

The Effect of Real Workloads and Synthetic Workloads on the Performance of Job Scheduling for Non-Contiguous Allocation in 2D Mesh Multicomputers

Saad Bani-Mohammad, Department of Computer Science, Prince Hussein Bin Abdullah College for Information Technology, Al al-Bayt University, Mafrqa, Jordan

ABSTRACT

The performance of non-contiguous allocation has been traditionally carried out by means of simulations based on synthetic workloads, and also it can be significantly affected by the job scheduling strategy used for determining the order in which jobs are selected for execution. To validate the performance of the non-contiguous allocation algorithms, there has been a need to evaluate the algorithm's performance based on a real workload trace. In this paper, the performance of the well-known Greedy Available Busy List (GABL) non-contiguous allocation strategy for 2D mesh-connected multicomputers is revisited considering several important job scheduling strategies based on a real workload trace, and the results are compared to those obtained from using a synthetic workload. The scheduling strategies used are the First-Come-First-Served (FCFS), Out-of-Order (OO), and Window-Based job scheduling strategies. These strategies have been selected because they are common and they have been used in related works (Ababneh & Bani-Mohammad, 2011). Extensive simulation results based on synthetic and real workload models indicate that the Window-Based job scheduling strategy can improve both overall system performance and fairness (i.e., maximum job waiting delays) by adopting a large job scheduling window. Moreover, the relative performance merits of the scheduling strategies when a real workload trace is used are in general compatible with those obtained when a synthetic workload is used.

Keywords: *Job Scheduling, Maximum Job Waiting Delay, Non-Contiguous Processor Allocation, System Utilization, Turnaround Time*

DOI: 10.4018/ijdst.2015010104

1. INTRODUCTION

Among the various multicomputer architectures, those based on the mesh network have received much attention due to the simplicity, structural regularity, partition-ability, and ease of implementation of this network (Yoo & Das, 2002; Seo & Kim, 2003; Bani-Mohammad, 2008; Ababneh, 2009; Bani-Mohammad et al., 2009; Ababneh & Bani-Mohammad, 2011). Meshes are suited to a variety of applications, including matrix computations, image processing and problems with task graphs that can be embedded naturally into the mesh (Varavithya, 1998).

Efficient processor allocation and job scheduling are critical if the full computational power of large-scale multicomputers is to be harnessed effectively (Chiu & Chen, 1999; Choo, Yoo & Youn, 2000; Yoo & Das, 2002; Bani-Mohammad et al., 2009). The goal of job scheduling is to select the next job to be executed while the goal of processor allocation is to select the set of processors on which parallel jobs are executed (Yoo & Das, 2002).

In most processor allocation strategies proposed for mesh-connected multicomputers, a parallel job is allocated its own sub-mesh of processors of the size and shape it has requested (Bhattacharya & Tsai, 1994; Sharma & Pradhan, 1996; Kim & Yoon, 1998; Ababneh, 2009; Bani-Mohammad et al., 2009). This can result in high external processor fragmentation, which occurs when free processors are not allocated to a parallel job because they do not satisfy the job's shape constraint. A recurring outcome in allocation studies is that contiguous allocation suffers from low overall system utilization (Lo et al., 1997; Choo, Yoo & Youn, 2000; Ababneh, 2009). It can reduce system utilization to low levels. Therefore, non-contiguous allocation strategies have been proposed so as to increase system utilization by allowing dispersed free processors to be allocated to a parallel job (Lo et al., 1997; Chang & Mohapatra, 1998; Bani-Mohammad, Ould-Khaoua, & Ababneh, 2007; Ababneh, 2008).

Another approach to improving system utilization is using job scheduling strategies that are not strictly first-come-first-served. Rather than always accommodating the oldest allocation request first, allocation to more recent requests is considered in order to decrease the number of idle processors and improve overall system performance (Mu'alem & Feitelson, 2001; Ababneh, 2001). These strategies, however, have several shortcomings. In (Ababneh, 2001), jobs are considered for allocation without ever waiting for the head of the queue or earlier requests. This out-of-order scheduling can lead to excessive waiting delays, including indefinite postponement, for large jobs. Because small jobs are easier to accommodate, they can block allocation to a large job that has arrived earlier. The scheduling strategy proposed in (Mu'alem & Feitelson, 2001) assumes that job execution time estimates are available upon job submission. Although many current systems require users to submit such estimates, users often provide inaccurate estimated job execution times (Feitelson, 2014). Therefore, we have limited ourselves to the case where execution time estimates are assumed to be unavailable. The scheduling strategy proposed in (Ababneh & Bani-Mohammad, 2011) aims to bound job waiting delays, while reducing processor fragmentation. In this strategy, it is possible to bypass the head of the waiting queue of jobs awaiting allocation, but this ability is limited to a window of consecutive jobs that starts with the head of the waiting queue. Limiting the ability to bypass the oldest waiting job to a window has for goal bounding job waiting delays, while bypassing the head of the waiting queue has for goal reducing processor fragmentation by finding jobs that can be accommodated, which can improve system utilization and average job turnaround times.

In this paper, we study, using detailed simulations, the effect of job scheduling on the performance of the Greedy Available Busy List (GABL) non-contiguous allocation strategy proposed in (Bani-Mohammad, Ould-Khaoua & Ababneh, 2007) based on both synthetic and

14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/the-effect-of-real-workloads-and-synthetic-workloads-on-the-performance-of-job-scheduling-for-non-contiguous-allocation-in-2d-mesh-multicomputers/120460

Related Content

Computer Network Vulnerability Detection and Semantic Data Analysis Optimization Based on Artificial Intelligence

Huiyan Liand Xinhua Xiao (2022). *International Journal of Distributed Systems and Technologies* (pp. 1-10).

www.irma-international.org/article/computer-network-vulnerability-detection-and-semantic-data-analysis-optimization-based-on-artificial-intelligence/308000

Face Detection-Induced Access Control System via Large Margin Metric Learning

Li'e Puand Jialin Sun (2022). *International Journal of Distributed Systems and Technologies* (pp. 1-11).

www.irma-international.org/article/face-detection-induced-access-control-system-via-large-margin-metric-learning/307987

Performance Evaluation of Full Diversity QOSTBC MIMO Systems with Multiple Receive Antenna

Hardip K. Shah, Tejal N. Parmar, Nikhil Kothariand K. S. Dasgupt (2011). *International Journal of Grid and High Performance Computing* (pp. 29-38).

www.irma-international.org/article/performance-evaluation-full-diversity-qostbc/60253

Intelligent Management of Mobile Systems Through Computational Self-Awareness

Bryan Donyanavard, Amir M. Rahmani, Axel Jantsch, Onur Mutluand Nikil Dutt (2021). *Handbook of Research on Methodologies and Applications of Supercomputing* (pp. 41-73).

www.irma-international.org/chapter/intelligent-management-of-mobile-systems-through-computational-self-awareness/273394

Implementing Geospatial Web Services for Cloud Computing

Gobe Hobona, Mike Jackson and Suchith Anand (2012). *Grid and Cloud Computing: Concepts, Methodologies, Tools and Applications* (pp. 615-636).

www.irma-international.org/chapter/implementing-geospatial-web-services-cloud/64506