Task Migration in 2D Wormhole-Routed Mesh Multicomputers

Tzung-Shi Chen

Department of Information Management
Chang Jung University
Tainan 711, Taiwan, R.O.C.
chents@mail.cju.edu.tw

Abstract. In this paper, two simple task migration schemes are first proposed for 2D mesh multicomputers with supporting X-Y wormhole routing in the one-port communication model. We next propose a hybrid task migration scheme which attempts to minimize the total transmission latency. Finally, we compare all of our proposed task migration schemes using performance analysis.

1 Introduction

Submeshes in a mesh multicomputer are scheduled to perform a sequence of jobs (tasks) according to a processor allocation strategy, with each job assigned to occupy processors of one submesh with an appropriate size. Over the past decade, numerous researchers have focused their attention on designing processor allocation or job scheduling schemes in hypercubes [3] and mesh multicomputers [2] [5].

When the execution of a job is completed, the occupied submesh should be released and deallocated for use in other jobs. However, after repeated allocation and deallocation, it is possible for the mesh system to become fragmented. When this occurs, an incoming job can not be assigned even if a sufficient number of free processors are present because there is no submesh with an appropriate size available to accommodate that job. Therefore, it might be advantageous for a highly fragmented mesh system to perform task compaction, reallocating some of the active jobs currently executed to a specified area, thus obtaining larger contiguous regions for upcoming jobs.

Efficiency of task compaction mainly depends on minimizing the transmission time of migrating tasks. A great deal of research has been aimed at developing and designing fast task migration schemes to reduce the transmission time in hypercubes by exploring disjoint paths between two subcubes, source and destination subcubes [1] [7]. This is because the greater degree to which nodes increased, the more parallel paths between subcubes there are. We know that the degree of each node is fixed, except for nodes on boundaries in mesh networks, such as 2D mesh networks, where the maximum number of degree is equal to 4. Hence, in most of cases, it is impossible to migrate a job to determine the parallel paths between two submeshes.
One of the most important switching techniques, wormhole routing, is widely used on modern parallel computer machines [4]. We intend to use 2D mesh multicomputers supporting one-port communication with X-Y wormhole routing (also called dimension-ordered routing) [4] as the target machines in this paper. Two simple task migration schemes are first presented for 2D mesh multicomputers. We next propose a hybrid task migration scheme. Finally, we compare all of our proposed task migration schemes using performance analysis.

The rest of this paper is organized as follows. Section 2 introduces the system model of mesh multicomputers and states the problem statement we intend to solve. In Section 3, we first present two simple task migration schemes for 2D mesh multicomputers. Next, we propose the hybrid task migration scheme. Performance comparisons among these schemes are analyzed and discussed in Section 4. Conclusions are summarized in Section 5.

2 Preliminaries

The architecture of the 2D mesh network system used in this paper provides X-Y wormhole routing with one-port communication [4], which allows one node to send one worm (or message) to and simultaneously receive another worm from two respective nodes at the same time.

A two-dimensional mesh system, \( M(H, W) \), consists of \( N = H \times W \) number of processors arranged in an \( H \times W \) two-dimensional grid, where \( H \) represents the height and \( W \) represents the width. A processor in the grid is denoted by the coordinate \((x, y)\). \( M(H, W) \) is defined as \((0, 0), (H-1, W-1)\). We define the submesh \( SM(h, w) \) with height \( h \) and width \( w \) within the mesh \( M(H, W) \) as a rectangular grid of processors embedded into \( M(H, W) \). \( SM(h, w) \) is represented by \( \{(x_1, y_1), (x_2, y_2)\} \), where \((x_1, y_1)\) and \((x_2, y_2)\) are respectively the coordinates of the bottom-left and top-right corners, where \( h = y_2 - y_1 + 1 \) and \( w = x_2 - x_1 + 1 \). In this paper we assume that the source submesh is \( SM(h, w) = \{(x_1, y_1), (x_2, y_2)\} \) and the destination submesh is \( SM'(h, w) = \{(x_3, y_3), (x_4, y_4)\} \). These two submeshes with the same shape and size are located in different locations, although they can partially overlap. One node \((x, y)\) in \( SM(h, w) \) is needed to route its assigned subtask to the corresponding destination node \((x', y')\) in \( SM'(h, w) \), where \( x' = x_3 + (x - x_1) \) and \( y' = y_3 + (y - y_1) \).

3 Task Migration

3.1 Diagonal Scheme

In this subsection, our main goal is to explore all disjoint paths in one phase in order to migrate a task from a source submesh to the destination submesh based on X-Y routing in the one-port communication model. First, we demonstrate our basic idea of developing the task migration scheme using the following example. Here, a task with several subtasks allocated within a \( 3 \times 5 \) submesh needs to be migrated to another \( 3 \times 5 \) submesh on a mesh \( \{(0, 0), (11, 11)\} \), as depicted in...