ISSN 2278-3091
Volume 4, No.2, March - April 2015
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/ijatcse/static/pdf/file/ijatcse06422015.pdf

Evaluation of Parallel Processing Systems through Queuing Model

Vikas Shinde
Department of Applied Mathematics,
Madhav Institute of Technology & Science, Gwalior-India

ABSTRACT

In this investigation, Jackson queueing network has been widely used to model and analyze the performance of complex parallel systems. M/G/1 queueing system is used to model a parallel processing system, which is expandable in vertical and horizontal manner. Determine a closed form solution for the system performance metrics, such as processor's waiting time, system processing power, etc.

Keywords: Queueing Network, Massive Parallel Processing, Shared Memory, Waiting Time.

1. INTRODUCTION

Parallel processing of the computer systems has been widely studied due to a significant role in day-by-day fast computing of the jobs. As parallel computing systems proliferate the need for effective performance evaluation, queueing techniques become ever more important. In fact, the performance of such systems depends on the hardware resources, (CPU, Memory, etc.,) on software (system programs, compilers, etc.,) and on the organization and management of these resources. In view of the increasing complexity of computing systems, it is more and more difficult to predict their performance indices based on analytical queueing models. In such models, it is convenient to represent the resources as 'servers' and the programs as 'customers'. A model of parallel processing system is a system which is expandable in vertical and horizontal manner and can be treated as cluster for a single queue of waiting jobs. A job is modeled as a sequence of independent stages which must be processed, where the number of processors desired by the jobs in each stage may be different. If, for some stage, the job in service requires fewer processors than the system provides, then the job will occupy the processors according to its need and the other processors will be idle for that stage. If, for some other stage, the job in service requires more processors than the system provides, then it will use all the processors in the system for an extended period of time such that the total work served in that stage is conserved.

Many researchers have extensively investigated processing systems via queue theoretic approaches. Al-Saqabi et al. [1] established a distributed scheduling algorithm that will track the available workstations i.e the workstations not being used by their owners in networks and act upon those workstations by scheduling processes of parallel applications onto them. Guan and Cheung [2] constructed a massively parallel processing system which has drawn a lot of attention to an important feature affecting the performance and characteristics of the architecture with an interconnection of multiple processors.

Jean-Marie et al. [5] introduced a hybrid analytical approach by using techniques from the theories of both stochastic task graphs and queueing networks. Jozwiak and Jan [6] discussed quality driven model based multi processor accelerator design method that adequately addresses the architecture design issues of hardware multi processors for the modern highly demanding embedded applications. Jan [7] studied communication architectures for massively parallel hardware multi processors. Systematic framework and a corresponding methodology for workload modeling of parallel systems was proposed by Kotsis [8]. Mohapatra et al. [10] proposed the structure for processors which is divided into groups or cluster and organized in several stages. Maheshwari and Shen [11] established a clustering algorithm wherein all the clusters have balanced amount of computation load and there is only one communication path between any pair of clusters. Nassar [12] evaluated the throughput of several multi buses as a discrete time Markov chain under different working conditions. Reijns [13] considered the delay effect caused by memory interference in a parallel processing system with shared memory was implemented using a machine repair queueing. Tomic [14] gave the matrix representation of the linear evolution operator of the certain class of parallel processing system and effectively used as a performance prediction tool for the modern parallel processing systems. Wasserman et al. [15] studied the problem of dynamic allocation of the resources of a general parallel processing system comprised of several heterogeneous processors.

The rest of paper is organized as follows. Model description is given in section 2. In section 3, described the governing equation and their performance analysis. Conclusion is mentioned in section 4.

2. MODEL DESCRIPTION

Every computer consists of a set of processors (CPUs) P_1, P_2, P_3, ……P_n and $m \geq 0$ shared memory units M_1, M_2, M_3,…..M_m which communicate via an interconnection network N, as illustrated in figure 1. The memory units constitute a global main memory that provides a convenient message depository for processor-to-processor communication. A system with this arrangement is called a shared memory computer. A global shared memory can be a serious bottleneck, particularly when the processors share large amounts of information, since normally only one processor can access a given memory module at a time. If the processors have their own local memories, then the global memory can be reduced in size, or even eliminated completely. To separate the functions of processing and memory, which refer to a CPU with no associated main memory, but with other temporary storage units such as register files and caches as a processing element (PE).

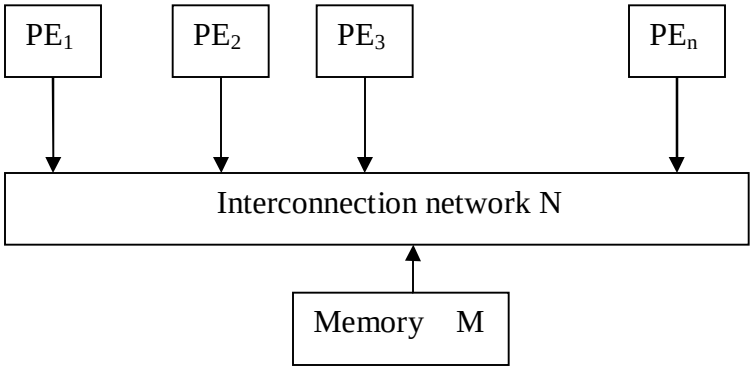

Figure **1: S**hared Memory

The basic cluster shown in figure 2, each processing unit has a local memory for its own computation and there is a shared memory for facilitating the communication between the processors. A horizontal communication network (HCN) is used for transmitting data between processors and shared memory. Moreover the basic cluster includes a unit for I/O operations and a unit for supervisory and managing the processors. A vertical communication network (VCN) is used for transmitting control signals and vertical expansion of the system.

The basic cluster is defined in two ways: (i) by increasing the number of the processing units or using several basic clusters with one additional memory that is shared by those clusters, and (ii) in a two stage system, it must be noted that in the second level of the system, there is a HCN that connects the VCN of each basic cluster to SM_2. The units that are located inside the basic clusters are indicated by (SM_1, HCN_1, ….), and the units that are located outside of the cluster are indicated by (SM_2, HCN_2,……)

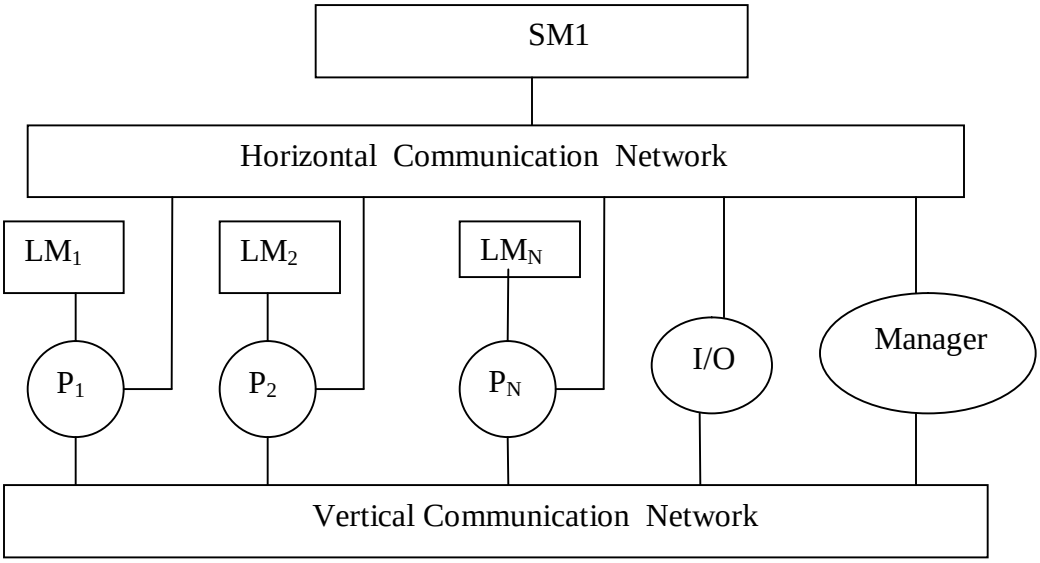

Figure **2:** Basic Cluster

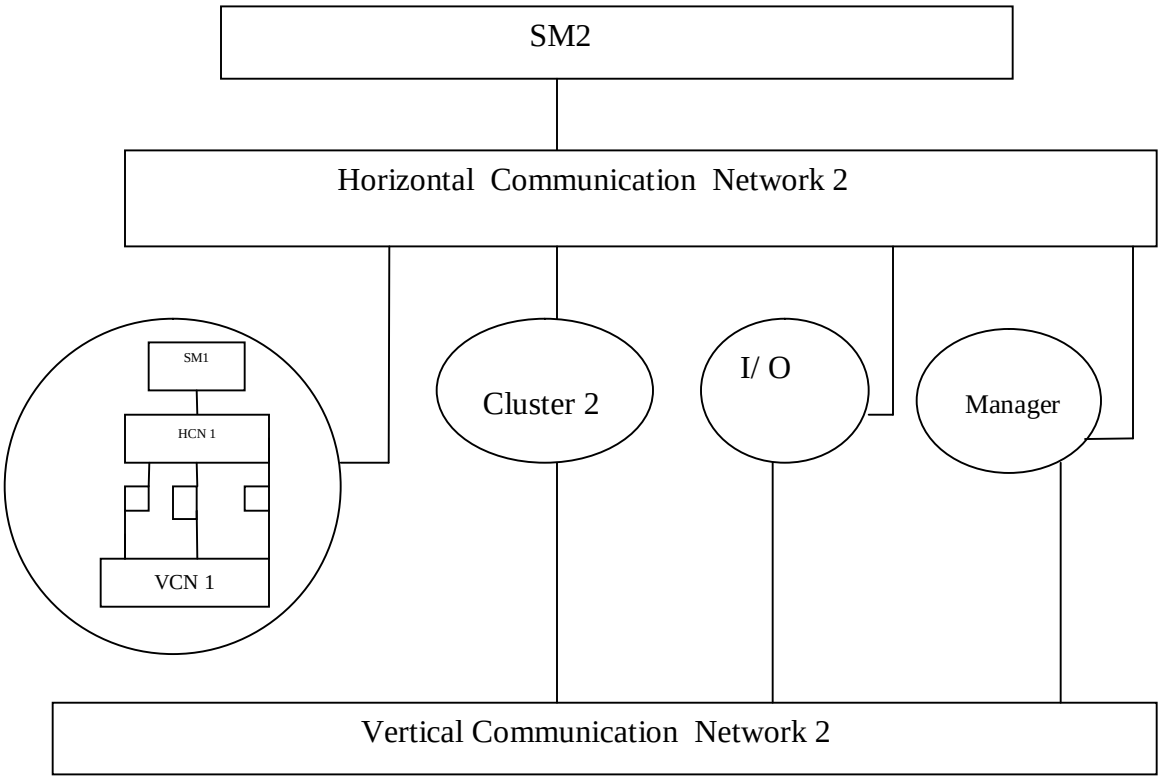

Figure **3:** Two Stage System

This method can expand the system vertically and constructing s-stages system. A cluster in i^{th} stage of the s-stages system is depicted in figure 4. Here cluster include some processing clusters or PC namely, one I/O cluster and one managing cluster. There are two interconnection networks, HCN_i and VCN_i that transmitting data inside and outside of the clusters respectively. Such systems can be expanded vertically by increasing the number of stages or horizontally by increasing the number of PCs in each level. In multistage clustering structure based system, if there are number of PCs that make a cluster will be equal for all clusters of i^{th} stage, the system is known as homogenous at level i. If system is homogenous in all level it will be called homogenous on the other hand if it will not be homogenous at least in one stage, it will be recognized as non homogenous or heterogeneous.

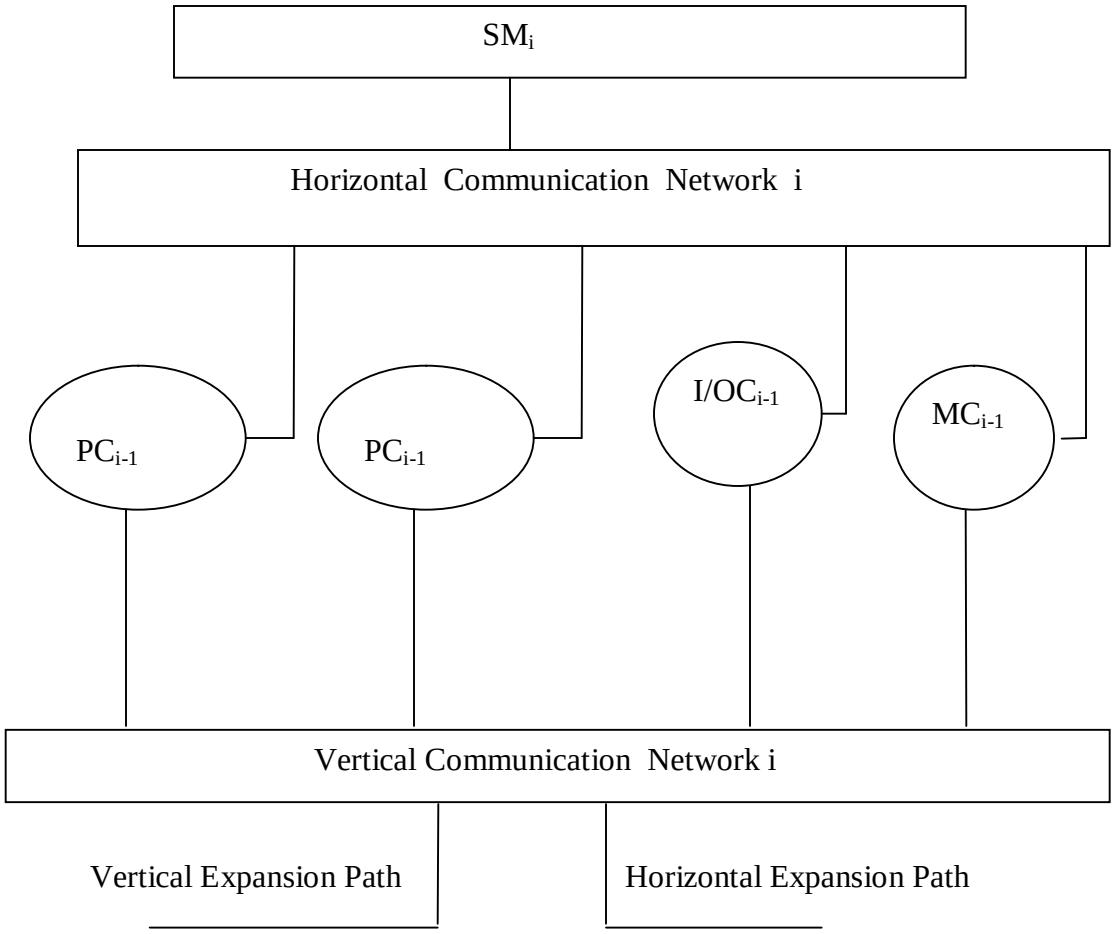

Figure **4:** Cluster in i^{th} stage of s stage system

3. THE PERFORMANCE ANALYSIS

For evaluating the performance of the system, let consider the system is constructed based on homogenous MSCS. In this system any processor performs a piece of the main program that is called processor's job. During the job execution, it is probable that a job needs to communicate with the other jobs. Therefore several queues can be constructed for each interconnection networks and shared memories.

Consider the following assumptions for analyzing the system.

- C_i is the number of PC_S in i^{th} stage of system and C_o is the number of processors in each basic cluster.
- Processors itself generated the inter job communication requests.
- The time between two consecutive requests have exponentially distributed with parameter λ.
- Access time to memory in i^{th} stage has exponentially distributed with parameter μ_{mi}.
- The destination of each request will be uniformly distributed between processor's

jobs and the probability of outgoing request from i^{th} stage is denoted by P_i.

- The service time of the inter connection networks in i^{th} stage have exponentially distributed with parameter μ_{hi} and μ_{vi} for HCN_i and VCN_i, respectively.
- Conflict over memory modules and interconnection networks will be resolved by the queueing center which is modeled as M/G/1.
- Request processors must be waited until they offer service as per above scheme and during waiting period, they can not generate any other request.

The parallel processing system in which the input rate of each stage must be computed and queueing problem is analyzed by developing the M/G/1 model. For analyzing the design of MPP's with a large number of units, the area of computation for closed queueing network will be very large. Apply queueing network methodology for analyzing the closed queueing network and also determine the input rate of each service center as a function of the input rate for previous center. This technique can reduce the calculation and simulation time.

As shown in the figure 5, all the request departs from HCN_i will pass through the SM_i with probability one. Therefore, compute input request rate of VCN_s and HCN_s. The processor requests will be directed to service center HCN_1 and VCN_1 by probability $(1-P_1)$ and P_1, respectively. If the request rate of a processor will be λ, the input rate of HCN_1 and VCN_1 that originated from that process will be $\lambda(1-P_1)$ and λP_1.

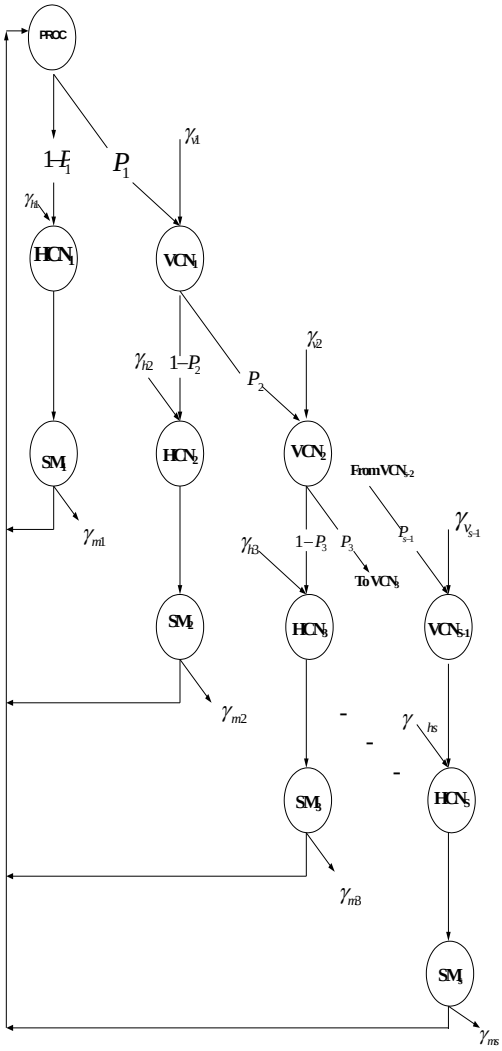

Figure **5:** Multi stage Cluster MPP's with s stage system

Since there are (C_0-1) processors in each basic cluster, the requests that receive to HCN_1 and VCN_1 originating from other processor in the same cluster, indicated by γ_{h1} and γ_{v1}, will be $\lambda(1-P_1)(C_0-1)$ and $\lambda P_1(C_0-1)$, respectively. So the total requests of the processors that received to service centers in the first stage can be computed by following equations:

$$\lambda_{v1} = P_1 \lambda + (C_0 - 1) P_1 \lambda = C_0 P_1 \lambda \quad (1)$$

$$\lambda_{m1} = \lambda_{h1} = (1-P_1)\lambda + (C_0 - 1)(1-P_1)\lambda = C_0 (1-P_1)\lambda \quad (2)$$

The input request rate at the i^{th} stage from each PCs is $\lambda_{(vi-1)}$

$$\lambda_{vi} = P_i \lambda_{v(i-1)} + (C_{i-1} - 1) P_i \lambda_{v(i-1)} = C_{i-1} P_i \lambda_{v(i-1)} \quad (3)$$

$$\lambda_{mi} = \lambda_{hi} = (1-P_i)\lambda_{v(i-1)} + (C_{i-1} - 1)(1-P_i)\lambda_{v(i-1)} = C_{i-1}(1-P_i)\lambda_{v(i-1)} \quad (4)$$

In the last stage there is no request for outer cluster, so that

$$\lambda_{vs} = 0 \quad (5)$$

$$\lambda_{ms} = \lambda_{hs} = C_{s-1}(1-P_s)\lambda_{v(s-1)} + C_{s-1} P_s \lambda_{v(s-1)} = C_{s-1}\lambda_{v(s-1)} \quad (6)$$

Now consider M/G/1 model to calculate the queue length at each mode for all stages, then the average of total waited processors in the system can be computed as.

By using Pollaczek-Khintchine formula, it give

$$L = \rho + \frac{\rho^2 + \lambda^2 \sigma^2_s}{2(1-\rho)} \quad (7)$$

The waited processors would not be able to generate the request. In this situation the effective processor's request rate would be lower than the required. The effective request rate will be decreased with the same ratio as there are active processor's in the system. L and λ have been calculated successively till their changes in two consecutive steps will be negligible. After calculating the effective request rate, the waiting time can be determine by Little formula as

$$\left[L = \lambda W \text{ or } W = \frac{L}{\lambda} = \frac{1}{\lambda}\left[\rho + \frac{\rho^2 + \lambda^2\sigma^2_s}{2(1-\rho)}\right] = \frac{2\rho - \rho^2 + \lambda^2\sigma^2_s}{2\lambda(1-\rho)} \right] \quad (8)$$

Here $P_{v_i}, P_{m_i}, P_{h_i}$ are the probabilities that referred to a processor request to $VCN_i, SM_i \,\&\, HCN_i$ respectively and computed by the following product type solution

$$P_{v_i} = \prod_{j=0}^{i-1} P_{j+1} \quad (9)$$

$$P_{m_i} = P_{h_i} = \frac{(1-P_i)}{P_i} \prod_{j=0}^{i-1} P_{j+1} \quad (10)$$

By determining the average waiting time of a processor for each communication request, which can determine the processor utilization as by using:

Processor Utilization = PU = $$\frac{1}{\lambda w} = \frac{2(\mu - \lambda)}{2\mu - \lambda + \lambda\mu^2\sigma^2_s} \quad (11)$$

Total processing power of the system (TPP), is obtained by considering the single processor power (SPP). Thus

TPP = N x PU x SPP = $$\frac{2(\mu-\lambda)SPP}{2\mu - \lambda + \lambda\mu^2\sigma^2_s} \prod_{i=0}^{s} C_i \quad (12)$$

4. CONCLUSIONS

In this investigation, the performance modeling of a parallel processing system as a sequence of stages, each of which requires a certain integral number of processors for a certain integral of time. This proposed a new structure and

developed an analytical model for massive parallel processing system based on queueing theory. The system performance metrics may provide insights to the system designers and decision makers to improve the system at optimal cost.

REFERENCES

1. Al-Saqabi, K., Sarwar, S. and Saleh, K.: Distributed gang scheduling in networks of heterogeneous workstations, J. Computer Communications, Vol. 20, No. 5, pp. 338-348. (1997)
2. Guan, H. and Cheung, To-Yat. : Efficient approaches for constructing a massively parallel processing system, J. of systems Architecture, Vol. 46, No. 13, pp. 1185-1190. (2000)
3. Hayes, J. P. : Computer Architecture and organization, McGraw-Hill. (1998)
4. Hwang, K. H. and Xu, .Z.: Scalable parallel computing, McGraw-Hill (1998)
5. Jean-Marie, A., Lefebvre-Barbaroux, S. and Liu, Z.: An analytical approach to the performance evaluation of master-slave computational models, J. Parallel Computing, Vol. 24, No. 5-6, pp. 841-862. (1998)
6. Jozwiak, L., Jan, Y.: Design of massively parallel hardware multi-processors for highly demanding embedded applications, J of Microprocessors and Microsystems, Vol. 37, pp. 1155-1172. (2013)
7. Jan, Y. and Jozwiak, L. : Scalable communication architectures for massively parallel hardware multi processors, J of Parallel Distribution Computing, Vol. 72, pp. 1450-1463. (2012)
8. Kotsis, G.: A systematic approach for workload modeling for parallel processing systems, J. Parallel computing, Vol. 22, No. 13, pp. 1771-1787. (1997)
9. Keleinrock, L.: Queueing Systems, Vol. II, Computer Application, New York Wiely. (1975)
10. Mohapatra, P., Das, C. R. and Feng, T. Y.: Performance analysis of cluster based multiprocessor, IEEE Trans. on Computer, Vol. 43, pp. 109-114. (1994)
11. Maheshwari, P. and Shen, H.: An efficient clustering algorithm for partitioning parallel programs, J. Parallel Computing, Vol. 24, No. 5-6, pp. 893-909. (1998)
12. Nassar, H.: A Markov model for multibus multiprocessor systems under asynchronous operation, J. Information Processing Letters, Vol. 54, No. 1, pp. 11-16. (1995)
13. Reijns, G. L. and Gemund, Van. J. C.: Analysis of a shared- memory multiprocessor via a novel queueing model, J of system Architecture, Vol. 45, No. 14, pp. 1189-1193. (1999)
14. Tomic, D.: Spectral performance evaluation of parallel processing systems, J. Parallel computing, Vol. 13, No. 1, pp. 25-38. (2002)
15. Wasserman, K. M., Michailidis G. and Bambos, N.: Optimal processor allocation to differentiated job flows, J. Performance Evaluation, Vol. 63, No. 1, pp. 1-14. (2006)