Multi-Cluster Performance Impact on the Multiple-Job Co-Allocation Scheduling

Héctor Blanco, Eloi Gabaldón, Fernando Guirado, Josep Lluís Lérida,

Abstract—
Multi-cluster environments are composed of multiple clusters of computers that act collaboratively, and thus allow computational problems to be treated that require more resources than those available in a single cluster to be dealt with. However, the complexity of the scheduling process is greatly increased by the heterogeneity of resources and the co-allocation process, which distributes the tasks of parallel jobs across cluster boundaries. In a previous work, the authors presented a new scheduling strategy made up of a job selection function and a linear programming model to find the best simultaneously allocation for multiple jobs from the system queue on a heterogeneous multi-cluster, by applying co-allocation when necessary.

In this paper the effectiveness of our proposed scheduling strategy is evaluated under multiple configurations for the multi-cluster environment (computation heterogeneity and network availability) and compared with other co-allocation strategies from the literature. The results showed that co-allocation has a negative effect on the response times when the network availability is low. On the other hand, the use of the multiple-job allocation contributes to maximize the multi-cluster resources usage. By this, our strategy was able to adapt to different multi-cluster configurations obtaining better scheduling decisions than the other schedulers.

Keywords—**Job Scheduling, Multi-Cluster Heterogeneity and Performance, Co-Allocation, Mixed Integer Programming**

I. Introduction

Computation problems that require more computational resources than those offered by a single just cluster can be resolved by the use of multiple clusters in a collaborative manner. These environments, known as multi-clusters, are distinguished from grids by their use of dedicated interconnection networks with a known topology and more predictable performance [1].

A critical aspect of exploiting the resources in a multi-cluster environment is the challenge of scheduling parallel jobs across different clusters [2]. This allocation strategy, known as co-allocation, can maximize the job throughput by reducing the queue waiting times, and thus, jobs that would otherwise wait in the queue for local resources can begin their execution earlier, improving system utilization and reducing average queue waiting time [2][3]. However, mapping jobs across the cluster boundaries can result in rather poor overall performance when co-allocated jobs contend for inter-cluster network bandwidth. Additionally, the heterogeneity of processing and communication resources increases the complexity of the scheduling problem [4][5].

The scheduling strategies with co-allocation in multi-cluster environments have generated great interest in recent years. The performance of different scheduling strategies using co-allocation based on job queues was analyzed in [2]. This work concluded that unrestricted co-allocation is not recommended and limiting the component sizes of the co-allocated jobs improves performance. Some other studies dealt with co-allocation by developing load-balancing techniques [6][7], selecting the most powerful processors [8] or minimizing the inter-cluster links usage [4]. These studies share the optimization of a single performance metric, such as the computing capability or the communication links usage, without finding a compromise between these. In order to fill this gap, a new analytical model was presented in [9] that measures the execution time of parallel applications by considering the resource availability of both processors and communication resources.

A common issue in those previous works is that jobs are allocated individually, considering all the available resources for them without taking other jobs in the waiting queue into account. In order to solve this problem, in [3] we presented a new scheduling strategy, named *PAS* for Package Allocation Strategy. This strategy selects those jobs from the waiting queue that can be concurrently executed with the available resources. Once the package of jobs is selected, a Mixed-Integer Programming model (MIP) is responsible for finding the best possible resource allocation for them. The *PAS* strategy was evaluated for a predetermined multi-cluster environment with different kinds of workload. The results show that applying the *PAS* strategy, the response times were lower compared with the most common scheduling strategies in the literature, making better use of the resources and preventing the saturation of the inter-cluster links.

An important result observed in [3] is that the performance of scheduling strategies is very sensitive to the availability of resources. In a multi-cluster made up of different clusters with heterogeneous and non-dedicated resources, the availability of resources and their capabilities are decisive for the performance of the scheduling strategies. Given this, it is necessary to conduct a more detailed analysis to determine the effect of the structure of the multi-cluster environment and the availability of resources on the effectiveness of the previously-tested scheduling strategies.

In the present paper, we evaluated thoroughly the multi-cluster configuration and resource availability on the effectiveness of different scheduling strategies. We assess the parallel application performance obtained for a fixed workload by varying the processor heterogeneity between clusters and the available bandwidth on the communication links. We con-

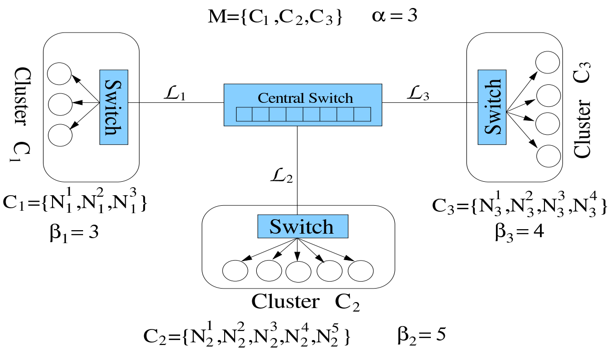

Fig. 1. Diagram of a multi-cluster topology

ducted this evaluation using the Package Allocation Strategy (PAS) that we proposed in [3] and comparing the results with those obtained for the scheduling strategies presented by Jones [4] and Naik [8].

The rest of the paper is organized as follows. In Section II, the multi-cluster and parallel application performance model used in this paper is presented. In Section III, we present our scheduling strategy for multiple-job co-allocation in a multi-cluster environment. Section IV shows the experimentation results obtained from comparison with other scheduling strategies in the literature by varying the heterogeneity and availability of the multi-cluster resources. Finally, the conclusions are presented in Section V.

II. Multi-cluster Model

Advances in computational and communication technologies have made it economically feasible to conglomerate multiple clusters of heterogeneous networked resources leading to the development of large-scale distributed systems known as multi-cluster systems. Generally, multi-cluster systems can be classified into super-clusters and cluster-of-clusters. A good example of super-cluster systems is DAS-2 [10], which is characterized by a large number of homogeneous processors and heterogeneity in communication networks. In contrast, cluster-of-clusters are constructed by interconnecting multiple single cluster systems. Thus heterogeneity may be observed in communication networks as well as processors. The LLNL [11] multi-cluster system, which is built by interconnecting of four single clusters is an example of cluster-of-clusters system.

A commonly used model for representing the general structure for multi-cluster systems is presented in Figure 1. The system is made up of as a collection of arbitrary sized clusters $\{C_1..C_\alpha\}$, each cluster i is composed of N_i processors of type T_i, $i = 1, ..., C$, where T_i could be different for each cluster. Also, Clusters are connected to each other through single-dedicated links $\{\mathcal{L}_\infty..\mathcal{L}_\alpha\}$, by means of a central switch.

In the present work, we focus our discussion on the cluster-of-clusters system where heterogeneity may be observed in both resources processors and communication networks. Thus, we need a model that considers this feature to assess the performance of parallel applications more accurately.

A. Analytic Performance model

In [9] we presented a new performance model for parallel jobs that defined the execution time by considering both the availability and heterogeneity of the processors and communication networks. This model defines the execution time (T_e) of a parallel job in a heterogeneous and non-dedicated environment as its execution time in a dedicated environment ($\overline{T_e}$) delayed by a slowdown factor (SD) produced by the heterogeneity and non-dedicated nature of the slowest allocated resources, and expressed by equation 1.

$$T_e = \overline{T_e} \cdot SD \tag{1}$$

The slowdown of a parallel application depends on the capacity and availability of both processing resources and communication network, and thus, we can express SD based on its processing SP and communication SC slowdowns by equation 2.

$$SD = \sigma \cdot SP + (1 - \sigma) \cdot SC \tag{2}$$

where σ denotes the relevance of the processing time with respect to the communication time of the corresponding job. The details of the calculation of the SP and SC values are presented below.

A.1 Processing Characterization

We assume that parallel job tasks are generally similar in size and executed separately, and thus, the job execution time is bounded by the slowest allocated resource. Taking this into account, the job processing slowdown (SP) is obtained from the allocated resource with maximum processing slowdown, expressed by equation 3.

$$SP^j = max\{SP_r | r \in \mathcal{P}^|\} \tag{3}$$

where $\mathcal{P}^|$ denotes the set of processing nodes allocated to job j. In heterogeneous and non-dedicated environments, the computing resources capabilities can be quite different. To measure these differences, we use the Effective Power metric (Γ_r) defined in [9], which relates the computing power of each resource with its availability. Thus, $\Gamma_r = 1$ when resource r has full capacity to run tasks at full speed, otherwise $\Gamma_r < 1$. Assuming this, the processing slowdown of such resource SP_r is inversely proportional to its Effective Power weight, $SP_r = (\Gamma_r)^{-1}$.

A.2 Communication Characterization

The parallel job co-allocation consumes a certain amount of bandwidth across inter-cluster network links (BW_k^j). These are shown by equation 4.

$$BW_k^j = \left(t_k^j \cdot PNBW^j\right) \cdot \left(\frac{n_T^j - t_k^j}{n_T^j - 1}\right), \quad \forall\, k \in 1..\alpha \tag{4}$$

where n_T^j is the total number of tasks of the job j, t_k^j denotes the total number of tasks allocated to cluster C_k and $PNBW^j$ is the average per-node bandwidth requirement by job j from the jobs. The first term in the equation is the total bandwidth required by all the nodes associated with job j in cluster C_k. The second term represents the communication percentage of job j in other cluster nodes (not in C_k) that will use the inter-cluster link k.

The saturation degree of inter-cluster links relates the available bandwidth of each link (ABW_k) with the bandwidth requirements of the allocated parallel applications, which is calculated by equation 5.

$$BW_k^{sat} = \frac{ABW_k}{\sum_{j,k} BW_k^j} \quad \forall\, k \in 1..\alpha \tag{5}$$

When the required bandwidth is lower than the available, the link is not saturated and the communications will not suffer delays. Otherwise, the network link is saturated, drastically reducing the performance of all the jobs sharing the link. Thus, the job communicating slowdown (SC) is obtained from the slowest, most saturated, communication link used by the job, calculated as the inverse of the saturation bandwidth with equation 6.

$$SC^j = max\{(BW_k^{sat})^{-1} | k \in 1..\alpha\} \tag{6}$$

The goodness of this analytic performance model was its ability to capture the performance for each individual application based on the characteristics and load conditions of the multi-cluster environment. Thus, we can use this slowdown model as a performance metric from parallel application point of view, allowing the scheduler to take best allocation-resource decisions according to stablished criteria.

III. Multiple-Job Co-allocation Strategy

A common feature of most on-line scheduling strategies in cluster, multi-cluster and grid environments is the individual allocation of resources to applications. First, the scheduler selects the next job to be executed according to a priority criterion. When there are insufficient resources to run the selected job, the scheduler can wait for the release of enough resources in order to follow a *First Come First Served* (FCFS) schema, or select a new job from the waiting queue that can be executed with the available resources by applying such a schema as *Fit Processors First Served* (FPFS) or *backfilling*, etc. Once a job is selected, it is individually allocated to the most appropriate resources according to the established criteria.

However, allocating the best available resources to a job without considering the requirements of the rest of the jobs in the waiting queue can impair the performance of future allocations and therefore the overall system performance. With the aim of overcoming this drawback, in [3], we proposed a two-phase scheduling strategy, named *Package Allocation Scheduling* (*PAS*). Firstly, a Job Package Selection function determines those suitable jobs in the waiting queue that will be allocated simultaneously. Secondly, a Mixed-Integer Programming (MIP) model returns the allocation that minimizes their global slowdown while preventing the saturation of the inter-cluster links and applying co-allocation when it is necessary.

A. Job Package Selection

The main aim of the selection function is to package the most suitable jobs from the waiting queue to be executed simultaneously, according to certain criteria established in terms of desired system performance objectives. The job package selection function ($\mathcal{F}$) can be expressed as $PCK = \mathcal{F}(\mathcal{Q}, \mathcal{R}, \mathcal{C})$ where $\mathcal{Q}$ is the set of jobs in the waiting queue, $\mathcal{R}$ the set of multi-cluster resources and $\mathcal{C}$ the criteria to be met by resources to allocate the job package.

There is a wide variety of criteria that can be applied from the point of view of resource utilization or the nature of the parallel applications. In the present work, in order to assess the effects of the multi-cluster configuration and resource availability on the scheduling strategies, we selected the most commonly used criterion which selects the set of jobs in the waiting queue that fits the free multi-cluster resources, and is expressed as

$$\exists\ PCK \subseteq \mathcal{Q} \mid \sum_{| \in \mathcal{PCK}} \tau^{|} \leqslant |\mathcal{R}'|$$

where PCK is the subset of jobs from the waiting queue ($\mathcal{Q}$) whose total number of tasks is less than, or equal to, the multi-cluster resources in $\mathcal{R}'$, which represents the subset of multi-cluster resources ($\mathcal{R}' \subseteq \mathcal{R}$) that meet the criteria ($\mathcal{C}$), which, in our case, are those resources non-assigned to other parallel jobs.

B. MIP Allocation Model

Once a package of waiting jobs was selected, the *PAS* Strategy must allocate the most suitable resources according to established performance criteria. Using the slowdown model expressed by equation 2 as a parallel application performance metric and defining the resource allocating problem as a Mixed-Integer programming model, as in [3], we obtained an allocation model that minimizes the global response time for the target job package.

The goodness of the *PAS* strategy was its ability to obtain the best resources for each parallel job considering the other applications that can be executed concurrently in the multi-cluster environment, and thus reducing the global response times while making better use of the resources and also preventing the saturation of the inter-cluster links.

However, the effectiveness of the scheduling strategies in these dynamic and heterogeneous environments is highly sensitive to the multi-cluster configuration and its availability. Therefore, it is necessary to conduct a thorough analysis of the effect of these parameters on the effectiveness of the scheduling strategies, allowing the scheduler to adjust according to the system characteristics and its status. In the next section, we assess the effects of the resource heterogeneity and availability on the effectiveness of the scheduler decisions.

IV. Experimentation

In this section, we present an experimental evaluation of the impact of the multi-cluster performance on the effectiveness of the co-allocation scheduling strategies in heterogeneous multi-cluster environments. The main goal of this evaluation was to determine the scheduler ability to adapt its decisions to take advantage of the multi-cluster performance characteristics and then obtain the best performance for the proposed workload. For this, we proposed a predetermined workload that was fixed for all the experimentation process, and was scheduled on different multi-cluster configurations.

The experimental study was carried out by modifying two main parameters such as the *Bisection Bandwidth* ($BSBW$) and the degree of heterogeneity (H). The first parameter $BSBW$, measured in Gigabytes/second (GB/s) [1], assessed the effect of the communication rate between two halves of the communication system. It determines the worst-case performance on a particular network, since it is related to the cost of moving data from one side of the system to other. We chose values in the range of 0.1GB/s to 1.1GB/s, representing a low to high transmission cost. This parameter would affect the schedulers to avoid jobs co-allocation when the network added a high communication slowdown. The second parameter, the multi-cluster heterogeneity degree H, meant the percentage of difference of the effective power between the individual clusters. Thus we defined three different configurations, $H = 10\%$ (low heterogeneity, near to a homogeneous multi-cluster), $H = 50\%$ (there were significant differences) and $H = 90\%$ (extreme heterogeneity). The effective power is an important factor for the schedulers because the use of the most powerful resources allows to reduce the job execution times, obtaining free resources as quickly as possible and then reducing the overall waiting times. On the other hand, this parameter is in conflict with the available bandwidth because for obtaining the best computing resources in many cases it is necessary to apply co-allocation.

The experimentation was done with our strategy *Package Allocation Scheduling* (PAS) as a basis, and the results were compared with two other co-allocation strategies from the literature. The first, presented by Jones in [4], named CBS for *Chunk Big Small*, tries to co-allocate a "large chunk" (75% of the job tasks) to a single cluster in an attempt to avoid inter-cluster link saturation. The second, presented by Naik in [8], named JPR for *Job Preferences on Resources*, allocates parallel jobs selecting the most powerful resources, co-allocating them when is needed. Both of these use a FCFS (First Come First Served) scheduling policy, the same that we used in our selection function to compare the different allocation strategies properly. For an accurate comparison, both techniques were implemented by using a MIP model using the *CPLEX* solver package. The multi-cluster environment was made up of 4 clusters composed of 24, 16, 16 and 8 nodes, interconnected by a Gigabit network.

Number of jobs	100 jobs
Avg. exec. time	170 time units
Interarrival time (Li'04)	*weibull* ($\lambda = 82.6$, $k = 0.6$)
Job sizes (Lublin'03)	*gamma* ($\alpha = 4.04$, $\beta = 0.77$), *"power of two" prevalence*
Job actual runtime (Li'04)	*weibull* ($\lambda = 200$, $k = 1$)
Comm. requ. (Jones'05)	$PNBW^j = BSBW\left(\frac{4(n_T^j - 1)}{(n_T^j)^2}\right)$

TABLA I

Workload parameters

A. Workload characterization

A detailed characterization of a super-cluster was presented in [12], where different distribution functions were obtained from the characterization of the behavior of different real cluster environments. Based on the results of this study and also taking into account the considerations made by Lublin [13] and Jones [4] about the size of the jobs and the characterization of the communications requirements respectively, we have defined the most representative workload for a cluster-of-clusters according to the literature. As can be seen in Table 1, the defined workload is made up of 100 jobs with an interarrival time defined by a *weibull* distribution with parameters $\lambda = 82.6$ and $k = 0.6$, the job runtime was characterized by a *weibull* distribution with parameters $\lambda = 200$ and $k = 1$, and an average execution time of 170 seconds. The jobs sizes were characterized by a *gamma* distribution with parameters $\alpha = 4.04$ and $\beta = 0.77$ and adjusted to fit the prevalence of *"power of two"* job sizes as is observed by Lublin in [13]. Finally, the communication requirements ($PNBW$) of the jobs were defined as proposed by Jones in [4]. Figure 2 depicts the distribution functions used to define the workload used in this experimentation.

B. Experimental results

In order to determine the effectiveness of the co-allocation scheduling strategies, we firstly evaluated the average response time for the predefined workload. The results obtained are shown in Figure 3 for the three H configurations and for each one varying the $BSBW$ parameter. For the $H = 10\%$ case study, where the multi-cluster is approximately homogeneous, two different tendencies can be observed. When the $BSBW$ is lower than 0.6, there was enough bandwidth for all job communications, and thus, all schedulers obtained similar results. However, when the $BSBW$ increased, JPR could not obtain an adequate allocation because its main goal was to obtain the most powerful computation resources without taking the communication overhead into account. PAS and CBS obtained similar results as they tried to reduce the network usage. However, PAS also selected the best computation resources from the multi-cluster, and this allowed it to slightly reduce the average response time.

For the $H = 50\%$ and $H = 90\%$ case studies, the impact of the different effective power capabil-

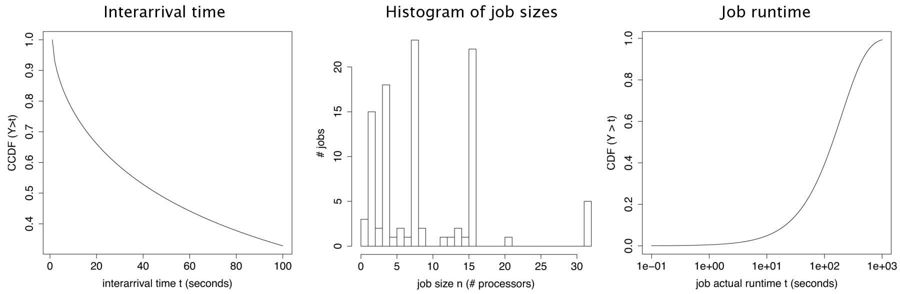

Fig. 2. Workload characteristics

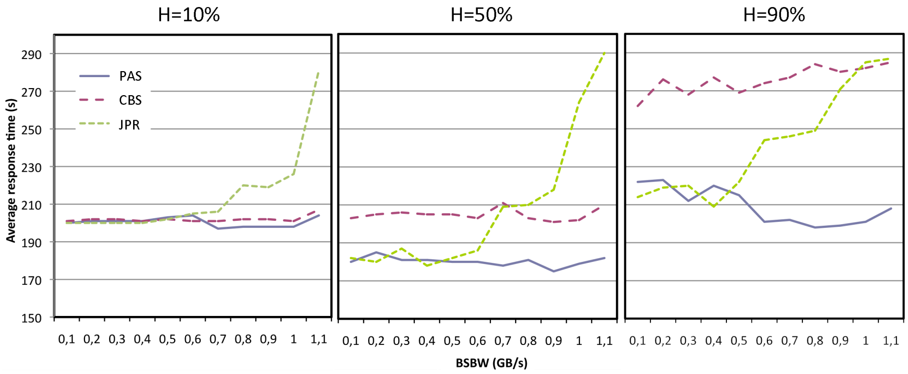

Fig. 3. Response Time.

ities among the clusters became an important factor. When the $BSBW$ was lower than 0.6, PAS and JPR were able to take advantage of the heterogeneous computation resources, obtaining lower response times. The worst results were obtained by CBS because its main goal was to reduce the network usage by forcing jobs to be allocated on individual clusters, without considering the computing capabilites. When the $BSBW$ increased, as in the previous case JPR became worse. CBS continued avoiding the use of the network communications while PAS was also able to select the computation resources that allowed the job execution times to be reduced.

From this results we conclude that the most sensitive parameter for the job response time is the network availability. When the network is available (below 0.6), for each H case study, the network availability has no effect on the response time. However, when it is less available (over 0.6), the use of inter-cluster links reduces even more the available bandwidth. Then, in order to get better response times, the schedulers should provide allocations that reduce as much as possible the inter-cluster communications.

Next, with the aim to determine how the schedulers treated the availability of the network, we evaluated how many jobs were co-allocated by each one. It is important to take into account that the co-allocated jobs are those that could be more affected by the low network availability. In Figure 4, the number of co-allocated jobs for each case of study is presented. As can be observed, the JPR was the scheduler with the higher degree of co-allocation, 60% of the total jobs present in the workload. This was due to its need to obtain the best computation resources, irrespectively of the network availability. The opposite case was CBS with an overall 14% of co-allocated jobs, as it tried to allocate at least the 75% of the job tasks to the same cluster. The PAS strategy took both, heterogeneity and network availability, into account in order to reduce the job response time. In this case, the number of co-allocated jobs was adapted to the multi-cluster characteristics. When the $BSBW$ was low, it was able to co-allocate more jobs. However, when $BSBW$ increased, and the communication slowdown penalized the response time, the degree of co-allocation was reduced forcing the jobs to be maintained inside the individual clusters when its size (number of tasks) allowed this.

The obtained results demonstrated that the co-allocation has an important performance impact on the response times when network availability decreases. By this, the schedulers should take this effect in their allocation decisions into account in order to improve the parallel applications performance.

Finally we studied the effect of the multiple-job allocation done by PAS, (both JPR and CBS, are not capable of doing multiple allocation). By using multiple-job allocation, PAS tries to maximize the multi-cluster resource usage and minimize the job package execution time.

The total number of jobs that were selected by the *Job Package Selection* is shown in Figure 5. As can be seen, despite there could be enough free computation nodes among the multi-cluster, the number of jobs treated together decreased when the $BSBW$ increased. This was because PAS tried to avoid as much as possible the saturation of the network and this implies the use of the co-allocation just only when it is profitable.

Finally, these experimental results corroborate

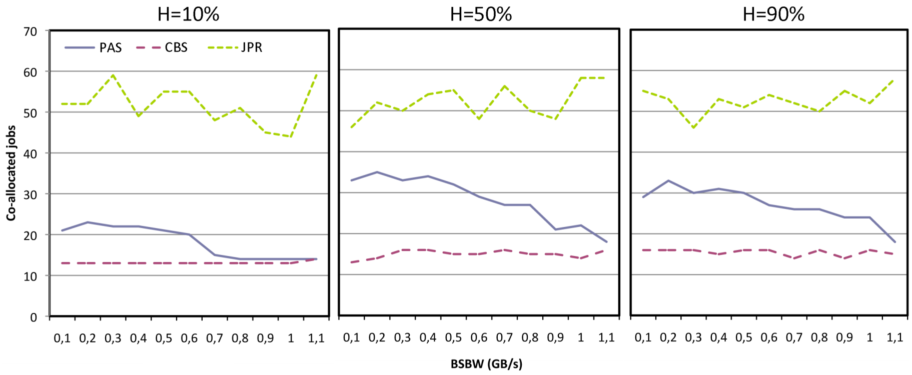

Fig. 4. Co-Allocation jobs.

that the most important multi-cluster parameter is the network availability. The more available it is, the lower effect it has on the response time. Otherwise, when the network is low available, the benefits of having high effective power on the computation nodes are less noticeable. By this, the best performance results were obtained when the schedulers tried to avoid as much as possible the inter-cluster communication links. On this situation, *PAS* obtained the best results adapting the multiple-job co-allocation to the resources availability.

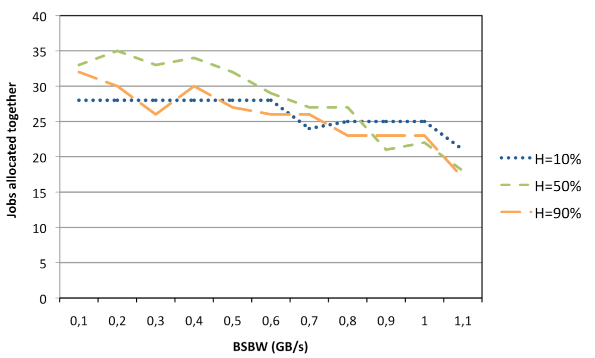

Fig. 5. Number of jobs treated in a multiple allocation.

V. Conclusions

In order to obtain the best performance, the schedulers need to be able of taking the dynamic availability on the multi-cluster resources into account. To corroborate this hypothesis, we evaluated the impact of the network availability and computation heterogeneity on the effectiveness of the scheduling process. We compared our strategy, *Package Allocation Scheduling* (*PAS*), with two other co-allocation strategies from the literature,*CBS* for *Chunk Big Small* and *JPR* for *Job Preferences on Resources*.

The results demonstrated that co-allocation has a negative effect on the response times when the network availability is low, and it must be used just only when it is profitable. On the other hand, the use of the multiple-job allocation contributes to maximize the multi-cluster resources usage while reducing the workload response times. By this, our *Package Allocation Strategy*, which is composed of both a *Job Package Selection* and *Multiple-Job Allocation*, was able to adapt to the different multi-cluster configurations and thus obtaining better scheduling decisions than the compared schedulers.

Finally, we consider that the multi-cluster schedulers need to take care not only of the current available bandwidth, but also to estimate how much bandwidth will be available after their scheduling decisions in order to obtain better results for the response times.

Acknowledgements This work was supported by the MEC-Spain under contract TIN2008-05913 and the CUR of DIUE of GENCAT and the European Social Fund.

References

[1] B. Javadi, M.K. Akbari, J.H. Abawajy, *A performance Model for Analysis of Heterogeneous Multi-Cluster Systems*, Parallel Computing,vol.32(11-12), pp.831–851, 2006.

[2] A.I.D. Bucur, D.H.J. Epema, *Schedulling Policies for Processor Coallocation in Multicluster Systems*, IEEE TPDS, vol.18(7), pp.958–972, 2007.

[3] H. Blanco, A. Montañola, F. Guirado, J. Lérida *Fairness Scheduling for Multi-Cluster Systems Based on Linear Programming.*, 10th International Conference Computational and Mathematical Methods in Science and Engineering (CMMSE'10), vol.1, pp.227-239, June 2010

[4] W. Jones, W. Ligon, L. Pang, D. Stanzione, *Characterization of Bandwidth-Aware Meta-Schedulers for Co-Allocating Jobs Across Multiple Clusters*, Journal of Supercomputing, vol.34(2), pp.135–163, 2005.

[5] J.Abawajy, S.Dandamudi, *Parallel Job Scheduling on Multicluster Computing Systems*, IEEE Int. Conf. CLUSTER'03,pp.11–18, 2003.

[6] E.M.Heien, N.Fujimoto, K.Hagihara, *Static Load Distribution for Communicative Intensive Parallel Computing in Multiclusters*, IEEE PDP'08,pp.321-328, 2008.

[7] C. Yang, H. Tung, K. Chou, W. Chu *Well-Balanced Allocation Strategy for Multiple-Cluster Computing*, IEEE Int. Conf. FTDCS'08, pp.178–184, 2008.

[8] V.K. Naik, C.Liu, L.Yang, J.Wagner, *Online Resource Matching for Heterogeneous Grid Environments*, IEEE/ACM Int. Conf. CCGRID'05,vol.2,pp.607–614, 2005.

[9] J.L. Lérida, F. Solsona, F. Giné, J.R. García, P. Hernández, *Resource Matching in Non-dedicated Multicluster Environments*, In VECPAR'08, pp.160–173, 2008.

[10] *The DAS-2 Supercomputer.* Available from: http://www.cs.vu.nl/das2

[11] *The Livermore National Laboratory Multi-cluster system.*, Available from: https://computing.llnl.gov/tutorials

[12] H. Li, D. Groep, L. Wolters. *Workload Characteristics of a Multi-cluster Supercomputer*, Job Scheduling Strategies for Parallel Proc., JSSPP'04, pp.176-193, 2005.

[13] U.Lublin, D.G. Feitelson. *The workload on parallel supercomputers: modeling the characteristics of rigid jobs.*, J.Parallel Distr.Comp., vol.63(11),pp.1105-1122, 2003.