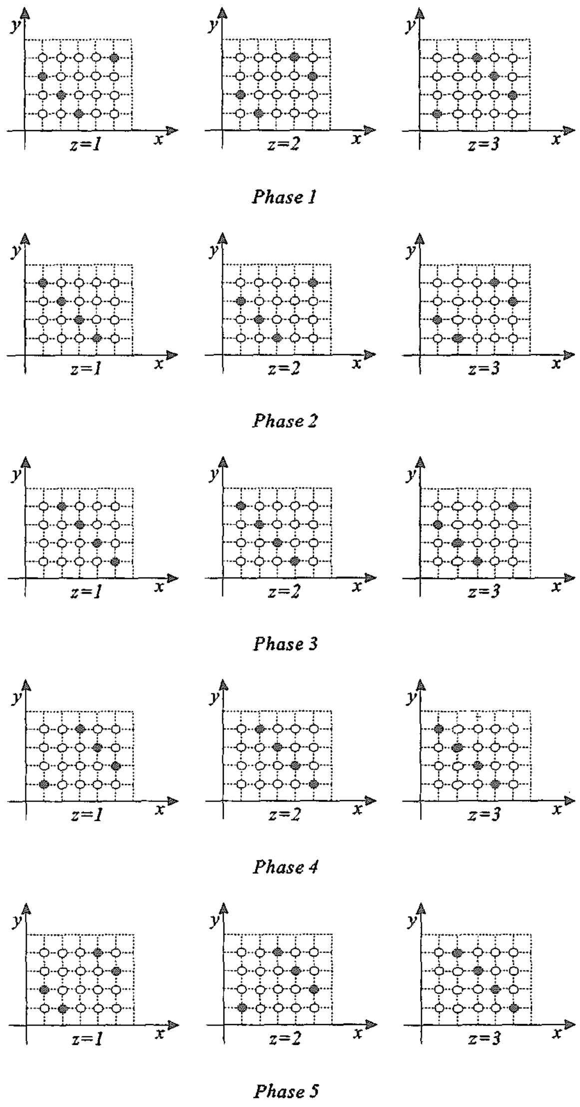

Figure 2: Schedule of each phase in task migration process using the proposed diagonal scheme. Those circles filled with black indicates processors which can be migrated to destination position with congestion-free manner in each phase.

Theorem 1 *In the diagonal scheme, the routing in each phase is congestion free.*

Proof: We will show that in each phase, there is exactly one node occupying the X-direction communication channels. By projecting nodes that are scheduled in the same phase to plane $x = 0$, all nodes will have different locations, so we have that all these nodes are congestion free along X-direction. We prove these arguments in what follows. We have planes $x + y + z = H$ for $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$, and $z_1 \leq z \leq z_2$, where $x_1 + y_1 + z_1 \leq H \leq x_2 + y_2 + z_2$. In phase P_k, nodes in plane

$$x + y + z = c \cdot \max(d, w, h) + k, \tag{3}$$

are scheduled to migrate subtasks to destination. Assume (x^*, y^*, z^*) is an node on the plane in Equation (3). Projecting (x^*, y^*, z^*) onto the plane $x = 0$ generates an corresponding position $(0, y^*, z^*)$. If there are more than one nodes projecting onto plane $x = 0$ generates the same location $(0, y^*, z^*)$, then these nodes will cause congestion when applying X-direction communication channels. So, we have to prove that there are exactly one nodes in phase P_k project onto the specific position $(0, x^*, y^*)$. Assume nodes (x, y^*, z^*) project onto $(0, x^*, y^*)$, then

$$x + y^* + z^* = c \cdot \max(d, w, h) + k$$

After some manipulation, we observed that

$$x = c \cdot \max(d, w, h) + (k - y^* - z^*).$$

Because $x_1 \leq x \leq x_2$, the following inequation holds:

$$x_1 \leq c \cdot \max(d, w, h) + (k - y^* - z^*) \leq x_2 \tag{4}$$

$$\Rightarrow 0 \leq c{\cdot}\max(d, w, h) + (k - y^* - z^* - x_1) \leq (x_2 - x_1). \tag{5}$$

$$\Rightarrow 0 \leq c{\cdot}\max(d, w, h) + (k - y^* - z^* - x_1) \leq \max(d, w, h). \tag{6}$$

Because $k - y^* - z^* - x_1$ is constant, there will be only one c satisfies the Inequation (5). So, for fixed y^* and z^*, there will be only one x^* fulfill equation (3). If the projected positions of all of these nodes are all different, then we can conclude that all of these nodes are congestion free along X-direction. The same arguments can be applied to Y and Z direction. If all of three direciton are congestion free, then we conclude that the migration process is congestion free.

Theorem 2 *The number of phases,* $\max(d, w, h)$, *with congestion-free routing is minimum based on dimension-ordered routing in one port communication model.*

Proof: Based on dimension-ordered routing, there are at most $w \times h$ nodes delivering data along X-direction simultaneously, at most $d \times h$ nodes delivering data along Y-direction simultaneously, and at most $w \times d$ along Z-direction simultaneously. For a submesh $SM(d, w, h)$, therefore, there are at most $\min(w \times d, d \times h, w \times d)$ nodes being simultaneously able to route their data in a way of congestion-free transmission. It is limited by boundaries of $SM(d, w, h)$; that is, the limitation is on the minimum value of depth, width, and height. Therefore, the minimum number of routing phase are $\frac{d \times w \times h}{\min\{w \times h, d \times h, w \times d\}}$, where $d \times w \times h$ is the total number of nodes on the submesh $SM(d, w, h)$. According to Theorem 1, each phase is congestion-free. We need exactly $\max(d, w, h)$ phases. Thus, the minimum number of routing phase, $\frac{d \times w \times h}{\min(w \times h, d \times h, d \times w)} = \max(w, h, d)$, whose value is minimum.

$\perp$

3.2 Gathering-Routing-Scattering Scheme

We next describe our second task migration scheme based on two collective communication schemes, gathering and scattering operations[7], as follows. Without loss of generality, we assume the source submesh has maximum length in X-direction. We use the gathering operation on nodes in the same X-direction to collect each subtasks of each node into one node. After the node migrates combined subtasks to the corresponding destination node, we use the scattering operation on nodes in a line along X-direction to disperse a couple of subtasks assigned to the destination node, to their respective nodes in the destination submesh. The gathering and scattering schemes can be referred to [7] and [11]. It is noted that the node in the end of line along X-direction is received the total amount of subtasks while we perform the gathering. It is necessary for routing to the node located at the designed position of specific phase to take an additional routing step in the source submesh. The scattering operation is also needed an additional routing step to route the data from one designed node located on the position of specific phase, to the node in the end of a line along X-direction in the destination submesh. If the source mesh has maximum length along Y or Z-direction, the above description holds.

Without loss of generality, assume the submesh has maximum length in X-direction. All of nodes (x, y, z) located on planes $x+y+z = c \cdot \max(d, w, h)$ are gathering subtasks distributed within all of nodes in X-direction of (x, y, z). The node (x, y, z) next migrates the combined subtasks to the corresponding node (x', y', z'). The node (x', y', z') finally scatters the combined subtask to the respective destination nodes on lines along X-direction to complete the task migration. Similar arguments holds when the maximum length is with Y-direction or Z-direction communication channels.

Theorem 3 *The gathering-routing-scattering scheme is congestion-free in each phase.*

Proof: In this approach, all of the gathering and scattering steps are congestion-free[7][11]. The middle step, the nodes scheduled to combine subtasks to the corresponding destination nodes, is alos congestion-free, which was shown in Theorem 1. Thus, the gathering-routing-scattering scheme is congestion free in each phase.

$\perp$

3.3 Hybrid Scheme

In this subsection, we present a hybrid task migration scheme on the basis of dimension-ordered wormhole routing. We combine two schemes stated in the previous subsections to a hybrid one. We first partition the source submesh $SM(d, w, h)$ into several cuboid subpartitions, which is also of submesh form with the size of $p \times q \times r$. That is the number of subpartitions is $\lceil \frac{d}{p} \rceil \times \lceil \frac{w}{q} \rceil \times \lceil \frac{h}{r} \rceil$, where $\lceil \frac{d}{p} \rceil$ is the depth, $\lceil \frac{w}{q} \rceil$ is the width, and $\lceil \frac{h}{r} \rceil$ is the height of the partitoned submesh. In the following, we state how to use the combined scheme to perform task migration. We first partition the source submesh into subpartitions. We then use the gathering scheme to collect subtasks into nodes, located at the designated positions, depending on the size of d, w, and h. This step is similar to the gathering-routing-scattering scheme we proposed. After that, we use the diagonal scheme, scheduled with $\max(\lceil \frac{d}{p} \rceil, \lceil \frac{w}{q} \rceil, \lceil \frac{h}{r} \rceil)$ phases, to route the aggregated subtasks to their corresponding nodes. In this step, each subpartitions is represented as a supernode to transfer their subtasks on each node to its corresponding destination subpartition. Finally, we use the scattering scheme to distribute subtasks on the direction which is with the maximum length in each partition to complete the task migration.

Theorem 4 *The hybrid scheme is congestion-free in each phase.*

Proof: The gathering scheme is first used to collect subtasks in each subpartition on X, Y, or Z-direction; hence, it is congestion-free. It is also congestion-free that each node scheduled in the same phase in each subpartition migrates its combined subtasks to its corresponding node. By Theorem 1 and 3, we know that this step is also congestion-free. The next step is to route the combined subtasks, scheduled with $\max(\lceil \frac{d}{p} \rceil, \lceil \frac{w}{q} \rceil, \lceil \frac{h}{r} \rceil)$ phases, ato their corresponding nodes. Finally, the scattering scheme is used to distribute each subtask in each subpartition on X, Y, or Z direction; hence, it is congestion-free. Therefore, the proposed hybrid scheme is congestion-free in each phase.

$\perp$

4 Performance Analysis

In this section, we will compare our developed schemes via performance analysis. At the beginning, some parameters measured and used in our proposed schemes are first described in below. Here assume that the startup latency is t_s and the transmission time of a flit (or byte) is t_x on a link between two neighboring nodes. We also assume that the size of subtask distributed in each node is the same, m flits for analyzed convenience. In general, in the wormhole routing model, the latency, a node sends one message with m flits to another node, is $t_s + t_x \times m$[7] [8].

Because the hybrid-scheme is combination of the diagonal scheme and the gathering-routing-scattering scheme, we first discuss the total transmission latency related to the hybrid task migration scheme in below. The hybrid task migration needs three steps, gathering, diagonal routing, and scattering steps. In the gathering, it takes $\max(\lceil \log p \rceil + 1, \lceil \log q \rceil + 1, \lceil \log r \rceil + 1)$ time steps to collect these subtasks into one node located at the designated lines of nodes in each subpartitions.

The combined message size is two times of that of previous one and we need one extra step to transmit the final combined subtasks to the node located at the position in the designated phase. Thus, the final total size of the collected message is $\max(m2^{\lceil \log_2 p \rceil}, m2^{\lceil \log_2 q \rceil}, m2^{\lceil \log_2 r \rceil})$. This concludes the gathering step takes the time in below which can be obtained by the above derivations.

$$\begin{aligned} T_{gather} &= \max(\lceil \log p \rceil + 1, \lceil \log q \rceil + 1, \lceil \log r \rceil + 1) \times t_s \\ &+ \max(m2^{\lceil \log_2 p \rceil}, m2^{\lceil \log_2 q \rceil}, m2^{\lceil \log_2 r \rceil}) \times t_x. \end{aligned}$$

In the diagonal routing step, we need $\max(\lceil \frac{d}{p} \rceil, \lceil \frac{w}{q} \rceil, \lceil \frac{h}{r} \rceil)$ phases to migrate these combined subtasks in this step by our proposed diagonal scheme. Thus, the diagonal routing is to take the time in below.

$$T_{diagonal} = \max(\left\lceil \frac{d}{p} \right\rceil, \left\lceil \frac{w}{q} \right\rceil, \left\lceil \frac{h}{r} \right\rceil) \times (t_s + \max(mp, mq, mr) \times t_x)$$

Finally, the scattering step takes the same time as the gathering step in below.

$$\begin{aligned} T_{scatter} &= \max(\lceil \log p \rceil + 1, \lceil \log q \rceil + 1, \lceil \log r \rceil + 1) \times t_s \\ &+ \max(m2^{\lceil \log_2 p \rceil}, m2^{\lceil \log_2 q \rceil}, m2^{\lceil \log_2 r \rceil}) \times t_x. \end{aligned}$$

Thus, we have the total transmission time

$$T_{hybrid} = T_{gather} + T_{diagonal} + T_{scatter}.$$

We know that the first two schemes proposed in the previous section is the special case of the hybrid scheme. The first task migration scheme, diagonal scheme takes the transmission time in below not containing the gather and scatter steps; that is, $p = 1$,$q = 1$, and $r = 1$.

$$T_1 = \max(d, w, h) \times (t_s + m \times t_x)$$

The second task migration scheme, gathering-routing-scattering scheme, is to take the transmission time in below; that is, $p = d$, $q = w$, and $r = h$.

$$\begin{aligned} T_2 &= 2 \times (\max(\lceil \log d \rceil + 1, \lceil \log w \rceil + 1, \lceil \log h \rceil + 1) \times t_s \\ &+ \max(m2^{\lceil log_2 d \rceil}, m2^{\lceil log_2 w \rceil}, m2^{\lceil log_2 h \rceil}) \times t_x) \\ &+ (t_s + \max(md, mw, mh) \times t_x). \end{aligned}$$

From the above analysis of time complexity, we know that the amount of startup latencies, $\max(d, w, h)$, in the diagonal scheme is larger than that of $2 \times (\max(\lceil \log d \rceil + 1, \lceil \log w \rceil + 1, \lceil \log h \rceil + 1) + 1)$ in the gathering-routing-scattering scheme in general. That is the reason why we use the collective communication to reduce the total amount of startup latency in migrating a task. However, the transmission size of the message between two submeshes in the diagonal scheme is generally smaller than that in the gathering-routing-scattering scheme. The hybrid scheme, therefore, is proposed for optimizing the time of migrating one task. In order to minimize the total amount of transmission delay T_{hybrid}, it is necessary for us to derive the optimum partitioning with respect to the values of p and q for a specific kind of mesh architecture.

Here we give some assumptions to the parameters of system architecture for the use of analyzing the routing performance. This analysis is formed on a $64 \times 64 \times 64$ submesh accommodating a job needed to be migrated to another location. We have the message startup latency t_s is 1.0 microsecond for the small startup latency and t_s is 10.0 microseconds for the large startup latency. The transmission time of a flit t_x on a link is 20.0 nanoseconds. The amount of a task in one node is assumed to be m flits; here we have the different values of 100, 300, and 600 flits to be discussed.

We discuss the impact on the transmission latency when using different partitioning sizes, i.e., we have different values of $p \times q \times r$. In order to balance, the routing phase and the subpartitions are with the shapes of the size of $1 \times 1 \times 1$, $2 \times 2 \times 2$, $4 \times 4 \times 4$, $8 \times 8 \times 8$, $16 \times 16 \times 16$, $32 \times 32 \times 32$ and $64 \times 64 \times 64$ used in our performance analysis. In the case of $1 \times 1 \times 1$, that is to the diagonal scheme. In the case of $64 \times 64 \times 64$ that is to the gathering-routing-scattering scheme. For the small startup latency, the incurred transmission latency are shown in Fig. 3. For large startup latency, the incurred transmission latency are shown in Fig. 4. As shown in Fig. 3, the minimum transmission latency occurred in the partition size $8 \times 8 \times 8$ in all three messages size. This means that the hybrid approach has better performance than diagonal scheme or gathering-routing-scattering scheme only. We also note that in small startup latency, the diagonal scheme ($1 \times 1 \times 1$) has smaller transmission latency than gathering-routing-scattering scheme ($64 \times 64 \times 64$) in large message size (600). On the other hand, when the message size is small, the gathering-routing-scattering scheme has smaller transmission latency than diagonal scheme. In the case of large startup latency, Figure 4, hybrid scheme with subparti-

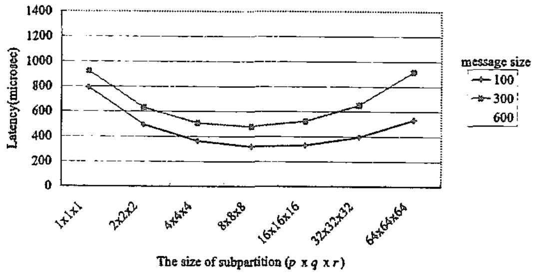

Figure 3: Transmission Latency for small startup latency in 3D mesh with various partition size.

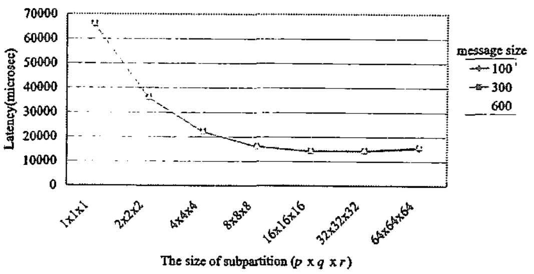

Figure 4: Transmission Latency for large startup latency in 3D mesh with various partition size

tion size $16 \times 16 \times 16$ has minimin transmission latency. Also, in all message size, the gather-routing-scattering scheme is better than diagonal scheme. This is because in large startup latency, the gather-routing-scattering has better efficency than diagonal scheme.

Briefly, from the above analysis, we have the following comments and suggestions as performing task migration. Generally speaking, we use the first scheme, the diagonal scheme, to perform task migration to gain the minimized transmission latency as the startup latency is smaller. It is easily to find out the time as using the second scheme is usually the most larger that as using the others as show in Figure 6. Thus, the second one is more unsuitable for performing task migration. The reason is that the collective communication used in each X, Y, or Z-direction is to take a lot of time so as to collect a large amount of messages. However, we have to take into account the factors of affecting the transmission latency, including the message size assigned in each node, the startup latency, as well as the partitioning size and shape together as the hybrid scheme is used. By evaluating these factors together, we are able to get the optimum solution with having the minimum time of T_{hrbrid}.

5 Conclusion

In this paper, two simple task migration schemes are proposed in 3D mesh multicomputers with supporting dimension-ordered worm-hole routing in one-port communication model. Furthermore, a hybrid task migration scheme was proposed with the attempt to minimizing the routing latency. Finally, we compare all of our proposed task migrations schemes via performance analysis. In addition, we discuss how to easily apply our proposed task allocation schemes to some different processor allocation schemes with contiguous methods. We will exploit and investigate the task migration schemes on higher dimensional mesh multicomputers in the future. The other research work is on investigating the job scheduling approaches, integrating task migration and processor allocation together, and evaluating the entire system performance including system utilization, job response time and more.

References

[1] G.-I. Chen and T.-H. Lai, "Constructing parallel paths between two subcubes," *IEEE Transactions on Computers*, vol. 41, pp. 118–123, January 1992.

[2] P.-J.Chuang and N.-F. Tzeng, "An efficient submesh allocation strategy for mesh computer systems," *Proceedings of 11th International Conference on Distributed Computing Systems*, pp. 256–262, May 1991.

[3] P.-J. Chuang and N.-F. Tzeng, "A fast recognition-complete processor allocation strategy for hypercube computers," *IEEE Transactions on Computers*, vol. 41, no. 4, pp. 467–479, 1992.

[4] J. Kim, C. R. Das, and W. Lin, "A top-down processor allocation scheme for hypercube computers," *IEEE Transaction on Parallel and Distrubuted Systems*, vol. 2, pp. 20–30, January 1991.

[5] K. Li and K. H. Cheng, "A two-dimensional buddy system for dynbamic resource allocation in a partitionable mesh connected system," *Journal of Parallel and Distrubuted Computing*, vol. 31, pp. 79–83, December 1991.

[6] V. Lo, K. J. Windisch, W. Liu, and B. Nitzberg, "Noncontiguous processor allocation algorithms for mesh-connected multicomputers," *IEEE Transaction on Parallel and Distributed Systems*, vol. 8, pp. 712–726, July 1997.

[7] P. K. Mckinley, Y.-J. Tsai, and D. F. Robinson, "Collective communication in wormhole-routed massively parallel computers," *IEEE Computers*, vol. 28, pp. 39–50, December 1995.

[8] L. M. Ni and P. K. Mckinley, "A survey of wormhole routing techniques in direct networks," *IEEE Computer*, vol. 26, pp. 62–76, Febeuary 1993.

[9] D. D. Sharma and D. K. Pradhan, "Submesh allocation in mesh multicomputers using busy-list: A best-fit approach with complete recognition capability," *Journal of Parallen and Distributed Systems*, vol. 36, pp. 106–118, January 1998.

[10] D. D. Sharma and D. K. Pradhan, "Job scheduling in mesh multicomputers," *IEEE Transactions on Parallel and Distrubuted Systems*, vol. 9, pp. 57–70, January 1998.

[11] Y.-C. Tseng, S.-Y. Ni, and J.-P. Sheu, "Toward optimal complete exchange on wormhole-routed tori," in *Proceedings of the 1997 International Conference on Parallel and Distributed Systems*, (Seoul, Korea), pp. 96–103, December 1997.

[12] N.-F. Tzeng and H.-L. Chen, "Fast compaction in hypercubes," *IEEE Transactions on Parallel and Distrubuted Systems*, vol. 9, pp. 50–56, January 1998.