Multi-Criteria Scheduling of Complex Workloads on Distributed Resources

Georgios L. Stavrinides
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
gstavrin@csd.auth.gr

Helen D. Karatza
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
karatza@csd.auth.gr

***Abstract*—The emergence of large-scale distributed platforms, such as cloud and fog environments, as well as the increasing complexity of the workloads processed on such resources, have made imperative the utilization of effective scheduling techniques. In this paper, we study and analyze multi-criteria scheduling of complex workloads on distributed resources. The workloads consist of multiple-task applications with different characteristics. Simulation is employed in order to evaluate the system performance. The simulation results reveal that the performance of the utilized scheduling techniques is depended on the employed routing strategies and the variability of the processor service times.**

Keywords—complex workloads, multi-criteria scheduling, distributed resources, performance.

I. Introduction

Cloud and fog computing, along with the rapid expansion of the Internet of Things, have contributed to an unprecedented growth of complex workloads that require a specific level of Quality of Service (QoS) within specific time constraints. The end users typically require lower turnaround times and fair service. Consequently, one of the major challenges is to effectively schedule multi-task jobs on such large-scale distributed resources. To achieve this, it is imperative to devise and utilize suitable workload scheduling strategies.

A plethora of scheduling techniques has been proposed in the literature in order to optimize the performance of various types of distributed systems [1-5]. In the investigated scenarios, the component tasks of the workload could be independent, while in other cases the constituent tasks could form linear workflows. A well-known class of complex jobs is *Bag-of-Tasks (BoT)* [6-8]. Jobs of the BoT type consist of tasks that are independent from each other.

Another multi-task job type is *Task-Chains* [9-11]. A task-chain comprises tasks with restrictions in their order of execution. Moreover, in order to leverage data locality, these tasks should be processed on the same processor. A class of jobs with more complex structure is *Bag-of-Task-Chains (BoTCs)* [12]. Each BoTC job consists of independent task-chains. Workloads consisting of BoTCs and single-task jobs or BoT jobs have been studied in [13] and [14], respectively.

Workflow scheduling has also been extensively studied in the research literature [3], [5]. There is a diversity of workflow job types, which are typically represented as directed acyclic graphs. A type of this class of jobs is linear workflows, where there is only one path from the first node (entry task) to the last one (exit task). In this work, we investigate scheduling of a complex workload which is a combination of BoTs and *Bag-of-Linear-Workflows (BoLW)* jobs on distributed resources.

Along with the 2 Random Choices routing policy, proposed by Mitzenmacher in [15], we also employ another routing technique, which is a combination of the 2 Random Choices and the probabilistic policy. We propose a novel multi-criteria scheduling technique. The proposed approach considers the computational demand of each BoT task and each linear workflow, the type of each job, and the job's arrival time, where the job priority is increasing with time at specific time steps. The proposed strategy is compared with the FCFS policy.

The objective of this research is to analyze the performance of the scheduling policies in different cases of routing algorithms, and to also examine the effect of service time variability on the system performance. Extensive simulation experiments are used for the performance evaluation.

Previous research works that examined BoTs, task-chains, BoTCs and combinations of them considered different workload models. Furthermore, they did not use the routing techniques and the multi-criteria scheduling policy that have been employed in this work. To the best of our knowledge, routing and scheduling of bags-of-tasks and bags-of-linear-workflows, in the context examined in this work, have not appeared in the literature before.

The rest of the paper is organized in the following order: Section II presents the distributed resources model and the workload. Section III describes the routing and scheduling algorithms. Section IV presents the performance parameters. Section V provides the experimental setup and analyzes the simulation results. Section VI concludes the paper and presents the goals of our future work.

978-1-7281-6544-8/20/$31.00 ©2020 IEEE

II. Problem Definition

A. System and Workload Models

The queueing network model of the distributed resources under study consists of P = 16 distributed processors, as shown in Fig. 1. Upon each job arrival, tasks of BoTs and linear workflows of BoLWs are routed to processor queues, according to the routing policy utilized in each case.

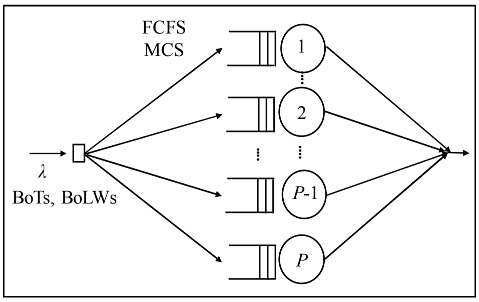
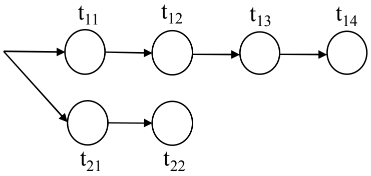


Fig. 1. The queueing model of the framework under study.

A BoT job J_1 consists of N_{J_1} independent tasks. A BoLW job J_2 consists of m linear workflows: LW$_1$, LW$_2$,…,LW$_m$. If t_i is the number of tasks in LW$_i$, i = 1,…, m, then the total number of tasks N_{J_2} of job J_2 is:

$$N_{J_2} = \sum_{i=1}^{m} t_i \qquad (1)$$

Fig. 2 shows an example of a BoLW job where it holds that m = 2, t_1 = 4 and t_2 = 2.



A Bag of two Linear Workflows

Fig. 2. A Bag-of-Linear-Workflows.

The following random variables characterize the workload:

- Number of tasks per BoT.
- Number of linear workflows per BoLW.
- Number of tasks per linear workflow.
- Service demand of BoT and BoLW tasks.
- Interarrival time of the jobs.

The following distributions are utilized for the above random variables:

1) Distribution of the Number of Tasks per BoT: The number of component tasks in a bag-of-tasks job is an integer uniformly distributed in the interval [1, 4]. Therefore, the average number of tasks in a BoT job is η_1 = 2.5.

2) Distribution of the Number of LWs per BoLW: The number of linear workflows in a BoLW job is an integer uniformly distributed in the range [1, 4]. Therefore, the average number of linear workflows in a BoLW job is η_2 = 2.5.

3) Distribution of the Number of Tasks per Linear Workflow: The number of tasks in a linear workflow is an integer uniformly distributed in the interval [2, 4]. Therefore, the average number of tasks in a linear workflow is η_3 = 3.

4) Distribution of Task Service Demand: The tasks of BoTs and the tasks of BoLWs have the same service requirements. The service time of tasks is a hyperexponential random variable, with coefficient of variation CV. The mean task service time is $1/\mu$.

5) Distribution of Interarrival Time of the Jobs: It is considered that both types of jobs, BoTs and BoLWs, arrive in a single Poisson stream, with rate λ. Consequently, the interarrival time of the jobs is an exponential random variable with mean 1/λ. We consider that a newly arrived job is a BoT with probability Pr_{BoT}.

The average number of tasks per BoLW job is $\eta = \eta_2 \times \eta_3$ = 7.5. Therefore, the average number of tasks of a BoT is 3 times smaller than the mean number of tasks in a BoLW. For this reason, we consider that:

$$\eta_1 \times Pr_{BoT} = (\eta_2 \times \eta_3) \times (1\text{-}\ Pr_{BoT}) \qquad (2)$$

Therefore, it holds that Pr_{BoT} = 0.75. Table I includes the parameters of the system and workload models studied in this work.

TABLE I. System and Workload Parameters

P	Number of processors
η_1	Mean number of tasks in a BoT
η_2	Mean number of linear workflows in a BoLW
η_3	Mean number of tasks in a LW
η	Mean number of tasks in a BoLW
$1/\lambda$	Mean job interarrival time
$1/\mu$	Mean service time of BoT and BoLW tasks
CV	Service time coefficient of variation
Pr_{BoT}	Probability for a newly arrived job to be a BoT

III. Routing and Scheduling Techniques

A. Routing Policies

In this paper, we employ the 2 Random Choices (2RC) approach [15] and two variations of it:

1) 2RC-1/1: According to this approach, every task of a bag-of-task and every linear workflow is assigned to the shortest processor queue between two randomly selected ones (2RC). This technique is employed in all cases of routing.

2) 2RC-1/2: In every two cases of routing, one time the 2RC policy is employed and the other time a probabilistic policy.

3) 2RC-1/4: In every four cases of routing, one time the 2RC policy is employed and three times a probabilistic policy.

B. Scheduling Strategies

1) First-Come-First-Served (FCFS): Bag-of-tasks component tasks and linear workflows are executed in the same order as the one they arrived at the system. This is the simplest scheduling algorithm. Its implementation does not involve any overhead, since it does not rearrange the processor queues at each job arrival. It is also the fairest of all other methods.

2) Multi-Criteria Scheduling (MCS): According to this policy, the queues of the processors are rearranged at time steps based on three criteria: α, β, γ, defined as follows:

α: depends on the type of job J (BoT or BoLW),

β: is based on service time s of BoT tasks and cumulative service time of linear workflows,

γ: depends on the number of time steps (*nts*) since job's J arrival time *timear*(J), where at time *clock*, *nts* is evaluated as the integer part of the following fraction:

$$nts = \text{int}\ ((clock - timear(J)) \ / \ \text{step}) \tag{3}$$

At each time step, the scheduling algorithm re-determines the priorities of all of the bag-of-tasks component tasks and linear workflows. In each processor queue, the scheduler gives priority to the BoT task or linear workflow that has the smallest value of the weighted-service-time:

$$Weighted\text{-}service\text{-}time = s \times (1 \ / \ (1+(nts \times weight\ )) \times \alpha \tag{4}$$

where *weight* = 1 and $\alpha = 3$ in the case of a task of a BoT, whereas $\alpha = 1$ in the case of a linear workflow of a BoLW.

We chose the above values for α because BoLW jobs have a number of tasks three times larger than BoTs and also because there are three times more BoTs than BoLWs in the workflow. Therefore, with this α, BoT tasks do not cause large delays to linear workflows, as they would in case the shortest time was the only criterion considered. It is also obvious that with this policy, the larger the number of time steps since a job's arrival is, the highest the priority of the job's tasks or linear workflows to be scheduled for execution, even if their service time is larger than the service time of other tasks or linear workflows of jobs that arrived at a later time step.

IV. Performance Evaluation Parameters

Table II includes all of the performance parameters used in our simulation experiments. The response time of a BoT or a BoLW is the time interval between the arrival of the job and the completion of all of its component tasks or linear workflows, respectively. In this paper, we weigh each job's J_i response time r_i with its size (number of tasks) N_{J_i}. Thereby, jobs with the same response time, but with different size, have different impact on performance. The average weighted response time *WRT* of n jobs is defined as:

$$WRT = \frac{\sum_{i=1}^{n} N_{J_i} \times r_i}{\sum_{i=1}^{n} N_{J_i}} \tag{5}$$

In this paper, the fairness in job execution is indicated by the *Maximum WRT* (*MWRT)* which is defined in Table II.

TABLE II. Performance Parameters

WRT	Average weighted response time of BoTs and BoLWs
D_{WRT}	Relative (%) decrease in *WRT* when the MCS approach is employed, compared to FCFS
MWRT	Average maximum weighted job response time
MWRT-Ratio	The ratio of *MWRT* in the case where MCS is utilized, divided by *MWRT* in the case where FCFS is used

V. Simulation Experiments & Results

A. Experimental Setup

The proposed approach was studied and analyzed by extensive simulations with synthetic datasets, based on the independent replications method. The calculated confidence intervals had half-widths that were smaller than 5% of their respective mean values. The simulation input parameters used in our experiments are included in Table III.

TABLE III. Input Parameters

$1/\lambda$	0.28
$1/\mu$	1
CV	2, 3
time step	10

In the framework under study, each bag-of-task consists on average of η_1= 2.5 tasks and each BoLW job consists of η = 7.5. Therefore, taking into account that the Pr_{BoT} is 0.75, the average number of tasks of all jobs (BoT and BoLW) is 3.75. In the case where all of the processors in the system are busy, an average of $P/3.75 = 4.27$ jobs can be served at each unit of time. Hence, in order for the system to be stable and the processor queues not to be overloaded, a λ should be chosen such that $\lambda < 4.27$. Consequently, the mean interarrival time $1/\lambda$ should be:

$$1/\lambda > 0.23 \tag{6}$$

For the above reasons, we chose a mean interarrival time equal to $1/\lambda = 0.28$. A time step equal to 10 units of time was selected, since during this step size several jobs arrived at the system.

B. Simulation Results

In all of the examined cases the mean processor utilization U is close to 0.84. Regarding *WRT,* Fig. 3, 4, 6 and 7 show that the MCS policy performs better than FCFS. This is because when FCFS is used, there are cases where bag-of-tasks component tasks and linear workflows that have a small total service time experience longer delays, waiting behind other larger BoT tasks and/ or large linear workflows in the queues.

The simulation results also reveal that *WRT* depends on the routing technique employed and that it increases with increasing number of times that the probabilistic policy is employed as compared to the 2 Random Choices policy. That is, the 2RC-1/1 routing technique performs better than the 2RC-1/2, which in turn performs better than the 2RC-1/4.

We can also observe that the difference in performance between the 2 scheduling policies (D_{WRT}) is greater in the case of 2RC-1/4, compared to the case where 2RC-1/2, which is more significant than in the case of 2RC-1/1. This is due to the fact that the more efficient the routing technique is, the less important the role of the employed scheduling policy.

In Fig. 4 and 7 we can also observe that the superiority of MCS policy over FCFS is more significant when $CV = 3$, compared to the case where $CV = 2$. This is due to the fact that service times present larger variability for larger *CV* values. Therefore, the advantages of MCS can be leveraged at a greater degree for a larger coefficient of variation.

We should mention that when a policy gives priority to the shortest bag-of-tasks component task or the shortest linear workflow in a queue, this does not mean that the corresponding job will finish execution earlier. This is due to the fact that some other tasks or linear workflows of the same job may be delayed longer in other queues, which in turn will result in larger synchronization delays. This is the reason that in this paper we use the MCS method which depends not only on service times, but also on how long the job has been waiting in a queue, since it prevents some large tasks/ linear workflows to experience very long delays. Furthermore, this method considers the characteristics of the jobs so that tasks of BoTs do not overcome linear workflows very frequently.

Fig. 5 and 8 illustrate that *MWRT-Ratio* is larger in the MCS case. This is due to queue rearrangements, which have as a result some large BoT tasks and/ or linear workflows to be delayed longer in the processor queues. Like D_{WRT}, also *MWRT-Ratio* is larger in the routing case 2RC-1/4, than in the case of 2RC-1/2, which in turn is more significant than in the case of 2RC-1/1. This is due to the fact that in a distributed system the more efficient the load balancing algorithm is, the smallest the probability for tasks of BoTs or linear workflows to be blocked in overloaded processors.

The simulation results presented in Fig. 5 and 8 also show that for each routing policy, *MWRT-Ratio* is almost equal in both *CV* cases. It should be mentioned that when the hyperexponential distribution is used, some very large service times occur, compared to the mean service time, which are larger for larger *CVs*. In each *CV* case, these very large service times affect the value of *MWRT* in both cases of scheduling algorithms.

VI. Conclusions & Future Work

We studied multi-criteria scheduling of BoTs and BoLWs on distributed resources. The simulation results revealed that the proposed MCS policy performed better than FCFS. MCS's impact on performance was more significant in the cases of the less efficient routing techniques, and also in the case of larger variability in service demands. The average maximum weighted response time, a metric that in this work indicated fairness, was greater in the MCS case, as opposed to FCFS. It was also larger in the cases where less efficient routing techniques were utilized. Our future work plans include studying MCS under different criteria and workloads.

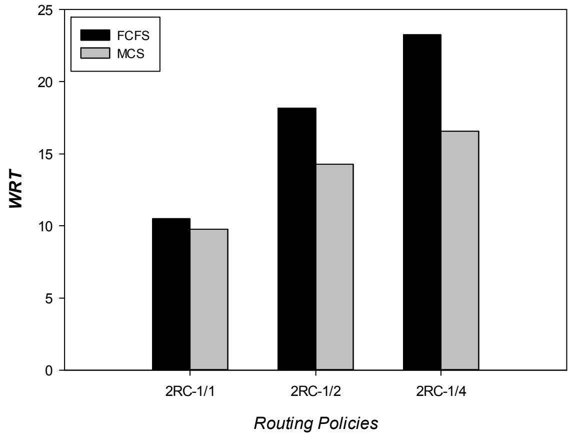


Fig. 3. *WRT* vs. Routing Policy in the $CV = 2$ case.

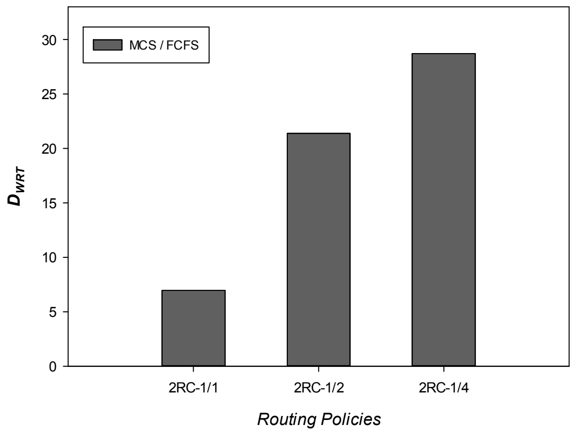


Fig. 4. D_{WRT} vs. Routing Policy in the $CV = 2$ case.

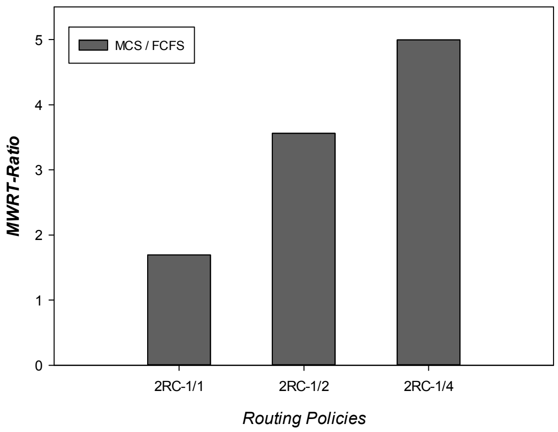


Fig. 5. *MWRT*-Ratio vs. Routing Policy in the $CV = 2$ case.

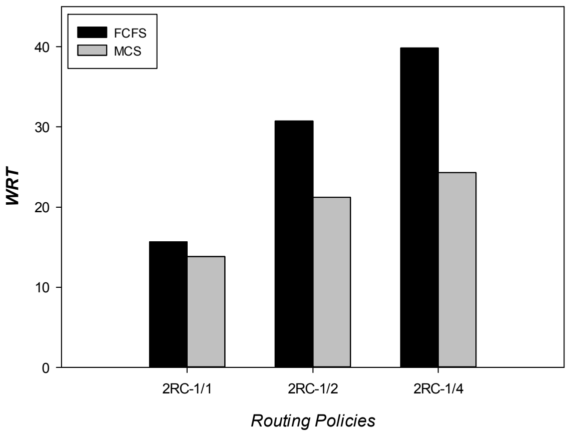


Fig. 6. *WRT* vs. Routing Policy in the *CV* = 3 case.

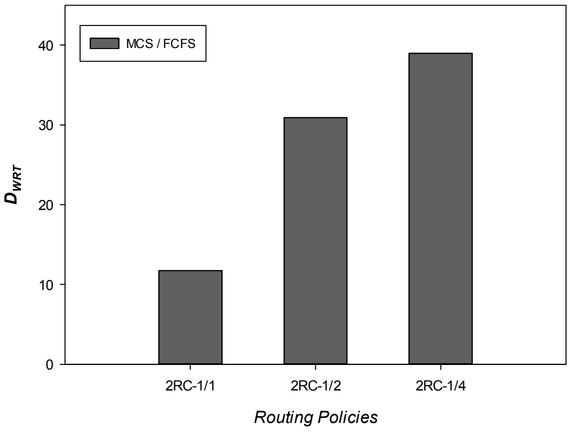


Fig. 7. D_{WRT} vs. Routing Policy in the *CV* = 3 case.

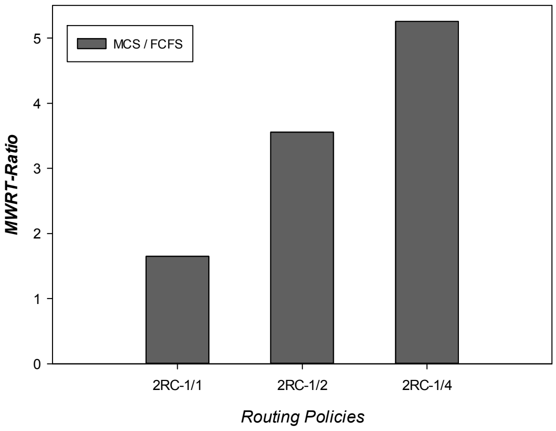


Fig. 8. *MWRT*-Ratio vs. Routing Policy in the *CV* = 3 case.

REFERENCES

[1] Z. C. Papazachos and H. D. Karatza, “Gang scheduling in multi-core clusters implementing migrations,” *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1153-1165, Oct. 2011.

[2] G. L. Stavrinides and H. D. Karatza, “Scheduling real-time jobs in distributed systems - simulation and performance analysis,” in *Proceedings of the First International Workshop on Sustainable Ultrascale Computing Systems (NESUS’14)*, Porto, Portugal, Aug. 2014, pp. 13-18.

[3] G. L. Stavrinides and H. D. Karatza, “Cost-aware cloud bursting in a fog-cloud environment with real-time workflow applications,” *Concurrency and Computation: Practice and Experience*, e5850, Jun. 2020.

[4] Z. C. Papazachos and H. D. Karatza, “The impact of task service time variability on gang scheduling performance in a two-cluster system,” *Simulation Modelling Practice and Theory*, vol. 17, no. 7, pp. 1276-1289, Aug. 2009.

[5] G. L. Stavrinides and H. D. Karatza, “Orchestration of real-time workflows with varying input data locality in a heterogeneous fog environment,” in *Proceedings of the Fifth International Conference on Fog and Mobile Edge Computing (FMEC’20)*, Paris, France, Jun. 2020, pp. 202-209.

[6] D. Tychalas and H. Karatza, “A scheduling algorithm for a fog computing system with bag-of-tasks jobs: simulation and performance evaluation,” *Simulation Modelling Practice and Theory*, vol. 98, 101982, Jan. 2020.

[7] Z. C. Papazachos and H. D. Karatza, “Scheduling bags of tasks and gangs in a distributed system,” in *Proceedings of the 2015 International Conference on Computer, Information and Telecommunication Systems (CITS’15)*, Gijón, Spain, Jul. 2015, pp. 1-5.

[8] G. L. Stavrinides and H. D. Karatza, “Dynamic scheduling of bags-of-tasks with sensitive input data and end-to-end deadlines in a hybrid cloud,” *Multimedia Tools and Applications*, May 2020.

[9] J. Du, J. Y. T. Leung, and G. H. Young, “Scheduling chain-structured tasks to minimize makespan and mean flow time,” *Information and Computation*, vol. 92, no. 2, pp. 219-236, Jun. 1991.

[10] J. Schlatow and R. Ernst, “Response-time analysis for task chains in communicating threads,” in *Proceedings of the 2016 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS’16)*, Vienna, Austria, Apr. 2016, pp. 1-10.

[11] M. Ait Aba, L. Zaourar, and A. Munier, “Approximation algorithm for scheduling a chain of tasks on heterogeneous systems,” in *Proceedings of the 23rd International European Conference on Parallel and Distributed Computing (Euro-Par’17), Parallel Processing Workshops*, Santiago de Compostela, Spain, Aug. 2017, pp. 353-365.

[12] G. L. Stavrinides and H. D. Karatza, “Scheduling bag-of-task-chains in distributed systems,” in *Proceedings of the 14th IEEE International Symposium on Autonomous Decentralized Systems (ISADS’19)*, Utrecht, The Netherlands, Apr. 2019, pp. 81-86.

[13] G. L. Stavrinides and H. D. Karatza, “Scheduling single-task jobs along with bag-of-task-chains in distributed systems,” in *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems (ICFNDS’19)*, Paris, France, Jul. 2019, pp. 32:1-32:6.

[14] G. L. Stavrinides and H. D. Karatza, “Scheduling a job mix of bag-of-tasks and bag-of-task-chains on distributed resources,” in *Proceedings of the 11th International Conference on Information and Communication Systems (ICICS’20)*, Irbid, Jordan, Apr. 2020, pp. 394-399.

[15] M. Mitzenmacher, “The power of two choices in randomized load balancing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1094-1104, Oct. 2001.