

# Communication-Load Impact on the Performance of Processor Allocation Strategies in 2-D Mesh Multicomputer Systems

Zaid Mustafa,

Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), University of Santiago de Compostela, Spain

J. J. Alshaer

Department of Computer Information Systems,  
Albalqa Applied University  
Alsalt, Jordan

O. Dorgham

Department of Computer Information Systems,  
Albalqa Applied University  
Alsalt, Jordan

S. Bani-Ahmad

Department of Computer Information Systems,  
Albalqa Applied University  
Alsalt, Jordan

**Abstract**—A number of processor allocation strategies have been proposed in literature. A key performance factor that can highlight the difference between these strategies is the amount of communication conducted between the parallel jobs to be allocated. This paper aims to identifying how the density and pattern of communication can affect the performance of these strategies. Compared to the work already presented in literature, we examined a wider range of communication patterns. Other works consider only two types of communication patterns; those are the one-to-all and all-to-all patterns. This paper used the C language to implement the different allocation strategies, and then combined them with the ProcSimity simulation tool. The processor-allocation strategies are examined under the First-Come-First-Serve scheduling strategy. Results show that communication pattern and load are effective factors that have a significant impact on the performance of the processor allocation strategy used.

**Keywords**—Processor allocation; Parallel computing; 2-D Mesh; Communication patterns; Multicomputer systems

## I. INTRODUCTION

Nowadays, Multi-computer computer systems are widely used, because they are cost-effective alternatives of the traditional supercomputers [8]. The Topology of a multi-computer defined as a style based on the processing units of interconnected multi-computer. Mesh-based topologies considered as one of these topologies, such as the two-dimensional (2-D) and the three-dimensional (3-D) topologies. These topologies are the most common topologies; according to their simple, regular and scalable [1, 7,4].

Most of the modern commercial and experimental multi-computer systems (such as are the IBM BlueGene/L and the Intel Paragon) have been built based on this architecture [8, 4].

In a multi-computer system, the processor allocator applies processor allocation strategies in order to assign unallocated multi-computer nodes to parallel jobs and identify them [8]. Both of processor allocation algorithms and better idle-nodes

recognition ability for multi-computers or idle sub-meshes can reduce the job waiting time and improve the chance of assigning a parallel job into the system in a significant way [8]. The proper choice of the processor allocation strategy in 2-D-Mesh multi-computer consider as a critical issue in deciding the performance of a given multi-computer system. As it is significantly affects the performance (system utilization, response time, throughput ... etc) of any parallel system [8, 4].

Allocation strategies fall into two main categories, as follows:

- First: contiguous allocation strategies seek to allocate a sub-mesh, such as a contiguous set of processing units of the same size and shape of parallel request. Contiguity condition can be summarized in the relaxed of the non-contiguous allocation strategy [4]. Contiguous allocation strategies have low system utilization, high internal and external fragmentation, which consider as a problem to the contiguous allocation strategies.
- Second: non-contiguous strategies which can produce high execution times of parallel jobs according to high communication latencies [4, 5, 6].

Fragmentation problem prevents an idle processors from being utilized. Which can be classified into internal and external fragmentation. Internal fragmentation is the result of allocating jobs only to certain size sub-meshes. It occurs when a job is assigned to more processors than it requires, or extra nodes allocated are not used for actual computation [8]. Also, internal fragmentation occurs when a job requests a sub-mesh that does not fit the requirement of the allocation algorithm. For example, internal fragmentation occurs when a job does not require a square sub-mesh with sides equal to power of two and is allocated using the TDB scheme. The external fragmentation happens when the allocation scheme cannot allocate the available processors to the incoming jobs [8].

In addition to, the non-contiguous allocation strategies tries to fix the problems of external and internal fragmentation, beside the low system utilization by allowing parallel requests to be partitioned and allocated non-contiguously into smaller sub frames in case contiguous allocation fails [8, 4, 3]. Studies showed that allocation strategies integrated the advantages of both contiguous and non-contiguous allocation strategies throughout preserving some level of contiguity within allocated parallel job [3].

In this paper, the authors comparatively evaluates a number of partially non-contiguous allocation strategies that are proposed in [4, 3] in terms of their performance at high communication loads. A number of communication styles will be experimented. The allocation strategies of interest will be implemented in the C language, and later integrated with the ProcSimity simulation tool [12, 16]. The processor allocation strategies will be examined under the scheduling strategy called First-Come-First-Serve as in [7, 3].

## II. PROBLEM STATEMENT AND EXPERIMENTED COMMUNICATION PATTERNS

In this paper, contiguous and partially non-contiguous allocation strategies will be comparatively evaluated in terms of their performance at high communication loads (a number of communication styles will be experimented [4, 3]). The goal is to compare the allocation strategies in terms of their performance at *high communication loads*.

This paper tries to answer the following questions:

- Does the behavior of the processor allocation strategy in use change as the communication behavior of parallel jobs change?
- Does the behavior of the processor allocation strategy used change as the communication load caused by the parallel jobs change?

One communication scheme to be tested is the Fast Fourier Transform (FFT) pattern. The FFT algorithm is introduced as an efficient algorithm to compute the discrete Fourier transform "DFT" and it is inverse [19].

Another important communication patterns is the NAS parallel Bench marks (NPB) which represents a small set of programs designed to help evaluating the performance of parallel supercomputers. The benchmarks are derived from computational fluid dynamics (CFD) applications and consist of five Kernals and three pseudo-applications. The original benchmarks identified in NPB mimic the computation and data movement in CFD applications, the five Kernals is IS, EP, CG, MG,FT; and the three pseudo applications is BT, SP, LU [20; 21; 23].

The authors also tested the Divide-and-Conquer communication pattern. A Divide-and-Conquer-based algorithm solves a programming problem by recursively partition that problem into sub-problems of roughly equal size, if sub-problem can be solved independently; there is a possibility increase the speed up by parallel computing [24].

## III. LITERATURE REVIEW AND RELATED WORK

The performance of parallel systems can be significantly affected by the processor allocation strategy used in the 2D-Mesh multicomputer system in hand [1]. A number of studies have proposed allocation strategies. All proposed strategies can fall under two categories: namely; contiguous and non-contiguous strategies.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

### A. Contiguous Processor Allocation Strategies

In contiguous processor allocation of a given parallel request, a contiguous set of processing units of the same size and shape of that request are assigned to job of interest [17, 1].

In contiguous allocation strategies a low level of system utilization can be raised and external and/or internal fragmentation can be observed [8,11]. When a job needs to be allocated to a number processor, it may be assigned more processing units than what it actually requires. This is referred to as the internal fragmentation problem [8,7]. External fragmentation on the other hand, occurs when enough number of idle processors is available in the system but cannot be allocated to the scheduled parallel job because of the necessity of contiguity [1,4]. Many research efforts tried to solve or decrease the problem of external fragmentation [8,4], and one proposed solution was to use non-contiguous allocation strategies.

All the above allocation strategies are referred to as contiguous allocations because they consider only contiguous regions for the execution of a parallel job. In contiguous allocation, communication cost is minimal [6]. However, the requirement that a parallel job has to be allocated to contiguous set multi-computers reduces the chance of successfully allocating the job due to the problem of fragmentation [6,7].

Two Dimensional Buddy (2DB) [25]: in the Two Dimensional Buddy (2DB) strategy, the system is assumed to be a square with side lengths equal to a power of two ( $2^i$ , where  $i=0, 1, 2, \dots$ ). The size of a requested sub-mesh is rounded up to a square with side-lengths as the nearest power of two. The resulting sub-mesh after rounding up can be larger than the original sub-mesh. Consequently, this allocation strategy suffers from internal fragmentation because it only allocates square sub-meshes whose side lengths are equal to a power of two. In other words, because of rounding up the sides of requests, the allocated sub-mesh can be larger than the originally requested sub-mesh.

Frame Sliding (FS) [8]: The Frame Sliding (FS) method was proposed to reduce the internal fragmentation problem of the 2DB allocation. This is achieved through allowing sub-meshes of any arbitrary size to be allocated to parallel jobs. This allocation strategy works as follows: Viewing the requested sub-mesh of the job in hand as a frame, the FS algorithm slides the frame across the system to examine for a free sub-mesh to execute the job [8].

The First-Fit (FF) and the Best-Fit (BF) Strategies [18]: The FF and BF algorithms guarantee the recognition of a free sub-mesh, provided it exists. The two algorithms work by scanning the entire mesh for possible allocation.

Adaptive-Scan (AS) Strategy [9]: The adaptive-scan changes the orientation of the sub-mesh being searched for if the required sub-mesh in the original orientation is not available. Thus, the AS strategy has better recognition capabilities than that of the BF and FF schemes.

#### B. Non-contiguous Processor Allocation Strategies Processor

When using non-contiguous allocation the contiguity condition is not a must [8]; therefore, a job does not have to wait for a single sub-mesh of the requested size and shape to be available because it will be executed on a number of disjoint smaller sub-meshes [8; 15].

The non-contiguous allocation of requests can successfully solve the drawbacks of contiguous allocation strategies. As experimentally validated, non-contiguous allocation strategies have been found to produce relatively high system utilization and eliminate fragmentation [8, 15, 14].

Communication latency is often high in non-contiguous allocation strategies since communication between processors running the same job may be indirect due to non-contiguity [14].

The introduction of wormhole routing [10] encouraged researchers to consider non-contiguous allocation on multicomputers [8]. With wormhole routing, the message communication latency is less dependent on the distance traveled by the message from source to destination [8, 10].

In non-contiguous allocation schemes, allocation requests of parallel jobs are subdivided into two partitions [8] if contiguous allocation fails. If allocating either of the two partitions fails, that partition is further subdivided into smaller sub-partitions until the allocation succeeds [11]. Using wormhole routing has made allocating parallel jobs to non-contiguous processors reasonable in terms of performance even in networks characterized by a relatively long-diameter, "The diameter is the shortest hop "the maximum of the shortest distance between any two nodes", a small communication delay happens when an interconnection graph has a small diameter between nodes [22]. Such the 2-D mesh. So the contiguity condition is relaxed which allowed jobs to be executed without waiting for sufficient and contiguous set of idle processing nodes to be available in the system [7; 3; 11].

When "Wormhole routing" is a special case of cut-through switching. It routes the head of a packet directly from incoming to outgoing channels of the routing chip. A packet is divided into a number of flits (i.e. flow control digits) for transmission. The route or the path followed by those flits from source to destination is determined by the header flits (or flits). Flits are forwarded through a chosen channel after examining the header flit(s) of a message. As the header flit of a given packet/message advances along a specific route, all following flits follow in through the same route. [13].

There are many important strategies of the non-contiguous allocation such as Multiple Buddy System (MBS) expresses the allocation request as a base-4 number, and bases allocation on this expression. In this strategy, the mesh of the system at hand is divided into non-overlapping square sub-meshes with side lengths that are powers of 2. The number of processors,  $p$ , requested by a scheduled job is factorized into a base-4 block. If a required block is unavailable, MBS recursively searches for a larger block and repeatedly breaks it down into four buddies until it produces blocks of the desired size. If that fails, the requested block is further broken into four sub-requests until the job is allocated [3].

In the Paging allocation strategy, for instance [2], the entire 2D mesh is virtually sub-divided into pages or sub-meshes of equal sides' length of  $2i$  where  $i$  is a positive integer Number that represents the index parameter of the paging Approach. That can scan for pages in multiple ways such as that snake-line order and row-major order [3].

A main disadvantage of this strategy is the fragmentation it may cause when some free processors' cannot be allocated because they are contained in pages that have been reserved to job [13].

A partially non-contiguous allocation strategy tries to find a contiguous set of processing units of the same size to active job using some contiguous strategy. If it fails, the active job is divided into two sub-requests. The two new sub-requests are then allocated using the same contiguous allocation approach again; this operation will continue recursively until the request is finished. Examples of this allocation strategies are (i) the PALD-FF (PARTitioning at the Longest Dimension with First-Fit), (ii) the PALD-BF (PARTitioning at the Longest Dimension with Best-Fit) [3] and (iii) the restricted size reduction (RSR) strategy [3]. The RSR strategy allows jobs to be executed on a reduced size sub-mesh adaptively and partitions it in two sub-blocks of equal sizes.

In the PALD\_FF strategy, First-Fit approach used to find a contiguous group of processing units of the same size and shape of the application at hand. In case of fail, the request at hand is divided into sub requests after removing one from the longest dimension of the request that is for a given request of size  $a*b$  and assuming  $b>a$ , the two partition-sizes are  $a*(b-1)$  and  $a*1$  after removing one from the longest dimension of the request. The two new sub-requests are then allocated using the First-Fit approach again; this procedure continues recursively until the request is fulfilled [3].

According to the previous, the PALD\_BF is the same of PALD\_FF but it applies the best fit strategy rather than the first fit. processor allocation of a given parallel request, a contiguous set of processing units of the same size and shape of that request are assigned to job of interest [17, 1].

#### IV. RESULTS AND OBSERVATIONS

Figures 1 through 5 show the relationship between the system load from one side, and, from the other side, (i) the mean job response time, (ii) the mean job service time, (iii) the mean packet blocking time, (iv) the mean packet latency, and (v) the percent system utilization.

Figures 1 through 5 are specific for the case of having all-to-all communication pattern. The allocation strategies used in these figures are: (i) the Random, (ii) the Multiple Buddy System, (iii) the First-Fit, and (iv) the Best-Fit allocation strategies. The first two strategies are non-contiguous strategies, while the last two are contiguous. The following measures are used to evaluate allocation techniques: MJRT (Mean Job Response Time), MJST (Mean Job Service Time), MPBT (Mean Packet Blocking Time), MPL (Mean Packet Latency) and PSU (Percent System Utilization). The following Figures show results of simulation. Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable. Figure 1 represents the relationship between the system load and mean job response time with the allocation strategies "MBS, FF, BF, Random" under the first come first serve scheduling mechanism and all-to-all communication pattern. The communication load parameter here is the mean number of messages sent by any scheduled job. This parameter has made fixed at 80. Observation 1 (figure 1): In general, contiguous allocation strategies show higher MJRT than the non-contiguous allocation strategies.

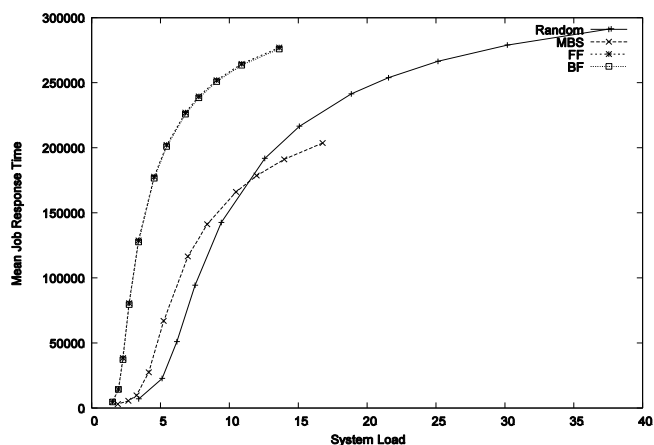


Fig. 1. Example Mean job response time (MJRT) vs. system load in multiple allocation strategies under the FCFS scheduling mechanism and all-to-all communication pattern (message count = 80)

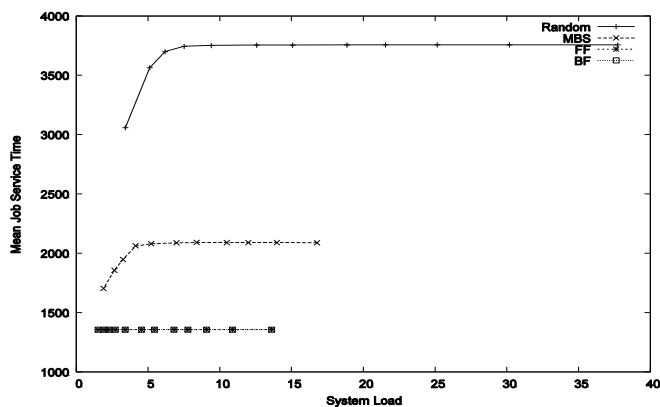


Fig. 2. Mean job service time vs. system load in multiple allocation strategies under the FCFS scheduling mechanism and all-to-all communication pattern (message count = 80)

This is because contiguous allocation strategies require that allocated processing units to be contiguous. The random allocation strategy showed less job response time compared to MBS at high system loads because that strategy randomly allocates the required processors.

Observation 2 (figure 2): Noncontiguous allocation strategies showed higher MJST than contiguous allocation strategies. This is because non-contiguous allocation strategies disperse the set of processors allocated to the parallel job when contiguous allocation fails. This increases the amount of time required for intra-process communication.

Observation 3 (figure 2): The MJST is higher when applying the random allocation strategy compared to the case when applying the MBS allocation strategy.

The random allocation strategy usually causes parallel jobs to be more dispersed compared to the MBS strategy. The MBS strategy is partially contiguous; it tries to maintain some level of contiguity between allocated processors.

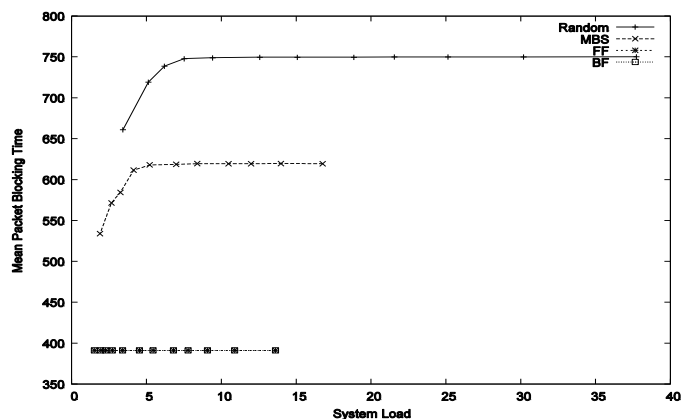


Fig. 3. Mean packet blocking time vs. system load in multiple allocation strategies under the FCFS scheduling mechanism and all-to-all communication pattern (message count = 80)

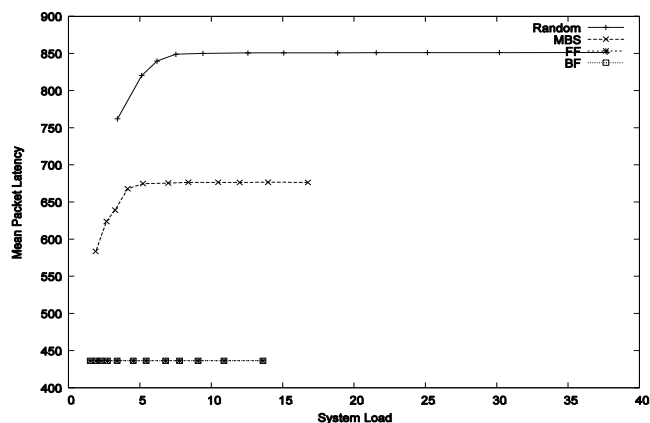


Fig. 4. Mean packet latency vs. system load in multiple allocation strategies under the FCFS scheduling mechanism and all-to-all communication pattern (message count = 80)

Non-contiguous allocation strategies causes more disperse to allocated parallel jobs producing longer distances between the sources and the destinations of messages. Since each message is blocked for sometime at each intermediate node

from source to destination, the mean packet blocking time increases as the number of these intermediate nodes increases. The mean packet latency also increases accordingly.

Observation 5 (figure 5): Contiguous allocation strategies showed low PSU values compared to noncontiguous allocation strategies. For example, at relatively high system loads, the random allocation strategy produces the highest percent system utilization of all experimented allocation strategies in figure 15. This is because it simply any idle processor from the list of free processors. The MBS strategy also allocates processors in a noncontiguous manner. Both strategies successfully allocate the parallel job in hand as long as enough number of free nodes is available. The FF and BF both forces the parallel job to wait if no enough contiguous set of processors is available in the system. This, in turn, may produce high waiting time for relatively large parallel requests.

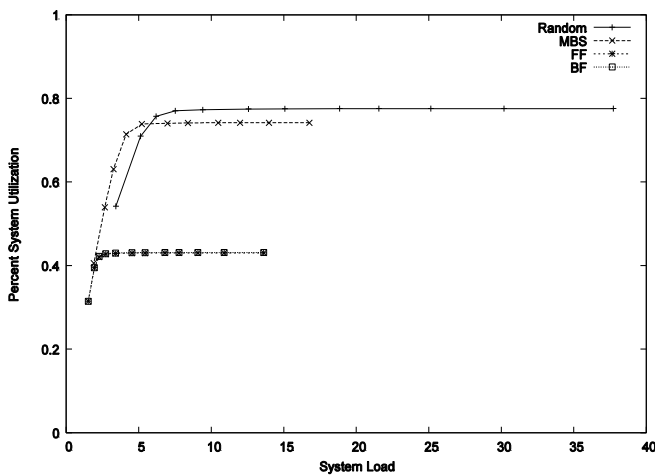


Fig. 5. Percent system utilization vs. system load in multiple allocation strategies under the FCFS scheduling mechanism and all-to-all communication pattern (message count = 80)

Figures 6 through 20 shows the relationships between the system load from one and, from the other side, (i) the mean job response time, (ii) the mean job service time, (iii) the mean packet blocking time, (iv) the mean packet latency, and (v) the percent system utilization. The impact of multiple communication patterns is studied next. Those patterns are: all-to-all, one-to-all, random, Fast Fourier Transform (FFT), and the NAS multigrid benchmark communication patterns on the key performance parameters of the multiprocessor system.

Figures 6 through 10 are specific for the contiguous allocation strategies. The authors use the BF allocation strategy as a representative for this group of allocation strategies.

Figures 11 through 15 are specific for the noncontiguous allocation strategies. The authors use the MBS allocation strategy as a representative for this group of allocation strategies.

Figures 16 through 20 are again specific for the noncontiguous allocation strategies. However, The authors use the BGP-BF allocation strategy as a representative for this

group of allocation strategies. In these experiments, the partitioning bound that is applied is equal to 4.

Observation 4 (figures 3 and 4): noncontiguous allocation strategies showed higher MPBT and MPL values than contiguous allocation strategies. This observation can be explained the same way as observations 3 and 4. Observation 9 (figures 6 and 7): One-to-all, All-to-all and random communication patterns produces the highest MJRT and MJST of all tested communication patters.

Next is the list of testbed communication patterns sorted starting with the one caused the highest MJRT: one-to-all, all-to-all, random, NAS-multigrid, FFT, and DQBT. This can be explained as follows, one-to-all and all-to-all communication patterns generate large number of messages to be transmitted over the interconnection network. This increases the service time of allocated parallel jobs and, thus, forces unallocated parallel jobs to wait more in the ready queue of the system. This, in turn, increases the mean job response time.

Figures 8 and 9 show the effect of communication pattern on the MPBL and MPL values of the system in the case of applying the BF allocation strategy. Notice the behavior of the system is quiet strange and that the figure is not directly conclusive. This is because that the number of jobs served during the simulator is relatively low because of the following factors:

(i)The allocation strategy applied in contiguous. This reduces the number of allocated jobs due to the condition of contiguity.

(ii) The scheduling mechanism applied is the FCFS. This may prevent many parallel jobs from being allocated if a relatively big parallel job is residing at the head of the ready queue of the system. Based on the above two factors, the authors believe that the behavior of the system is not clear in the figures 8, 9 and 10. To remedy this problem and to better capture the impact of communication pattern type. The authors next examines applying a non-contiguous allocation strategy.

Next, the authors present their observations on these figures.

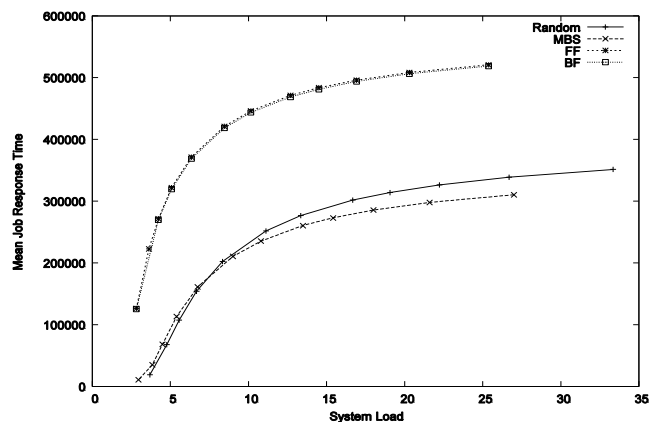


Fig. 6. Mean job response time vs. system load in Best Fit allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

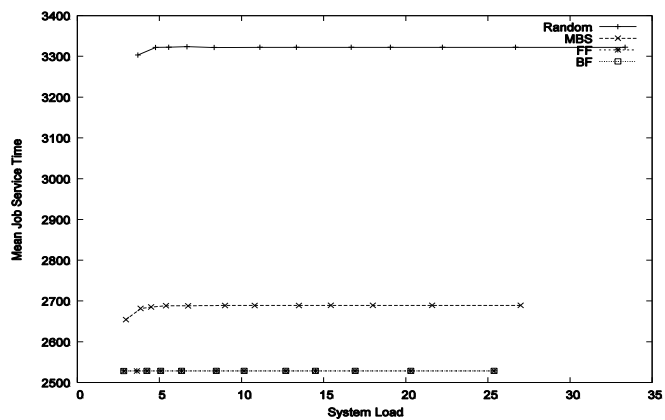


Fig. 7. Mean job service time vs. system load in Best Fit allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

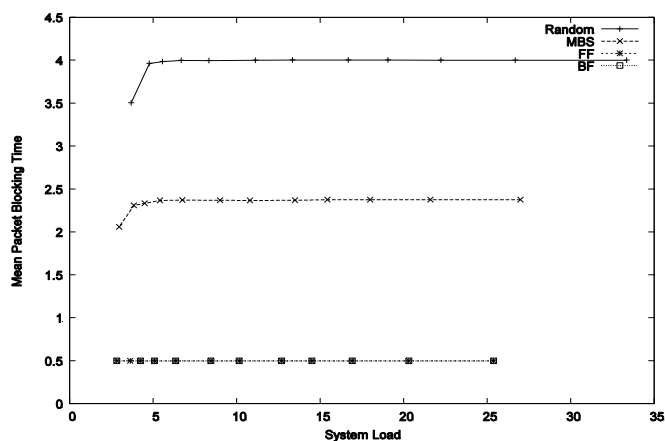


Fig. 8. Mean packet blocking time vs. system load in Best Fit allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

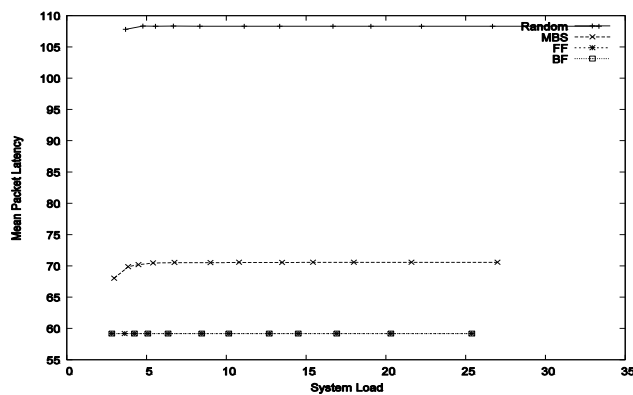


Fig. 9. Mean packet latency vs. system load in Best Fit allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 0)

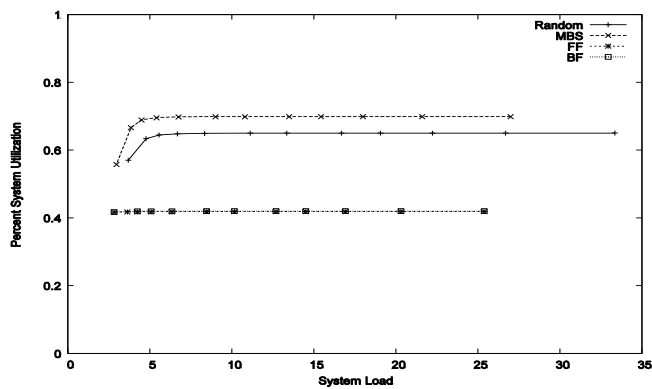


Fig. 10. Percent system utilization vs. system load in Best Fit allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

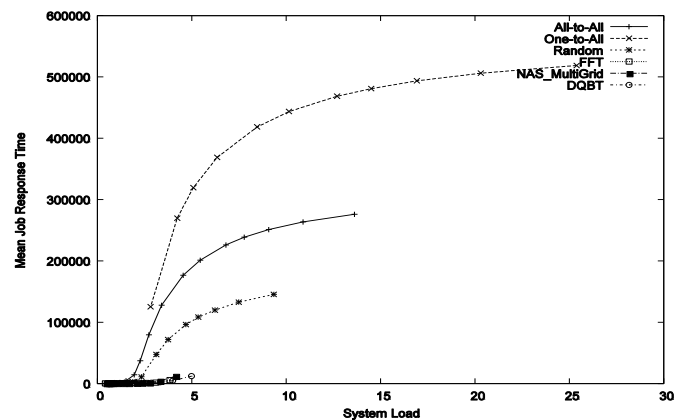


Fig. 11. Mean job response time vs. system load in MBS allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

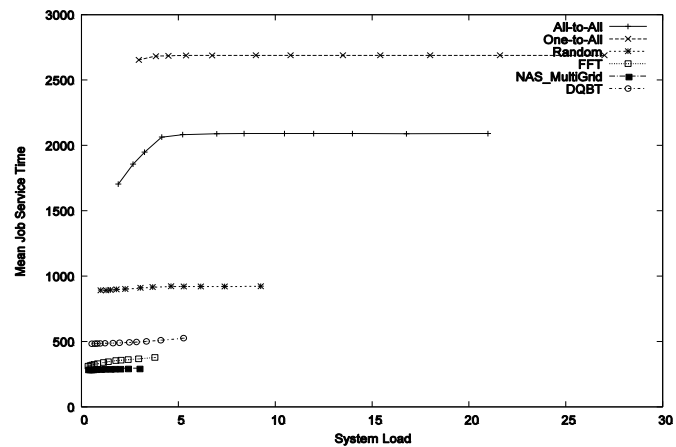


Fig. 12. Mean job service time vs. system load in MBS allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

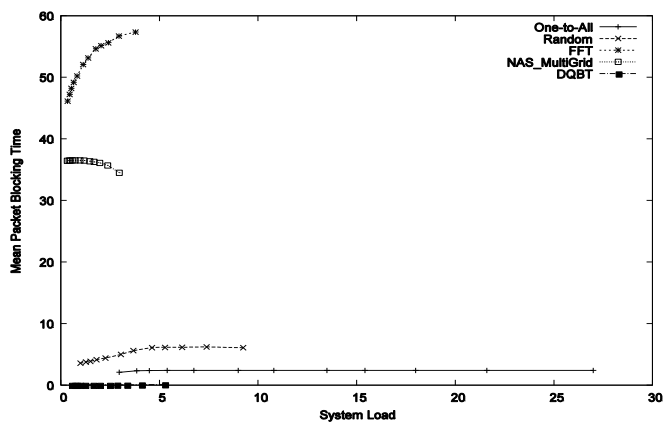


Fig. 13. Mean packet blocking time vs. system load in MBS allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

Figures 13 and 14 plot the MPBT, the MPL and the PSU vs system load when using the MBS allocation strategy and multiple communication patterns.

Observation 11 (figures 8, 9, 10, 13, 14, and 25): the behavior of the parallel system can be significantly affected by the communication pattern of the parallel jobs being allocated.

Figures 11 through 15 are specific for the noncontiguous allocation strategies. The authors use the MBS allocation strategy as a representative for this group of allocation strategies.

Figures 11 and 12 showed similar results to figures 6 and 7 that have been explained before (with the BF allocation strategy). One difference can be observed.

Figures 16 to 20 show the relationships between the system load from one side, and, from the other side, (i) the mean job response time, (ii) the mean job service time, (iii) the mean packet blocking time, (iv) the mean packet latency, and (v) the percent system utilization., that's focus to the BGP-BF allocation strategy.

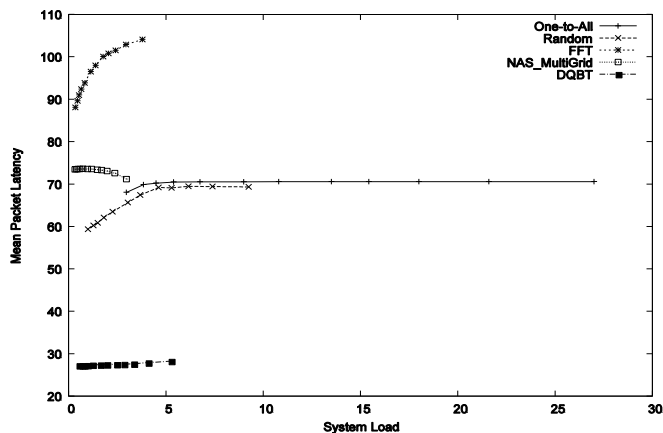


Fig. 14. Mean packet latency vs. system load in MBS allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

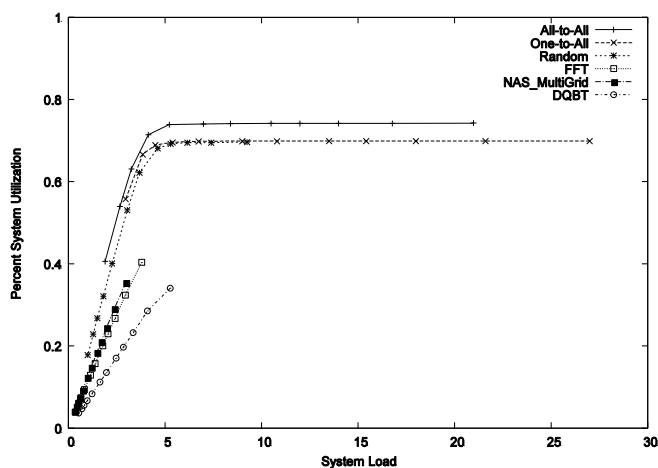


Fig. 15. Percent system utilization vs. system load in MBS allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

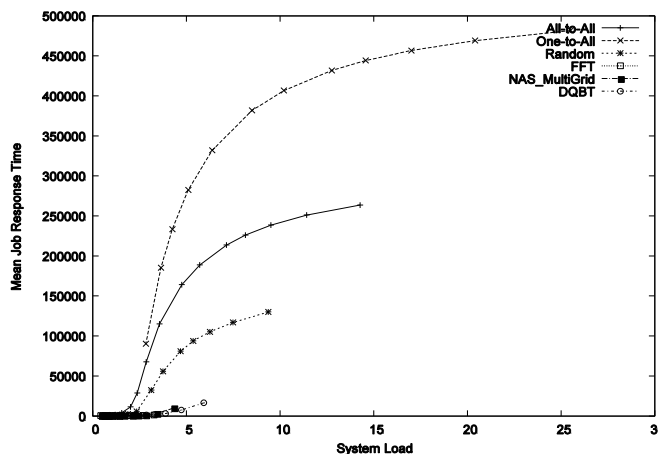


Fig. 16. Mean job response time vs. system load in the BGP-BF (partitioning bound = 4) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

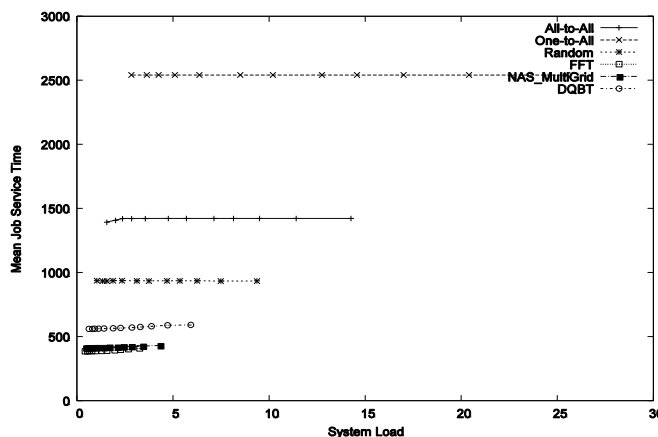


Fig. 17. Mean job service time vs. system load in the BGP-BF (partitioning bound = 4) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

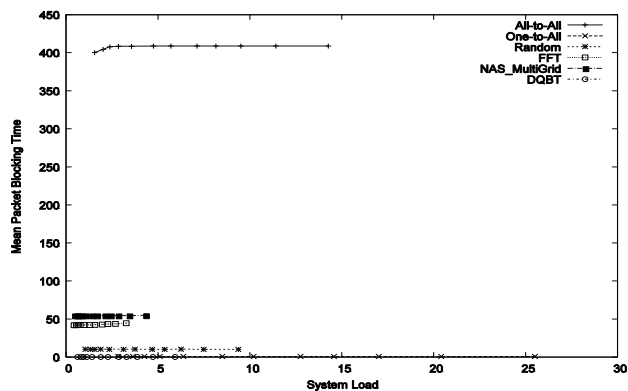


Fig. 18. Mean packet blocking time vs. system load in the BGP-BF (partitioning bound = 4) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

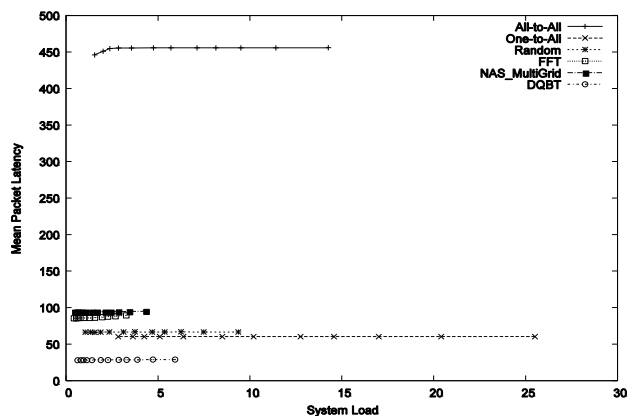


Fig. 19. Mean packet latency vs. system load in the BGP-BF (partitioning bound = 4) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80). Conclusion

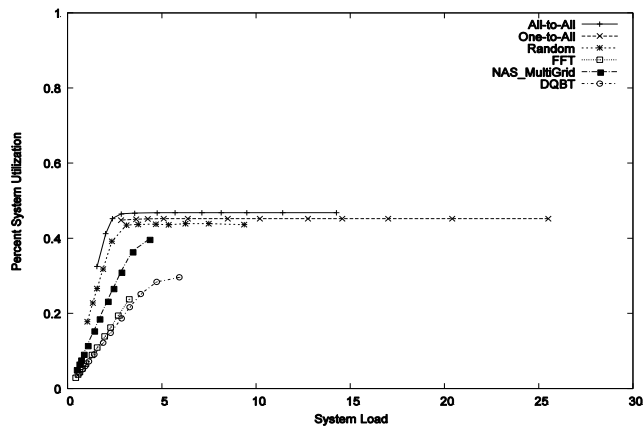


Fig. 20. Percent system utilization vs. system load in the BGP-BF (partitioning bound = 4) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

Observation 12 (figures 10, 15 and 20): the behavior of the parallel system can be significantly affected by the communication pattern of the parallel jobs being allocated.

For instance, the random and the one-to-all communication patterns produced similar PSU value when applying the MBS allocation strategy. However, the one-to-all allocation strategy

outperformed the random communication pattern when using the BF and the BGP-BF strategy. Next, the authors study the impact of the partitioning bound of the BGP-BF allocation strategy on the different system performance parameters. This will be in figures 11 through 20. Figures 11 to 15 are specific for the all-to-all communication pattern. While figures 16 to 20 are specific for the communication pattern one-to-all.

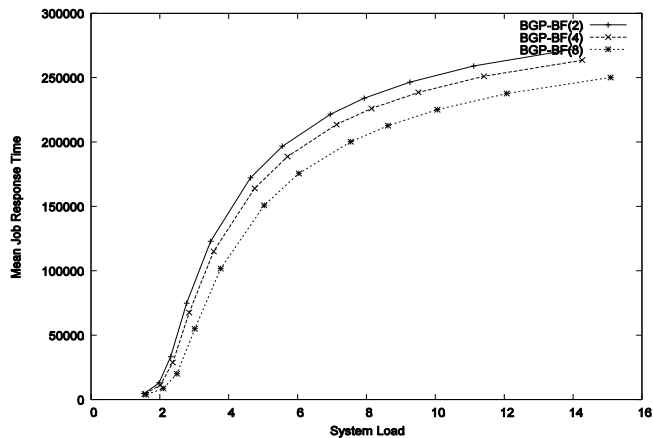


Fig. 21. Mean job response time vs. system load in the BGP-BF (partitioning bounds = 2, 4, and 8) allocation strategy under the FCFS scheduling mechanism and all-to-all communication pattern (message count = 80)

Observation 13 (Figure 21): In general, when the bounded is high the mean job response time is less.

For example, the BGP-BF (2) produced the highest MJRT value and followed by BGP-BF (4) and BGP-BF (8).

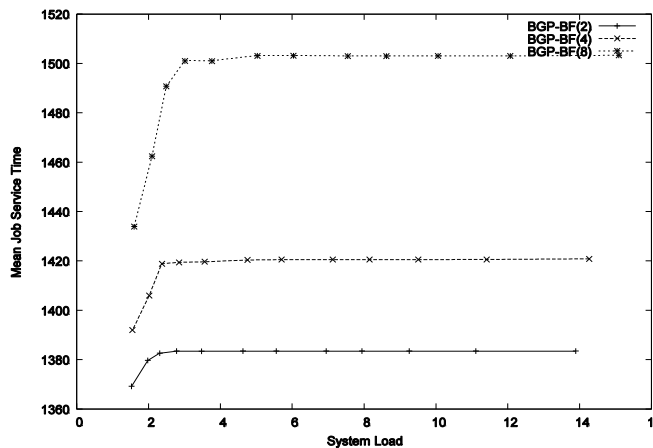


Fig. 22. Mean job response time vs. system load in the BGP-BF (partitioning bounds = 2, 4, and 8) allocation strategy under the FCFS scheduling mechanism and all-to-all communication pattern (message count = 80)

Figure 22 represents the mean job service time and system load in BGP-BF (partitioning bounds=2, 4 and 8) allocation strategy.

Observation 14 (figure 22): When increasing the partitioning bounded, the mean job service time increases also.

This is because when increasing the partitioning bound, the execution time increases because the level on non-contiguity increases. The results in an increase in the service time of the job.



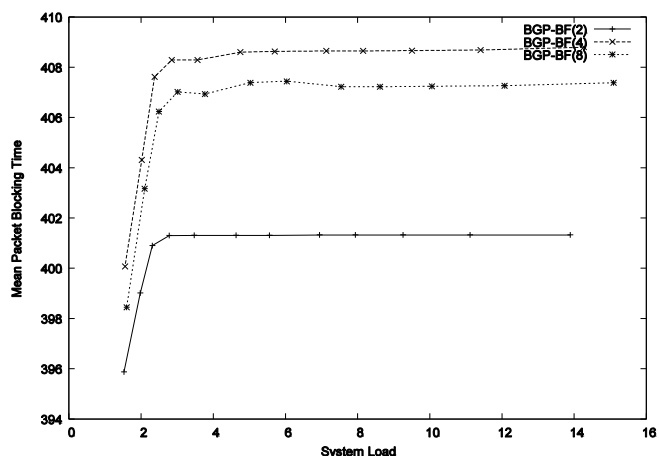


Fig. 23. Mean packet blocking time vs. system load in the BGP-BF (partitioning bounds = 2, 4, and 8) allocation strategy under the FCFS scheduling mechanism and all-to-all communication pattern (message count = 80)

Figures 23, 24 plot the relation between the mean packet blocking time and the mean packet latency (y-axis) and the system load (x-axis) when applying the BGP-BF (partitioning bounds = 2, 4, and 8) allocation strategy under the FCFS scheduling mechanism and all-to-all communication pattern.

Observation 15 (figures 23 and 24): In general, when the partitioning bound is high the MPBT and the MPL is high.

This is because when the bound is high the time that message packets spend blocked in network buffers, waiting for access to their next channel is high.

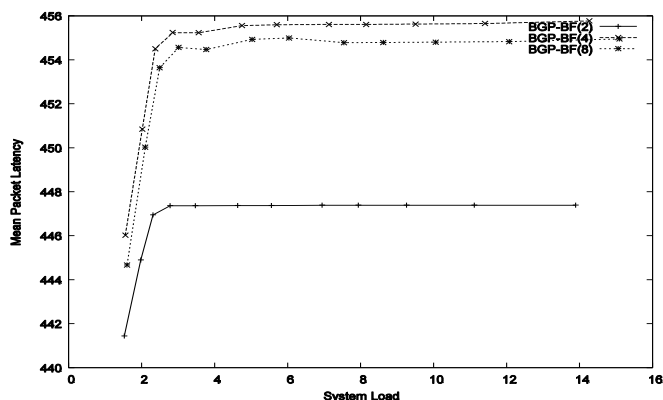


Fig. 24. Mean packet latency vs. system load in the BGP-BF (partitioning bounds = 2, 4, and 8) allocation strategy under the FCFS scheduling mechanism and all-to-all communication pattern (message count = 80)

The figure 24 that represent the relation between mean packet latency and system load in the BGP-BF "partitioning bounds=2, 4 and 8" allocation strategy under the FCFS scheduling mechanism and all-to-all communication pattern.

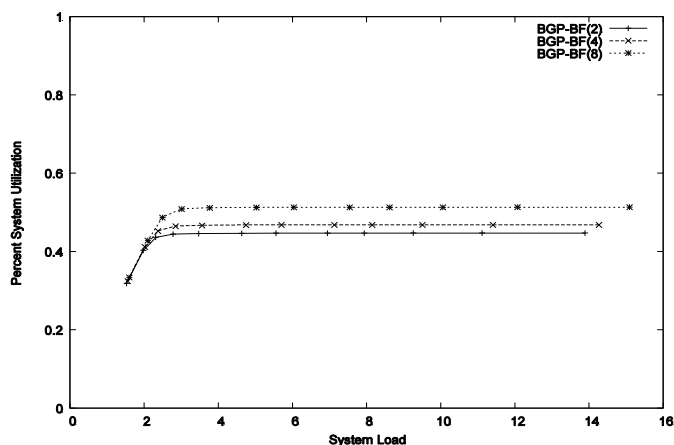


Fig. 25. Percent system utilization vs. system load in the BGP-BF (partitioning bounds = 2, 4, and 8) allocation strategy under the FCFS scheduling mechanism and all-to-all communication pattern (message count = 80)

Figure 25 plots the percent system utilization and system load in the BGP-BF (partitioning bounds = 2, 4, and 8) allocation strategy under the FCFS scheduling mechanism and all-to-all communication pattern.

Observation 16 (figure 25): when the partitioning bound increase so the percent system utilization increases.

BGP-BF (8) showed the highest PSU values, followed by the BGP-BF (4), BGP-BF (2) respectively. This because it's when the bound is higher the processors of the multicomputer system is utilized more efficiently.

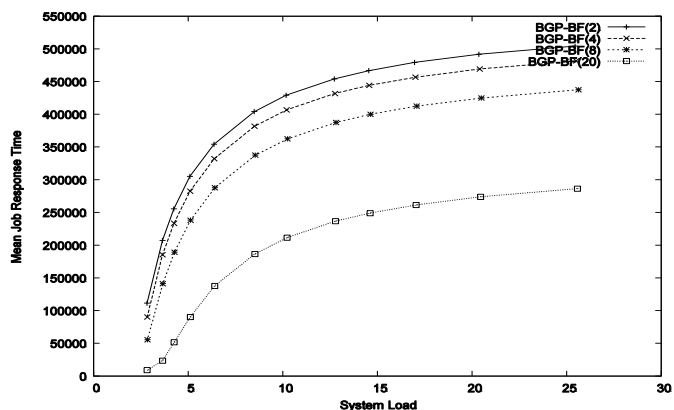


Fig. 26. Mean job response time vs. system load in the BGP-BF (partitioning bounds = 2, 4, 8, and 20) allocation strategy under the FCFS scheduling mechanism and one-to-all communication pattern (message count = 80)

Figures 26 through 30 (one-to-all communication pattern) gives similar observations to the figures 36 through 40 (all-to-all communication pattern).

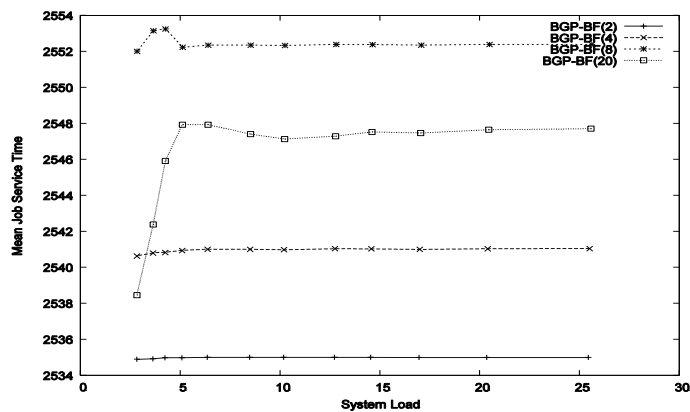


Fig. 27. Mean job service time vs. system load in the BGP-BF (partitioning bounds = 2, 4, 8, and 20) allocation strategy under the FCFS scheduling mechanism and one-to-all communication pattern (message count = 80)

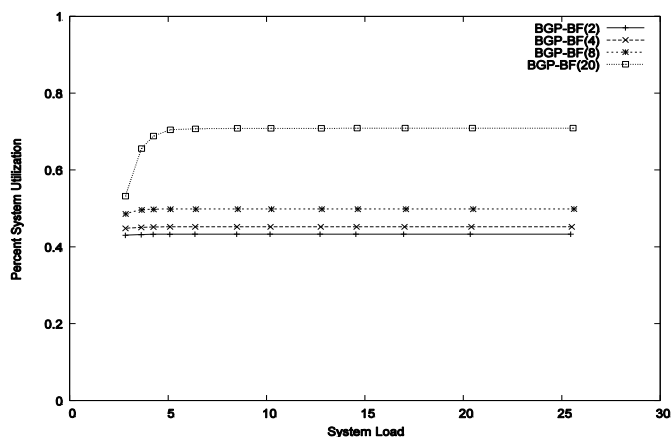


Fig. 30. Percent system utilization vs. system load in the BGP-BF (partitioning bounds = 2, 4, 8, and 20) allocation strategy under the FCFS scheduling mechanism and one-to-all communication pattern (message count = 80)

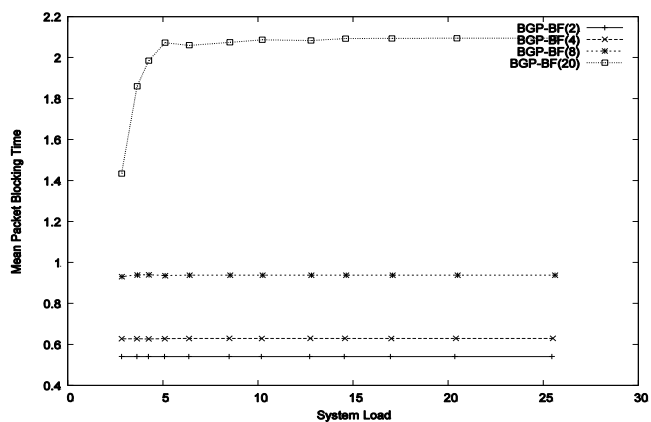


Fig. 28. Mean packet blocking time vs. system load in the BGP-BF (partitioning bounds = 2, 4, 8, and 20) allocation strategy under the FCFS scheduling mechanism and one-to-all communication pattern (message count = 80)

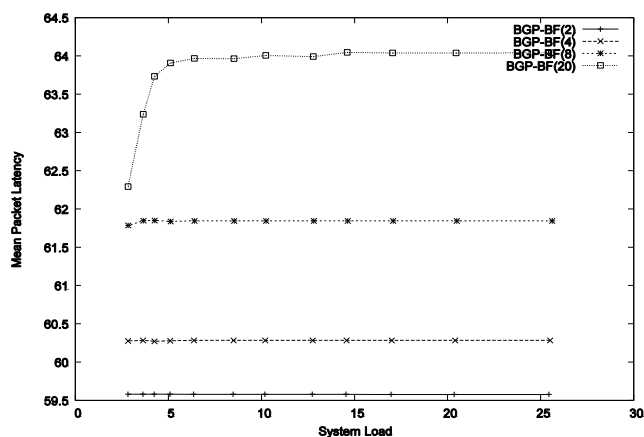


Fig. 29. Mean packet latency vs. system load in the BGP-BF (partitioning bounds = 2, 4, 8, and 20) allocation strategy under the FCFS scheduling mechanism and one-to-all communication pattern message count = 80)

Figures 31 through 35 represent the relation between the system load and the following system performance parameters: the mean job response, the mean job service time, the mean packet blocking time, mean packet latency and the percent system utilization. Here, the authors focus on the BGP-BF (partitioning bound=8) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns such as all-to-all, one-to-all, random, NAS\_multigrid, DQBT.

Notice that how the key system parameters; the MJRT, MJST, MPBT, MPL and the PSU change over changing the communication pattern of the parallel jobs being scheduled.

The above discussion and set of observations clearly shows that, depending on the performance parameter of most interest, the system can dynamically modify the partitioning bound of the BGP allocation strategy.

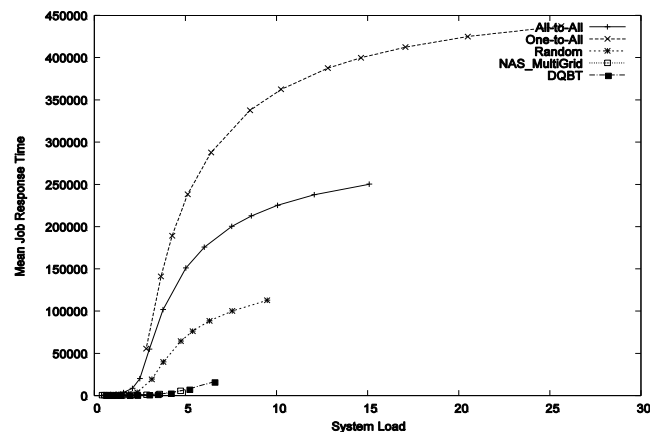


Fig. 31. Mean job response time vs. system load in the BGP-BF (partitioning bound = 8) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

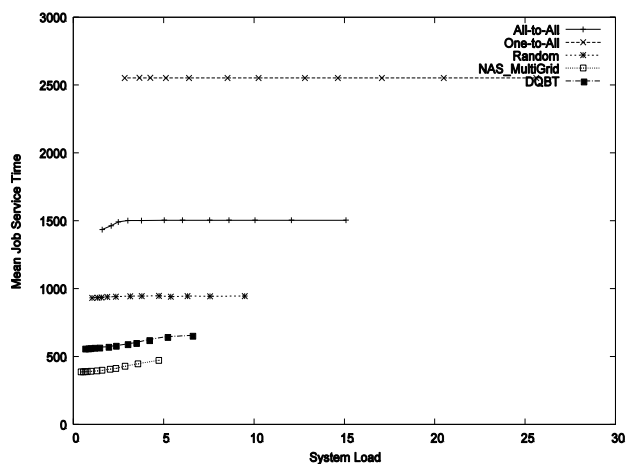


Fig. 32. Mean job service time vs. system load in the BGP-BF (partitioning bound = 8) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

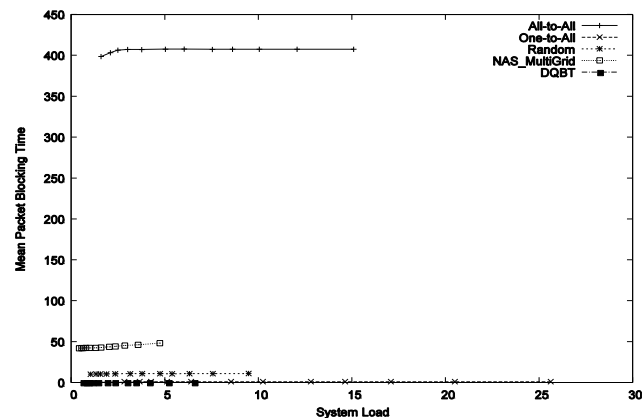


Fig. 33. Mean packet blocking time vs. system load in the BGP-BF (partitioning bound = 8) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

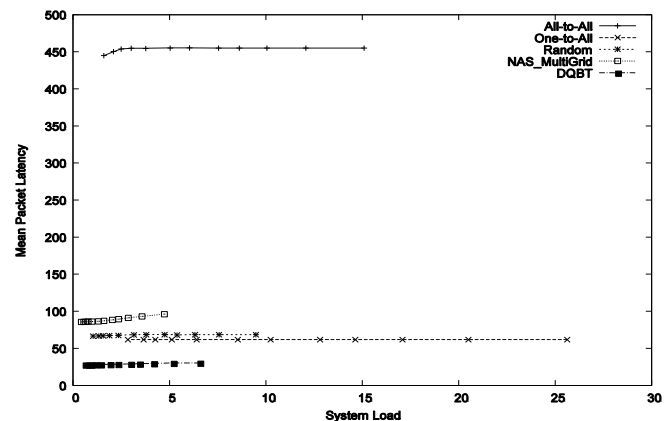


Fig. 34. Mean packet latency vs. system load in the BGP-BF (partitioning bound = 8) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

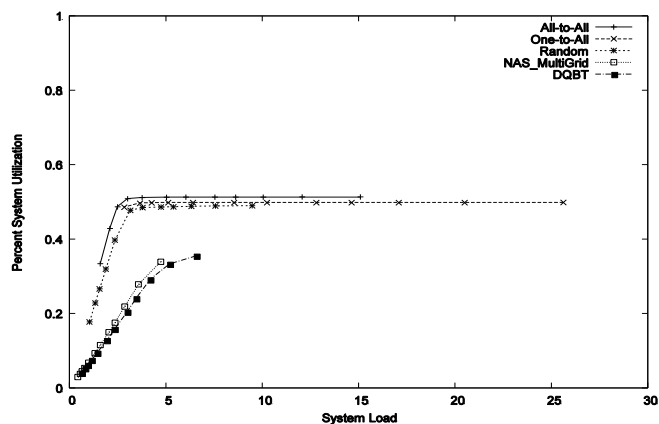


Fig. 35. Percent system utilization vs. system load in the BGP-BF (partitioning bound = 8) allocation strategy under the FCFS scheduling mechanism and multiple communication patterns (message count = 80)

## V. CONCLUSION

A key performance factor that can highlight the difference between allocation strategies is the amount of communication conducted between the parallel jobs to be allocated. In this paper, the authors show that's the type and pattern of communication can affect the performance of these strategies, compared to the communication pattern that are usually used in literature to evaluate processor allocation strategies, the authors examined wider range of communication patterns in the current work. Other works consider only two types of communication patterns; those are the one-to-all and all-to-all patterns. The authors found that the communication behavior of the parallel jobs being allocated can have a significant impact on the performance of the processor allocation strategy being applied. This observation is correct for both, contiguous and non contiguous strategies.

## REFERENCES

- [1] Ababneh, I., "An efficient free-list sub-mesh allocation scheme for two-dimensional mesh-connected multicomputers", *Journal of Systems and Software*, vol. 79, no. 8, Elsevier Science Inc., New York, NY, USA, August 2006, pp. 1168-1179.
- [2] Alqadi, R. and M. Khammash. "An Efficient Parallel Gauss-Seidel Algorithm for the Solution of Load Flow Problems". *The International Arab Journal of Information Technology*. Vol. 4, No. 2. April 2007.
- [3] Bani-Ahmad, S., "Sub-mesh Allocation in 2D-Mesh Multicomputer: Partitioning at the Longest Dimension of Requests". *Proceedings of the Fourth International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2010)*. October 25-30, 2010, Florence, Italy.
- [4] Bani-Ahmad, S., "Bounded Grand-Request-Based Allocation Strategies in 2D Mesh-based Multicomputers". *International Journal of Digital Content Technology and its Applications*. Volume 5, Number 1, January 2011.
- [5] Bani-Ahmad, S., "Comparative Evaluation of Request-Partitioning-Based Processor Allocation Strategies in 2D Mesh-based Multicomputers". *International Journal of Computer Applications (0975-8887)*. Volume 26, Number 7, July 2011.
- [6] Bani-Ahmad, S., "Processor Allocation with Reduced Internal and External Fragmentation in 2D Mesh-based Multicomputers". *Journal of Applied Sciences*. Volume 11, Issue: 6, 2011, pp 943-952.

- [7] Bani-Mohammad, S.; Ould-Khaoua, M.; Ababneh, I., and Machenzie, L., "Non-contiguous Processor Allocation Strategy for 2D Mesh Connected Multicomputers Based on Sub-meshes Available for Allocation", Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS'06), vol. 2, IEEE Computer Society Press, USA, 2006, pp. 41-48.
- [8] Chuang, P. J. and N.F.Tzeng, "Allocating Precise sub-mesh in Mesh-Connected Systems" IEEE Trans. on Parallel and Distributed Systems, pp.211-217, Feb, 1992.
- [9] Ding, J. and L. N. Bhuyan, "An Adaptive Sub-mesh Allocation Strategy for Two-Dimensional Mesh Connected Systems", Proceedings of International Conference on Parallel Processing, Vol.11, pp.193-200, Aug.1993.
- [10] Kumar, V., Grama A., Gupta A., and Karypis G.. "Introduction To Parallel Computing", The Benjamin/Cummings publishing Company, Inc., Redwood City, California, 2003.
- [11] Lo, V., Windisch K., Liu W., and Nitzberg B.. "Non-contiguous processor allocation algorithms for mesh-connected multicomputers", IEEE Transactions on Parallel and Distributed Systems, vol. 8, no. 7, IEEE Press, Piscataway, NJ, USA, July 1997, pp. 712-726.
- [12] ProcSimity V4.3 User's Manual, University of Oregon, [ftp://ftp.cs.uoregon.edu/pub/lo/procsim\\_man.ps.gz](ftp://ftp.cs.uoregon.edu/pub/lo/procsim_man.ps.gz), 1997.
- [13] Seydim A. Y., "Wormhole Routing in Parallel Computers", school of Engineering and Applied Sciences, Southern Methodist University, May 1998.
- [14] Srinivasan, T., Seshadri J., Chandrasekhar A., and Jonathan J.. "A Minimal Fragmentation Algorithm for Task Allocation in Mesh-Connected Multicomputers", Proceedings of IEEE International Conference on Advances in Intelligent Systems – Theory and Applications – AISTA 2004 in conjunction with IEEE Computer Society, ISBN 2-9599-7768-8, IEEE Press, Luxembourg, Western Europe, 15-18 Nov 2004.
- [15] Suzaki, K., Tanuma H., Hirano S., Ichisugi Y., Connelly C., and Tsukamoto M.. "Multi-tasking Method on Parallel Computers which Combines a Contiguous and Non-contiguous Processor Partitioning Algorithm", Proceedings of the Third International Workshop on Applied Parallel Computing, Industrial Computation and Optimization, Springer-Verlag, UK, 1996, pp. 641-650.
- [16] Windisch, K.; Miller, J. V.; and Lo, V., "ProcSimity: an experimental tool for processor allocation and scheduling in highly parallel systems", Proceedings of the Fifth Symposium on the Frontiers of Massively Parallel Computation (Frontiers'95), IEEE Computer Society Press, Washington, USA, 6-9 Feb 1995, pp. 414-421.
- [17] Yoo, B. S. and Das, C. R.. "A Fast and Efficient Processor Allocation Scheme for Mesh-Connected Multicomputers", IEEE Transactions on Parallel & Distributed Systems, vol. 51, no. 1, IEEE Computer Society, Washington, USA, January 2002, pp. 46-60.
- [18] Zhu, Y. H., "Efficient Processor Allocation Strategies for Mesh-Connected Parallel Computers", Journal of Parallel and Distributed Computing, 16, pp 211-337, 1992
- [19] Massey, G. & McClintock, J.. "Packet Size Matters in IP Transport", Copyright 2006 IMAS Publishing (USA), Inc.
- [20] David Bailey, Tim Harris, William Saphir, Rob Van der Wijngaart, Alex Woo, Maurice Yarrow (1995), "The NAS Parallel Benchmarks 2.0", Report NAS-95-020, December, 1995.
- [21] F. Shimojo, R. K. Kalia, A. Nakano, and P. Vashishta, "embedded divide-and-conquer algorithm on hierarchical real-space grids: parallel molecular dynamics simulation based on linear-scaling density functional theory," Comput. Phys. Commun. 167, 151.
- [22] Rolf Riesen, "Communication patterns", IEEE In Workshop on Communication Architecture for Clusters CAC'06, Rhodes Island, Greece, April 2006.
- [23] Singh, K. & Kilarikar, N.. "Tcp Flow Control Mechanisms For Single Packet Loss", paper presented in the Proceedings of the National Conference "NCNTE-2012" at Fr. C.R.I.T., Vashi, Navi Mumbai, Feb. 24-25, 2012. 1
- [24] K. Wendy Tang, Member, IEEE, and Sanjay A. Padubidri, " Diagonal and Toroidal Mesh Networks", IEEE TRANSACTIONS ON COMPUTERS ,VOL.43 ,NO.7, July, 1994
- [25] Li, K. and K.H Cheng, "A Two-Dimensional Buddy System for Dynamic Resource Allocation in a Partitionable Mesh Connected System", Journal of Parallel and Distributed Computing, 12, pp.79-83, 1991.