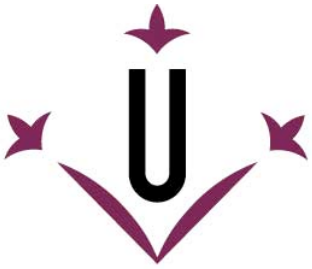
Universitat de Lleida

Meta-Heuristics for Scheduling in Cluster Federated Environments

Eloi Gabaldon Ponsa



Universitat de Lleida

TESI DOCTORAL

Meta-Heuristics for Scheduling in Cluster Federated Environments

Eloi Gabaldon Ponsa

Memòria presentada per optar al grau de Doctor per la Universitat de Lleida
Programa de Doctorat en Enginyeria i Tecnologies de la Informació

Director/a
Josep Lluís Lérida
Fernando Guirado

Tutor/a
Josep Lluís Lérida

2017

Abstract

Many organizations, companies or universities have accumulated, over the years, a large number of computing resources grouped in Clusters. *Cluster Federated Environments* arise as a new architecture with the objective of joining all these resources, increasing the global computing capacity of the organization without making a great economic investment.

However, the high number of machines and computing resources imples great energy consumption. Due to the economic and sustainable connotations that this entails, recently a new line of investigation has focused on reducing energy consumption while maximizing the performance of the applications and the usage of the system.

The scheduling in these environments, responsible for allocating the applications to the system resources, offers the possibility of obtaining great improvements, as managing the resources correctly can have a great impact on the system performance and energy efficiency. However, this process is very complex, since it belongs to the NP problem group.

This PhD studies the problem of scheduling large batch *workloads* extracted from diverse real traces. The proposed techniques consider the heterogeneity of the system resources as well as the ability to apply co-allocation in order to take advantage of the leftover resources across clusters. The proposals will use sophisticated multi-criteria tactics, based on Genetic Algorithms and Particle Swarm Optimization, focused on reducing both the execution time of the jobs and the energy consumption of the system.

The results show the effectiveness of the proposed methods, which provide solutions that improved the performance compared with other well-known techniques in the literature, opening new and interesting research lines in the scheduling field in highly distributed and heterogeneous environments.

Resum

Avui en dia, moltes organitzacions, empreses o universitats han anat acumulant, durant anys, un gran nombre de recursos computacionals agrupats en clústers. Els *Entorns Clúster Federats* sorgeixen com una nova arquitectura amb l'objectiu d'unir tots aquests recursos, augmentant la capacitat de còmput global de l'organització sense haver de fer una gran inversió econòmica.

No obstant això, l'elevat nombre de màquines i recursos computacionals, comporten un gran consum energètic. A causa de les connotacions econòmiques i sostenibles que això implica, recentment s'ha obert una nova línia d'investigació que s'ha centrat a reduir el consum d'energia i maximitzar el rendiment de les aplicacions i la utilització dels recursos computacionals .

La planificació en aquests entorns, responsable d'assignar les aplicacions als recursos del sistema, ofereix la possibilitat d'obtenir grans millores, ja que gestionar correctament els recursos pot tenir un gran impacte en el rendiment del sistema i en l'eficiència energètica. Tanmateix, aquest procés és molt complex, ja que pertany al grup de problemes NP.

Aquesta tesi estudia el problema de la planificació de grans *workloads* extrets de diverses traces reals. Les tècniques proposades consideren l'heterogeneïtat dels recursos del sistema, així com també la capacitat d'aplicar la co-assignació per aprofitar els recursos sobrants de cada clúster. Les propostes utilitzaran tàctiques sofisticades *multi-criteri*, basades en Algoritmes Genètics i Particle Swarm Optimization centrades en la reducció tant del temps d'execució dels treballs com del consum energètic del sistema.

Els resultats mostren l'efectivitat dels mètodes proposats, proporcionant solucions que milloren el rendiment respecte a altres tècniques presents en la literatura. Obrint una nova i interessant línia d'investigació en el camp de la planificació en entorns altament distribuïts i heterogenis.

Resumen

Hoy en día, muchas organizaciones, empresas o universidades han ido acumulando, durante años, un gran número de recursos agrupados en clústeres. Los *Entornos Cluster Federados* surgen como una nueva arquitectura con el objetivo de unir todos estos recursos, aumentando la capacidad de cómputo global de la organización sin tener que hacer una gran inversión económica.

Sin embargo, el elevado número de máquinas y recursos de computo, comportan un gran consumo energético. Debido a las connotaciones económicas y sostenibles que ello implica, recientemente se ha abierto una nueva línea de investigación que se ha centrado en reducir el consumo de energía y maximizar el rendimiento de las aplicaciones y utilización de los recursos.

La planificación en estos entornos, responsable de asignar las aplicaciones a los recursos del sistema, ofrece la posibilidad de obtener grandes mejoras, ya que gestionar correctamente los recursos puede tener un gran impacto en el rendimiento del sistema y en la eficiencia energética. Sin embargo, este proceso es muy complejo, ya que pertenece al grupo de problemas NP.

Esta tesis estudia el problema de la planificación de grandes *workloads* extraídos de distintas trazas reales. Las técnicas propuestas consideran la heterogeneidad de los recursos del sistema, así como también la capacidad de aplicar la co-asignación para aprovechar los recursos sobrantes de cada clúster. Las propuestas utilizarán tácticas sofisticadas *multi-criterio*, basadas en Algoritmos Genéticos y Particle Swarm Optimization centradas en la reducción tanto del tiempo de ejecución de los trabajos como del consumo energético del sistema.

Los resultados muestran la efectividad de los métodos propuestos, proporcionando soluciones que mejoran el rendimiento respecto a otras técnicas presentes en la literatura. Abriendo una nueva e interesante línea de investigación en el campo de la planificación en entornos altamente distribuidos y heterogéneos.

Live as if you were to die tomorrow.

Learn as if you were to live forever.

– Mahatma Gandhi

Acknowledgements

I would like to thank all those people who helped and supported me during the preparation of this PhD. First of all, I would like to thank my supervisors, Dr. Josep Lluís Lérida and Dr. Fernando Guirado, who supported and guided my work steadily and supervised every aspect of my research.

I want to acknowledge the Computer Science Department at the University of Oviedo, especially Dr. Jose Ranilla and Dr. Luciano Sanchez, for their help provided while developing my techniques.

I would also like to thank all the members of the Distributed Computing Group at the University of Lleida. In particular, I would like to thank Fernando Cores, Francesc Gine, Francesc Solsona and Concepció Roig.

I thank all my co-workers, the ones at the university, Jordi Vilaplana, Ivan Teixidó, Jordi Mateo, Jordi Lladós, Ismael Arroyo, Miquel Orobitg, Josep Rius, Anabel Usié and Sergi Vila, for their support and friendship, my co-workers at Fractal S.L., Marc Solé, Marc Gonzàlez and Arnau Torrente, and the ones at the BeeGroup-CIMNE, Jordi Cipriano, José Santos Lopez, Gerard Mor and Josep Mayos, for their patience while finishing my PhD.

I also want to mention Montse Espunyes for her splendid management and best practices in all administrative processes required during the PhD.

Finally, I would like to thank my friends and family for their support during these tough times, especially my parents, Pere and Lourdes, who provided all sort of opportunities during my childhood and encouraged my education. Also Núria, Blai and Diego, for being a great sister, brother and brother-in-law.

Thanks to all.

This thesis has received a grant for its linguistic revision from the Language Institute of the University of Lleida (2017 call)

Contents

List of Figures

Chapter 1

Introduction

Throughout the history of computer science, there has always been a need to run applications that are continuously growing in computational requirements, thus needing more and more powerful computers to be executed. This need has empowered the investigation and development of new methods and strategies to allow the execution of these applications, improving their performance.

Initially, the computing systems were evolving to increase the power of a single processor. The appearance of supercomputers was a great leap forward in terms of computing performance, as they allowed the use of great amounts of processors together to execute the applications. Despite the high computational power provided by this kind of environments, its scalability is limited, as we cannot add unlimited processors or memory to a single computer without a very high cost and redefining the architecture.

With the reduction in price of commodity computers and the increase in their of computing power, in the 90's, the interest of using these resources cooperatively to create a new architecture emerged. These systems were called Clusters. This architecture consists of the connection of many desktop computers to a communication network in order to take advantage of the joint computing power. Clusters allowed the execution of scientific applications that are growing continuously, thanks to their high scalability with a low price, as we can increase their computational power by adding new computers to the network.

Nowadays, the number of computing resources available in companies, organizations or universities is growing exponentially. The amount of available computational resources encouraged researchers to study and develop new techniques to take advantage of them, allowing the execution of applications with higher requirements. However, the vast amount of computers working concurrently produces high energy consumption and tons of CO_2 emissions into the environment, opening another interesting research topic focused on reducing the environmental impact of these systems.

Besides this, another complex problem to solve in these distributed architectures is the scheduling of applications, which consists of allocating them to the available resources in the system. Due to the high amount of resources and the heterogeneity among them, proper scheduling is crucial to efficiently manage all resources to reduce the computational cost and energy consumption. This PhD work focuses on solving this scheduling problem by proposing sophisticated techniques able to find solutions that increase the system performance and optimize the energy consumption.

Different distributed architectures have been developed to take advantage of the joint of combined computing resources. In order to introduce the scope of this PhD, Section 1.1 introduces the concepts of distributed computing and enumerates the most common distributed computing architectures. Next, the scheduling problem considering the architecture selection is introduced in Section 1.2, describing all the characteristics that have a significant impact on the system behavior considered in this work for the optimization. Having introduced the basic concepts and the architecture used in the present PhD, Section 1.3 presents the modeling used to predict the behavior of the system. In Section 1.4 an in-deepth introduction of the existing related work in this field is presented. Finally, Section 1.5 shows the general structure of the PhD dissertation.

1.1 Distributed Computing

A Distributed Computing System is composed of a group of computer nodes connected through a communication network; each node has its own memory and the

communication between them is performed through messages. Thus, Distributed Computing refers to the concurrent execution of the same application in two or more different computers. This definition must not be confused with the term of Parallel Computing, which refers to the execution of the applications in memory-shared multiprocessor systems.

Distributed Computing Systems have emerged as a new paradigm to solve complex computational problems and were encouraged thanks to the advances in the computing capacity of commodity computers and the development of high-speed communication networks. In the last three decades, a vast amount of scientific studies have contributed to this field with improvements and techniques to increase and take advantage of these architectures.

In Distributed Computing Systems, a wide range of different architectures can be found. Next, we will introduce the most common, and most related to the scope of this work.

Cluster architectures are the first and most basic distributed computing systems that were developed. They are composed of a set of computational nodes connected, by using a dedicated network, to the same administrative domain. The nodes collaborate to execute several parallel and non-parallel applications. These systems reached popularity in the 90s, thanks to the improvements in the communication networks [LLM88, BSS+95, ELvD+96]. Clusters can achieve very high performance even when using computing nodes of medium computational power however, the energy consumption can be high due to the number of different computing resources that they are using. Figure 1-1 shows a cluster diagram, where we can see the different computing resources connected to a central switch that interconnects all the computers; these resources collaborate to run the applications led by a Cluster Scheduler.

Cluster Federation architectures are the conjunction of the different clusters that can be found in the same institution or organization. They were created with the idea of increasing the computational power of one institution

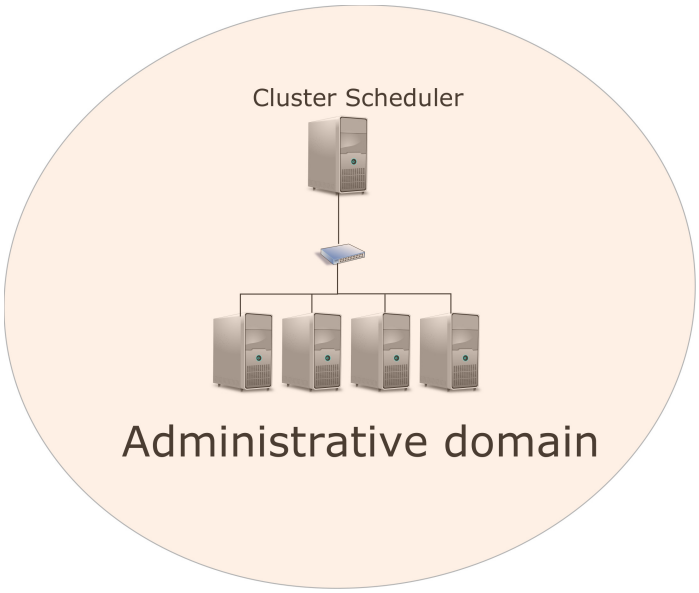


Figure 1-1: Cluster architecture

by taking advantage of the resources acquired over the years. In this architecture, the resources can be in different administrative domains. As in the cluster architecture, the network is dedicated; thus, the performance of the communications can be estimated [BB01].

Despite the great computational power that can be achieved by these systems, the use of a great amount of resources in an organization increases the energy consumption to high levels. For this reason, it is also a very important issue to find methods to reduce the energy consumption. Figure 1-2 shows the cluster federation architecture where different clusters are connected, using a dedicated network, to a central switch in order to interconnect all the resources.

Grid architectures are a connection of different computing systems such as clusters, supercomputers or commodity computers that can be found in different organizations. These resources are connected through universal connection protocols like the Internet to create a big network of computing resources. In these systems, the energy consumption is shared among the different organizations, as each is responsible for maintaining its own infrastructure. Although the economica costs are shared, these systems have a large effect on the carbon

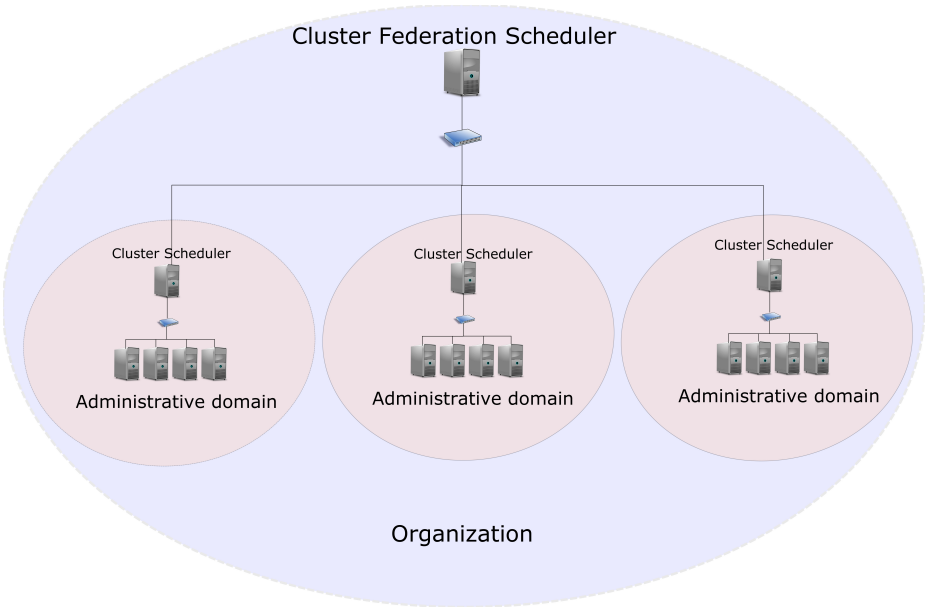


Figure 1-2: Cluster federation schema

footprint produced due to the huge energy consumption of the whole system.

In Grid systems, the users can access all the resources available in the infrastructure; however, for high performance computing applications, due to the unpredictable performance of Internet communication links, the application execution must be confined in the selected system boundaries [FK03] [Sto07]. This peculiarity differentiates this architecture from that explained previously, where the network is dedicated. Figure 1-3 shows a Grid system diagram, where we can see examples of different resources placed in different institutions, connected through the internet to share their computing resources.

Cloud architectures are organized in three different layers depending on the service they offer. The lowest layer, called Infrastructure as a Service (IaaS), consists of offering hardware, storage and physical devices over the Internet. The middle layer, called Platform as a Service (PaaS) offers the capability of deploying applications or services without managing or controlling the underlying Cloud infrastructure. The highest layer, called Software as a Service (SaaS) offers software and hosted applications over the Internet destined for the final

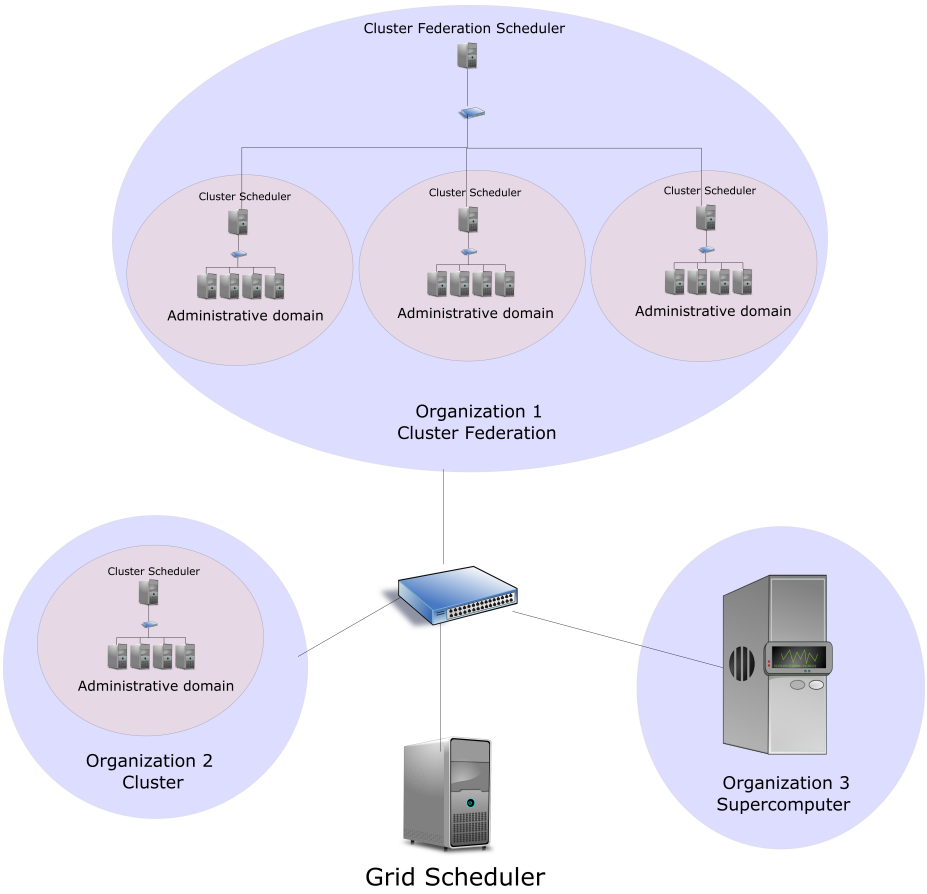


Figure 1-3: Grid architecture

user [May15].

For distributed computing purposes, only cloud solutions served as IaaS can be considered, because the other options hide the possibility of configuring a distributed computing system. With these infrastructures and thanks to the use of virtualization, the users can configure their own executing environment depending on their needs.

These systems can be observed from the Cloud User or the Cloud Provider point of view. Cloud Users only have access to their configured resources or virtualized systems (by paying the provider fee), releasing them from any administrative issue such as maintaining costs and scalability. Cloud Providers take care of the real system infrastructure, which is usually composed of server farms, powerful clusters or supercomputers to provide resources for all their clients. Figure 1-4

shows a cloud diagram, where the users can access the resources in the Cloud by using the internet, without any knowledge of the real infrastructure system.

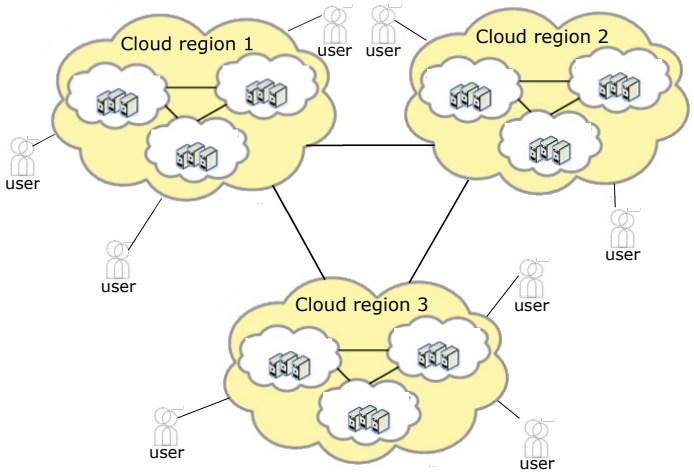


Figure 1-4: Cloud architecture

This PhD focuses on promoting the possibility of taking advantage of the existing resources available in a single organization, allowing the execution of bigger parallel applications than permitted by each administrative domain independently. The architecture in the scope of this PhD is the Cluster Federated Environment, as it can increment the computational power of an organization without the expenses of buying more computational resources or paying Cloud fees. Finally, the optimization will be addressed with two objectives, energy consumption and global execution time.

1.2 Scheduling in Cluster Federated Environments

The scheduling problem consists of allocating the applications to the available resources in the system, satisfying one or more performance requirements. This problem is really interesting, as finding a proper allocation of the tasks can greatly determine improvements in the system usage, execution times and reduction of energy consumption. However, as stated by Mosheiov in [Mos98], the scheduling when dealing with multiple machines is categorized as an NP-Hard problem, thus meaning that the

complexity of finding a good solution to the problem increases exponentially with the problem size (number of parallel applications and computing resources).

In this section, we present an overview of the elements that interact with the scheduling problem in Cluster Federated Environments and the main issues that turn this problem into a challenge to be solved.

1.2.1 Heterogeneity

One of the main aims of using Cluster Federated Environments is the global use of the different infrastructures accumulated over years in an organization. However, the number of resources in organizations is growing continuously and the resources can be completely different in performance and characteristics.

A common scenario of a Cluster Federated Environment is when some organization with different clusters joins the older clusters to the new ones to increase the available computational power.

In this scenario, the system composed will have a vast number of heterogeneous resources working together, with different computational power and energy consumption values. Thus, in these systems, the scheduling plays a crucial role to achieve good performance as the execution time and energy consumed during the execution will change depending on the resources selected [JLPS05].

This PhD considers heterogeneous Cluster Federated Environments aiming to solve the scheduling problem in the scenario described.

1.2.2 Workloads

The set of applications, also referred to as jobs, that the system has to execute are called Workloads. The characteristics of these workloads are also an important aspect to take into account. We can differentiate the workloads that are executed in the system in two main groups, based on the availability of the jobs at the starting point of the execution. Thus, the workload can be classified as batch or online. Below, the main differences between them are presented.

In batch mode, all the applications are already in the system queue at the start of the execution, allowing the scheduler to analyze the characteristics of the workload and prepare the scheduling in advance.

In online mode, the applications reach the system unpredictably during the course of the execution, providing an unknown factor to the scheduler that prevents any future planning.

With this differentiation of the workload modes, it can be observed that when working with batch workloads, the system can analyze the near future, creating new scheduling opportunities to increase the system performance or reduce the energy consumption by properly selecting the heterogeneous resources [BdF12]. However, when working with online mode, the scheduler only has information of the applications at the time when they reach the waiting queue and has to determine their scheduling. When the application arrival rates are higher than the system throughput, the applications accumulate in the system waiting queue, providing a set of applications that can be considered as a batch workload. When the arrival rates of the applications are low, there are only a few applications in the queue and there is no competition for the best resources; thus, the scheduler only has to allocate the applications to the best available resources to obtain the best performance.

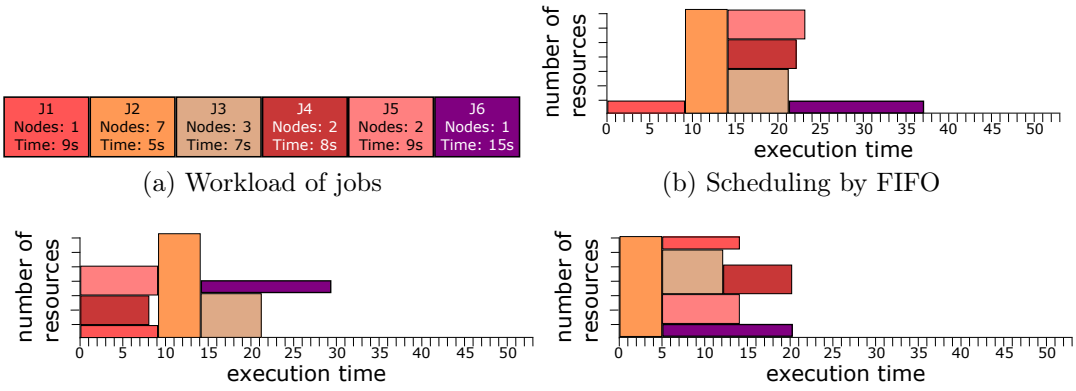
This PhD will focus on scheduling when the workloads accumulate in the waiting queue, considering either batch workloads or online mode with high arrival rates. The workloads used in the experimental study of this dissertation were extracted from the real traces that can be found in the Workloads Parallel Archive webpage managed by Feitelson [Fei14]

1.2.3 Scheduling Ordering

Regarding the scheduling of batch workloads, we can find techniques that take the jobs one by one from the waiting queue, following the order in which they arrived, in most cases obtaining a far from optimal allocation. To increase the performance of the allocation in the literature we can find the backfilling technique [SF05]. This

technique aims to detect the gaps where the nodes are not being used and search for a future job in the queue that fits in this gap without delaying any application in the waiting queue, increasing the system utilization and reducing the execution time. Besides this, several authors have demonstrated that changing the order of the jobs that are going to be executed in the system, by treating the workload as a set of jobs, can provide better results [BdF12], as it allows better scheduling opportunities to be created in order to increase the performance or reduce the energy consumption of the whole workload. Example 1 shows the behavior after applying different techniques to treat the jobs and the result obtained.

Example 1. Considering the set of jobs of Figure 1-5a, each one with different node requirements and execution time. The order determines their position in the waiting queue. The allocation of these jobs to the system resources is represented by a chart, where the X axis shows the execution time and the Y axis shows the number of resources. By using a heuristics that only treats the first job in the waiting queue such as First In First Out (FIFO), the allocation provided does not take advantage of all the resources of the system. Job 1 is allocated to the first node; then, when allocating Job 2, it requires more resources than those available (as Job 1 is using one node) and the job has to wait, leaving resources empty and delaying the execution of the rest of the Jobs (Figure 1-5b). Using the backfilling technique, the performance of the system can be improved with respect to FIFO; however, it is still far from optimal. Taking the FIFO allocation as the starting point, the backfilling goes through the waiting queue searching for jobs to fill the gap without delaying the execution of the other jobs; it moves Job 3 and Job 4, freeing resources to allow the execution of Job 6 earlier (Figure 1-5c). Finally, by treating the workload as a set of jobs and changing their execution order, optimal performance can be achieved. The technique analyzes the waiting queue and determines that allocating Job 2 as the first job is the optimal allocation for the whole set of jobs, as it allocates all the jobs, avoiding any gap and obtaining the best execution time (Figure 1-5d).



(a) Workload of jobs

(b) Scheduling by FIFO

(c) Scheduling by using backfilling

(d) Scheduling with sets of jobs

Figure 1-5: Scheduling example with different techniques

1.2.4 Co-allocation

Another aspect that has to be taken into account when scheduling in Cluster Federated Environments is related to the co-allocation technique. This technique consists of allocating a parallel job by distributing its tasks between two or more different clusters. Thus, the system is able to run applications that require more resources than those available in a single cluster, increasing the applications that can be executed simultaneously. This technique can be used in Cluster Federated Environments thanks to the dedicated nature of the communication links that interconnect the system clusters, as the communication volume and cost of the different tasks can be predicted and taken into account when doing the scheduling.

To properly explain the co-allocation technique, first we will take a look at the scheduling in most distributed systems where co-allocation can not be applied. In these cases, the execution of the different tasks of a jobs has to be confined to a single cluster, and therefore, even if the whole system has hundreds of computing resources available, only the ones in the same administration domain are available for a single job. Example 2 shows the allocation of a set of jobs without using the co-allocation technique.

Example 2. Consider a cluster federation composed of two clusters, each one with a different number of computing resources 4 and 3. We also have a waiting queue of 3 jobs, with different node requirements, 3, 2 and 2. Figure 1-6 shows the allocation

obtained, where Job 1 is allocated to C1, by using 3 of its nodes, and Job 2 is allocated to C2, by using two of its computing nodes. With no co-allocation, job 3 has to wait in the queue as there are not enough free nodes in a single cluster to allocate the job.

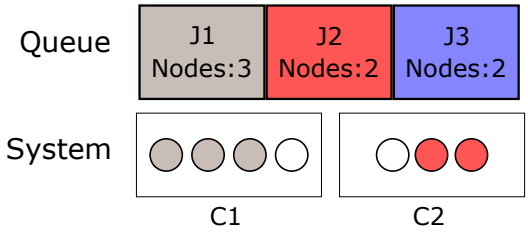


Figure 1-6: Example without co-allocation

We call internal fragmentation the number of free resources in a system that remain free because they cannot be used to allocate a job, as there are not enough resources for the queue jobs to be allocated. In the previous example, we had an internal fragmentation of one node at each cluster. Even when the sum of the resources was enough to allocate Job 3, this allocation was not possible without using the co-allocation technique.

Taking advantage of the dedicated link that connects the different clusters in a Cluster Federated Environment and that allows the job tasks that are on different clusters to communicate, the co-allocation technique can be used to increase the performance of the system, joining the free resource of each cluster and allocating Job 3. With this allocation, the internal fragmentation of the system is reduced and the utilization increased [SCJG00, JLPS05].

This technique can greatly improve the usage of the system; however, it can also provide some problems that must be avoided. All the co-allocated tasks must use the inter-cluster network links to pass the required information during their execution. If the co-allocation is used in excess, the amount of data that has to be sent through the same link can exceed the maximum bandwidth of the communication link, and this will cause the saturation of the network, producing an overall performance loss in the execution system [BE07]. Example 3 shows an example of scheduling using

co-allocation.

Example 3. Consider an environment composed of 3 Clusters: C1 of 4 nodes, C2 of 3 nodes and C3 of 5 nodes. In this example, the workload to be executed is composed of 2 jobs that require 5 and 7 computing nodes each. Figure 1-7 shows the scheduling using co-allocation, where Job 1 is allocated to all nodes of cluster C1 and one node of cluster C2, and Job 2 to all nodes of cluster C3 and two nodes of cluster C2. However, this allocation makes the jobs compete for the bandwidth of the communication link that connects cluster C2 to the Cluster Federated Environment, and could produce saturation if the communication requirements of Job 1 and Job 2 exceed the maximum bandwidth of the shared link.

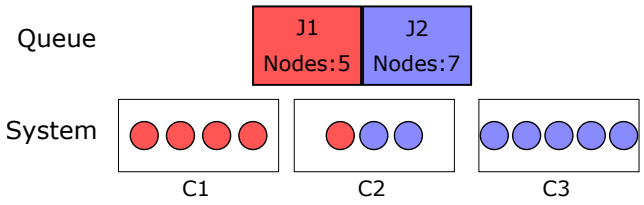


Figure 1-7: Figure of Example 3

1.3 Cluster Federation Environments Modelling

In order to predict the behavior of the Cluster Federated Environments when executing a set of jobs, it is necessary to model the whole environment. The Distributed Computing Research Group of the University of Lleida, where this PhD work was performed, has developed some models for this purpose over the last years [Mon09, BdF12]. These models focus on the prediction of the execution time of the applications, taking into account not only the characteristics of the computing processors, but also the performance of the communication links that interconnect the different administrative domains and the effects on the co-allocation technique.

This PhD aims to go a step further in the research, proposing to reduce the energy consumption of the system while also taking into account the execution time.

However, the energy efficiency of the system is a new field of research and has not being widely studied in the research group until this work. Thus, we present an energy consumption model that creates synergies with the execution model of the group in order to allow the optimization of this metric.

The rest of this section introduces the architecture, application, execution time and energy consumption models used in this PhD.

1.3.1 Cluster Federation Model

A Cluster Federation Environment consists of a set of α arbitrarily sized clusters that can contain several heterogeneous computing nodes. These systems differ from the other distributed systems such as Grid and Cloud in that all the computational nodes are physically in the same institution, allowing the ability to control and predict their availability. Also, unlike the other distributed systems, that use the Internet to connect the resources, the communication between the different clusters in a Cluster Federation Environment is performed by means of dedicated links. This allows the prediction of the performance of the communication operations.

A Cluster Federation Environment model is shown in Figure 1-8. We consider $\mathcal{C} = \{C_1, \ldots, C_\alpha\}$ as the set of clusters that compose the system and $\mathcal{N} = \{N_1^1, \ldots, N_n^\alpha\}$ the set of computing nodes. Finally, for the inter-cluster connection links we use $\mathcal{L} = \{L_1, \ldots, L_\alpha\}$, where L_k is the link between the cluster C_k and the central switch and $\{B^1, \ldots, B^\alpha\}$ are the corresponding maximum bandwidths for each link.

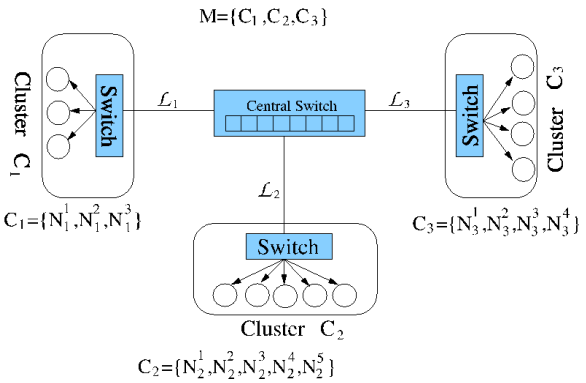


Figure 1-8: Cluster Federation Infrastructure

In a heterogeneous environment, all the resources can have a different computing power. In this model, the computing power is characterized by its effective power (Γ_n). The effective power relates the computational power of the computing node to another in reference. To calculate Γ_n, a benchmark is performed in all computing nodes, then one computing node is selected as reference node (N_{ref}). The effective power is calculated by relating the power of the reference computing node P_{ref} to the power of the corresponding node P_n by the Equation: $\Gamma_n = P_{ref}/P_n$. Then, if $\Gamma_n = 1$ the computing node has the same computing power as the reference node, if $\Gamma_n < 1$ the power is lower , and if $\Gamma_n > 1$ means that the power is higher than the reference node.

1.3.2 Application Model

Several application models have been developed to model the jobs that are executed in the distributed systems to provide an approximation of the applications' behavior. The dependency of the jobs and their precedence, the distribution of the jobs into different tasks, their characteristics, requirements and costs are important information when predicting the job execution time of the applications.

In Cluster Federated Environments, not all the applications submitted for execution are parallel, as it is also a common practice to submit several applications of the same type, with only one task, and no dependence among them. The BSP (Bulk-Synchronous Parallel) Model [SHM97] is selected to define the behavior of the execution applications. BSP has acquired considerable interest in the recent years when it was adopted by Google as a major technology for graph analytics on a mass scale. This model is used in technologies like Pregel and MapReduce, and, there are also new active open source projects to add explicit BSP programming, such as Apache Hama and Apache Giraph.

Taking this type of applications as the basis, the characteristics assumed for the jobs in the present work are as follows:

- All jobs are independent from each other.

- The jobs are divided into several tasks of similar computing and communication requirements.
- Each task must be allocated to only one computational resource.
- The tasks combine computing operations and communication operations; however, the communication can be null.
- The tasks are considered to use an all-to-all communication graph.
- All tasks of the same job must be allocated until the last task of this job finishes its execution. A Barrier synchronization is performed at the end of the execution.

The problem we want to solve consists of scheduling a set of n jobs that compound a workload $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ into the previously described system. A job J_i is composed of a fixed number τ_i of collaborative tasks. Each task consists of various processing, communication and synchronization phases, and can only be executed in one node. Given a schedule, job J_i is in node N_r, expressed as $J_i \in N_r$, if there is at least one task of job J_i being executed in node N_r. In our model, the Job assignment is static, avoiding re-allocations while the job is being executed. Additionally, jobs can be co-allocated to different clusters in order to reduce their execution time and the internal cluster fragmentation.

1.3.3 Execution Model

In order to estimate the execution time for a job in a heterogeneous Cluster Federated Environment, based on the model presented in [Mon09], we characterize every job by two factors: the Processing Slowdown (PS) and the Communication Slowdown (CS). The PS for job J_j is obtained from the slowest processing node J_j is assigned to, i.e. the allocated node providing the maximum processing slowdown. Equation 1.1:

$$PS_j = \max_{\forall r: J_j \in N_r} \{PS_j^r\}, \quad J_j \in \mathcal{J} \tag{1.1}$$

where PS_j^r is the slowdown of job J_j in node N_r, which is inversely proportional to the node computation power $(1/\Gamma_r)$.

The co-allocation of a parallel job J_j consumes a certain amount of bandwidth in each inter-cluster link L_k, which is calculated with Equation 1.2:

$$B_j^k = \left(t_j^k \cdot B_j\right) \cdot \left(\frac{\tau_j - t_j^k}{\tau_j - 1}\right), \quad J_j \in \mathcal{J}, C_k \in \mathcal{C} \tag{1.2}$$

where B_j is the required per-task bandwidth in cluster k, τ_j is the number of tasks in job J_j, and t_j^k is the number of tasks of job J_j allocated to cluster C_k. The first term in the equation is the total bandwidth consumed by tasks of job J_j in cluster C_k, and the second term is the percentage of communication with other clusters.

Saturation occurs when co-allocated jobs use more bandwidth than that available, and jobs sharing the link are penalized by an increment in their communication time. The inter-cluster Saturation Degree SD^k relates the maximum bandwidth B^k of each link L_k to the bandwidth requirements of the allocated parallel jobs, Equation 1.3:

$$SD^k = \frac{B^k}{\sum_{\forall J_j \in N_k}(B_j^k)}, \quad L_k \in \mathcal{L}. \tag{1.3}$$

where B_j^k is the bandwidth of job J_j when it is in node N_k.

When $SD^k < 1$ the link L_k is saturated, and jobs using the link are delayed; otherwise, it is not. Then, the communication slowdown for job J_j and link L_k, which depends on the saturation, is expressed by Equation 1.4.

$$CS_j^k = \begin{cases} (SD^k)^{-1} & \text{when } SD^k < 1 \\ 1 & \text{otherwise} \end{cases} \qquad J_j \in \mathcal{J}, C_k \in \mathcal{C} \tag{1.4}$$

The communication slowdown for job J_j is the CS_j^k from the most saturated link used, expressed by Equation 1.5

$$CS_j = \max_{\forall N_k : J_j \in N_k} \{CS_j^k\}, \quad J_j \in \mathcal{J} \tag{1.5}$$

Finally, the estimated execution time for a parallel job J_j is calculated by Equation 1.6

$$T_j^e = Tb_j \cdot tc_j, \quad J_j \in \mathcal{J} \tag{1.6}$$

where Tb_j is the base time of job J_j in dedicated resources, and tc_j is the time cost factor when a job is allocated. It is assumed that the base-time Tb_j is known from user-supplied information, experimental data, job profiling, etc. Previous authors computed the time cost tc_j from the allocated resources without considering communications [JLPS05] or considered a fixed communications penalty when co-allocation is applied [EHS+02]. In contrast, we model the time cost based on the heterogeneity of the processing resources selected and the availability of the inter-cluster links used. The time cost for job J_j is expressed by Equation 1.7

$$tc_j = \sigma_j \cdot PS_j + (1 - \sigma_j) \cdot CS_j, \quad J_j \in \mathcal{J} \tag{1.7}$$

where PS_j denotes the maximum processing slowdown from the allocated resources, CS_j is the communication slowdown from the inter-cluster links, and σ_j is the portion of the total execution time spent on processing.

1.3.4 Energy Model

The energy consumption model has been developed to predict the energy consumption by the system when executing the applications in a workload. The energy consumption of distributed systems has been an emerging concern over the last decade; thus, several models were developed in order to predict the energy consumption of the systems [VMT+15, OLG08, LLQ09]. However, these models were designed to work in distributed systems such as Grid, Cloud and Clusters. For this reason, this PhD proposes an adaptation of the existing models to work with the Cluster Federation Environments that create synergies with the execution model explained previously.

The energy model considers that the computing nodes can have two different states, depending on whether they are executing a task (Computing state) or they

are idle (Idle state), with no task assigned to them. This hypothesis was previously stated by the model expressed by Orgerie in [OLG08]. Once the energy consumed by each node has been defined, the energy consumption during the execution is estimated by multiplying each consumption value by the time spent in each state. Equation 1.8 defines the energy consumption using the proposed model.

$$energy = \sum_{n}(C_n * CT_n + I_n * IT_n) \quad \forall n \in \ N \tag{1.8}$$

where C_n is the energy consumed by node n when it is computing and I_n when it is idle, CT_n is the computing time of the node n and IT_n is its idle time.

1.3.5 Optimization Techniques

As stated earlier in this section, the scheduling of applications in a Cluster Federation Environment is an NP-Hard problem, making it very costly to be solved when the problem involves just a few jobs and resources, requirements that are easily found in real Cluster Federated Environments.

Thus, it is extremely important to select the optimization techniques used to perform the scheduling properly, as they can determine the scheduling efficiency and also the system performance. The optimization techniques can be categorized in the following types, depending on whether or not the solution found is optimal and on the nature of the technique. (Figure 1-9):

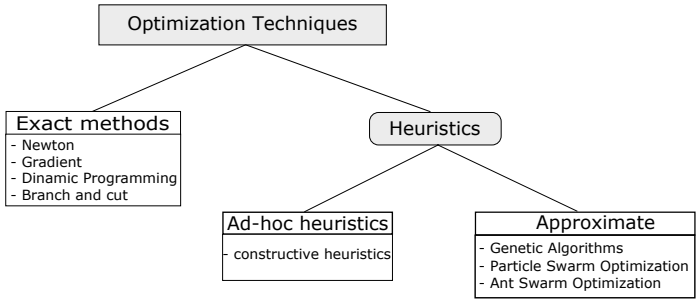


Figure 1-9: Techniques organization schema

Exact Methods guarantee that the optimal solution is found if the method is given sufficient time and space. However, the time cost of calculating the solutions grows exponentially with the problem. These techniques can encounter trouble when working with big NP problems, as there are no expectations to find polynomial time solutions, and they cannot be used in real systems due to the high computational cost even with small problem sizes [BdF12].

Ad-Hoc Heuristic Techniques are solution methods belonging to the heuristics group. They construct the solution by using a fixed set of operations. However, there are no guarantees about the solution quality. The heuristics can be differentiated in two sub-groups: deterministic or stochastic, depending on whether it always returns the same result for the same input, or it has some randomness in its responses. Heuristics is typically used to solve real-life problems because of its speed and its ability to handle large instances [JLPS05, NLYW05, LLQ09].

Approximate Heuristics is a special class of heuristics that can provide a near optimal solution in less computational time than exact methods. These algorithms find solutions that successfully satisfy a fitness function in less computational time than the exact methods [OBBB15, BV98]. They start with a bad solution that, through iterations, is approximated to the optimal solution.

Taking into account the nature of the problem, we can state that the exact methods, which find the optimal solution, cannot be used for large problems as the computational cost grows exponentially, not being practical for real situations. However, the use of exact methods and the analysis of the results can help to construct effective heuristics. The knowledge obtained in our research group in the use of these techniques [BdF12] helped us to propose some ad-hoc heuristics [BLGL12]. Other authors have developed a wide variety of ad-hoc heuristics in order to tackle this problem [JLPS05, NLYW05, LLQ09]. However, the solutions found by these techniques can be far from optimal as they are only focused on finding feasible solutions

that fit some simple criteria (allocate to the most powerful nodes available, avoid communication link saturation, execute the shortest job first, etc).

This dissertation proposes advanced techniques focused on finding optimal or near optimal solutions for scheduling problems in large heterogeneous systems. The techniques proposed are based on approximate heuristics such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO) and also Hybrid algorithms (composed of PSO and GA) focused on optimizing the Makespan, the Energy Consumption or both metrics by using advanced Multi-Criteria approaches.

The basic operation of Genetic Algorithms and Particle Swarm Optimization is briefly described below in this section. The proposal based on these techniques is elaborated in the papers present in Chapter 3.

Genetic Algorithms are based on the evolution of the species and the survival of the fittest individual. These algorithms start with a set of randomly generated solutions called individuals. The group of individuals existing at each iteration is called Population. At each iteration, the best individuals from the current population are taken and used to form a new population through reproduction. This is motivated by the hope that the new population will be better than the old one. These algorithms provide a global exploration of the solution space, avoiding convergence to a local minimum.

Particle Swarm Optimization is inspired by the social behavior of bird flocking or fish schooling. It starts with a set of randomly generated solutions called particles, spread throughout the search space. Each particle moves through the solution space at its own velocity, attracted by the best particle found in the population. After a few iterations, the particles are reunited around the best solution found so far, exploring the nearby solutions with the aim of refining and improving them. The algorithms provide less exploration of the solution space than the GA, providing a focused local search, with less computational cost.

1.4 Related work

Scheduling techniques have evolved together with computing systems' architectures. Initial studies about distributed systems were undertaken by Feitelson and Rudolph in [FR90]. Since then, many improvements have been developed that improved the architecture of these systems [TD01].

One of the most studied issues to improve the performance and efficiency of these architectures is job scheduling. Initially, traditional techniques treated the jobs in the waiting queue individually without considering the remaining jobs in the batch queue [BSB+01], thus limiting the scheduling opportunities for future allocations and decreasing overall system performance. However, some research has proposed techniques to improve the performance of the system by looking at future jobs. Shmueli et al. [SF05] proposed a backfilling technique in which later jobs are packaged to fill in holes and increase utilization without delaying the earlier jobs. Tsafrir et al. [TEF07] proposed a method to select the most suitable jobs to be moved forward based on system-generated response time predictions. These techniques, however, are based on the job arrival order, only moving jobs forward that accomplish specific deadline requirements.

More modern research studied the evaluation of all the jobs in the queue, treating them together, in order to find the optimal solution. Blanco et al. [BLCG11, BGLA12] proposed diverse techniques to determine the best scheduling of sets of job packages, proposing a new job execution order to minimize their overall execution time based on a Mixed-Integer programming model.

However, finding an optimal solution to the scheduling problems using this technique has a high computational cost due to the large amount of data to process. For this reason, nature-inspired meta-heuristics, such as Simulated Annealing (SA), Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), have emerged as effective techniques in complex large-scale environments in an attempt to obtain pseudo-optimal solutions in practical times.

GAs are well known for their robustness and have been applied successfully to

solve scheduling problems in a variety of fields. Zomaya and Teh [ZT01] used GAs in dynamic load balancing problems. Braun et al. [BSB+01] compared the efficiency of a simple GA-based scheduler and the well-known techniques such as MinMin, MinMax and Minimum Completion Time (MTC) algorithms. Carretero and Xhafa presented, in [CXA07], an extensive study of GAs to design efficient Grid schedulers where makespan and flowtime are minimized to include QoS in the solutions, but considering independent jobs without inter-cluster communications.

PSO is also utilized in scheduling techniques, as it provides similar results to GA but with a simpler and less costly algorithm, which allows a faster convergence to the near optimal solutions. Thanushkodi, in [TD11], presented a PSO to schedule tasks in a multiprocessor system in order to reduce the waiting times and execution time of jobs. Surendra, in [PSS15], presented a PSO scheduling technique focused on scheduling jobs in GRID systems.

Reaching the modern era, the society has acquired a better awareness of the environment and ecological sustainability of all technologic progress. Thus, energy consumption has become a great challenge in the field of high-performance computing. In order to reduce energy consumption, two primary methods are commonly used: switching off underutilized resources [CRS15, CAT+01, OLG08] and using voltage and frequency scaling (VFS) techniques [CTSB15, KKWZ15, KBK07]. Cocaña et al. [CRS15] presented a software tool that predicts the future node requirements using a machine-learning approach, and then stopping those that will not be required in the near future. Chae et al. [CAT+01] illustrate a method to determine the aggregate system load and the minimal set of computational resources that can process the workload. Orgerie et al., in [OLG08], present a three-step strategy based on a framework able to control the computing requirements by switching the unused nodes off, predicting their usage in order to switch them on again and finally aggregating some reservations in order to avoid frequent on/off cycles. Christobel et al., in [CTSB15], proposed an energy-aware scheduling approach for scientific workflows based on Particle Swarm Optimization and Dynamic Voltage Scaling.

While taking care of the environment and improving the performance of the system

resources are two opposing goals, the future of the optimization of these systems belongs to finding a good trade-off between both metrics. For this reason, many research studies have proposed Multi-Criteria optimization algorithms able to find a good solution that comprises multiple criteria. Kolodziej et al., in [KKWZ15], and Kim et al., in [KBK07], address independent batch scheduling in computational grids as a bi-objective global minimization problem with makespan and energy consumption criteria, using a VFS model directed by a GA-based grid scheduler.

Moreover, over the last few years, the hybridization of evolutionary algorithms has been used to take advantage of the strengths of these heuristic algorithms and overcome their shortcomings in terms of dispersion and convergence. D. Liu et al., in [LAH10], presented a cyclic swap method with GA and Simulation Annealing (SA) to minimize energy consumption in cloud datacenters. Moganarangan et al., in [MBB+16], combined ant colony optimization (ACO) and cuckoo search (CS) to minimize energy consumption in cloud computing environments. Sharma et al., in [SG16], designed a resource allocation and VM migration focused on energy saving and SLA violation combining GA and PSO.

In this work, different proposals are presented based on these algorithms. The first focus on optimizing one objective while the last one is a multi-objective hybrid method.

1.5 Document Structure

This PhD dissertation is organized as follows.

Chapter 1 introduces Cluster Federated Environments, describes the models used for the architecture, the execution time and the energy consumption. It also elaborates the scheduling problem in this kind of environments and presents a brief description of the most important techniques from the literature to solve this problem.

Chapter 2 proposes the problem statement of the work, and then the objectives of this dissertation and the milestones that guided the research are detailed. After that, it defines the methodology applied during the research process.

Chapter 3 presents the most relevant publications proposed during this work. Each publication consists of a scheduling technique that resolves the corresponding milestones presented in Chapter 2.

Chapter 4 presents a global discussion of the results obtained for each publication. It remarks the impact of the publication and discusses the future research opened up.

Chapter 5 presents the principal conclusions extracted from the PhD and sets out some of the possible future research lines.

Chapter 2

Methodology

The problem of scheduling applications in heterogeneous environments has been widely studied in the literature. There are many factors that have to be taken into account to improve the system performance and the user satisfaction such as application requirements, resource heterogeneity, system load, etc..

Optimization with these problems can be focused on many different criteria, and the most common can be encompassed in the following two main aspects: *improving the applications' performance from the user point of view* (i.e. makespan, flowtime, throughput, etc.) or *improving the system performance from the system administrator point of view* (i.e. energy consumption, system utilization, cost reduction, etc.). Also, these criteria can be addressed in isolation (single objective techniques), trying to find the best solution for the selected objective, or in conjunction of two or more objectives (multi-objective techniques) which has the added complexity of finding a good trade-off between the comprehended objectives based on the decision maker.

The vast amount of aspects to take into account when scheduling applications in Cluster Federated Environments, and the complexity of the problem when optimizing the different objectives, makes the scheduling problem an open research field. This PhD work aims to contribute to this by providing new scheduling techniques that improve the performance of the ones existing in the literature.

This Chapter introduces the problem statement, as it is the starting point of this PhD. The main objective is presented, followed by the proposed milestones and

roadmap that guided our steps during this research. Finally, the research methodology used during the preparation of this PhD is introduced and detailed.

2.1 Problem statement

The scheduling problem has been a constant mainstay in the Distributed Computing Research Group of the University of Lleida (GCD). The group has been actively working on solving this problem in different distributed architectures, ranging from Cluster systems to Cloud, since its formation.

The investigation in Cluster Federated Environments has produced two PhD works. The first one, Lérida in [Mon09] proposed a model and a system manager for a Cluster Federated Environment and stated the interest in finding an optimal scheduling of the jobs to increase its performance. Blanco in [BdF12] took the lead in the research and used the knowledge acquired by the group to develop exact models to find the optimal solution for the scheduling problem. These techniques were able to treat the applications as a package in order to obtain the best scheduling, based on a single objective function. However, due to the NP nature of the problem, the proposed models were only used for analytical and research purposes and were not practical for real environments. Blanco also proposed an ad-hoc heuristic technique based on the knowledge obtained through exact methods behavior, able to find good solutions with a low computational cost.

This PhD continues with the work carried out in the research group by proposing new sophisticated techniques to find near optimal solutions in a competitive computational time. We will make use of the well-known nature-based approximate heuristics Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) to generate single and multi-objective techniques.

2.2 Main objective

The main objective of this PhD is:

The study and proposal of techniques for scheduling applications in Federated Cluster Environments focused on the reduction of makespan (the time elapsed since the first job starts its execution until the completion of the last job) and the energy consumption during the applications execution.

The proposed techniques are based on evolutionary algorithms, a subgroup of approximate techniques that are known to provide great exploration of the solution space with low computational cost. The techniques proposed during this work aim to provide optimal solutions considering a mixture of different application job ordering options and performance metrics. To do so, the techniques will take into account the heterogeneity of the resources, the computing and communication requirements of the applications and the energy cost of executing these applications in the system.

2.3 Milestones

To achieve the main objective of this PhD, we propose the following set of milestones that divide the general objective into more specific goals. These milestones are organized following a roadmap that allows us to learn and overcome the complexity of the problem. It can be summarized as the study of the state of the art, optimization of the makespan, optimization of the energy consumption, in addition to multi-criteria optimization of multiple objectives.

- **M1: Exhaustive study of the state of the art in the field of scheduling in distributed and heterogeneous systems.**

  To start the research into the scheduling field, the first step was to perform an in-depth study of the literature and related works. We analyzed the diverse proposals related to scheduling in distributed systems, paying special attention to the ones that used approximate techniques. Some of the heuristics studied are later implemented in order to compare their performance with the proposed heuristics of this PhD Work.

- **M2: Optimization of the makespan**

One of the most commonly used metrics for optimizing the workload execution is the makespan, which consists of the time elapsed since the first job starts its execution until the last job ends. This metric is related to the user point of view aspect as a reduction in the makespan, and implies a faster execution of the jobs present in the workload. We addressed the optimization of this metric with the proposal of some techniques that take the workloads divided into sets of jobs and search for a near optimal ordering and allocation depending on the infrastructure characteristics.

- **M3: Optimization of the energy consumption**
  With this milestone, the goal is to propose a heuristics to reduce the energy consumed during the execution of a workload. This metric is related to the system administrator point of view, as a reduction in the energy consumption will lessen the expenses of maintaining the system and make it more environmentally sustainable. For this milestone, we analyzed the literature to find a model to predict the energy consumption in a Cluster Federation Environment. Unfortunately, none of the existing models resolved the problem. Thus, we proposed a new model to predict the energy consumption for Cluster Federated Environments and then presented some heuristics able to significantly reduce the energy consumption.

- **M4: Optimize both criteria using multi-objective algorithms**
  Finally, having analyzed both optimization criteria individually (makespan and energy), we accomplished the objective of joining them using a multi-criteria algorithm, dealing with the problems related to the multi-objective optimization. When reducing both objectives at the same time, the algorithm is forced to search for a trade-off result, as the reduction in one of the objectives individually can produce an increase in the other. Moreover, we analyzed different combinations of evolutionary algorithms in order to obtain good solutions, reducing the computational cost.

2.4 Research methodology

In Computer Science there is no standard research methodology defined, as this field is composed as a merge of different scientific and engineering fields, each one with its own scientific method. For this reason, many scientific methods have been proposed to deal with its complex nature. This PhD work follows the directives of the hypothetico-deductive method, adapted to the Computer Science field proposed by Adrion in [Adr93]. In this method, the research is divided into 4 different parts, that can be repeated depending on its results.

1. **Observe the existing solutions.**

   This part consists of an in-depth analysis of the related work done in the same or similar fields. This part is necessary in order to avoid to working on a solved problem and to produce updated and quality research. Moreover, the study of other scientific fields can provide new ideas to be applied in the field. We analyzed the models already proposed about the infrastructure, execution time, energy consumption and also the exact and approximate techniques to solve scheduling problems, and obtained a great deal of information on how to treat the problem for different infrastructures, application requirements and circumstances.

2. **Propose better solutions.**

   With the wide knowledge gained during the study of the related work, it is time to analyze the existing solutions to the purpose of providing them with improvements. In this part, we also adapted some techniques in the literature to Federated Cluster Environments, and proposed innovative improvements to increase the performance of our proposals.

3. **Build or develop new solutions**

   In this part, we proposed and implemented new scheduling techniques. When problems appear, we repeat step 2 to find a new solution. This step can also lead us to repeat step 1 to analyze in the literature how to solve a specific

problem. This part is where the proposed techniques are implemented.

4. **Measure and analyze the new solution**

   Finally, the solutions are tested and evaluated with the literature heuristics in order to compare the results. If the results are not good enough, the previous steps are repeated until the solutions are improved.

Chapter 3

Papers

In this chapter we set out our proposals to solve the scheduling problem in Federated Cluster Environments. The main contributions to address each of the milestones indicated in Chapter 2 are also discussed.

3.1 Multi-Criteria Genetic Algorithm Applied to Scheduling in Multi-Cluster Environments

In this paper, a Genetic Algorithm (GA) meta-heuristic was presented to optimize the scheduling of jobs in heterogeneous environments. The proposed meta-heuristic focused on optimizing the execution of different job sets from the user point of view, taking the makespan and the flowtime as the optimization metric, by tackling the goals established in milestone M2.

This proposal is able to consider both the computational and the communication resources, following the multi-cluster model presented by Lerida et al. [Mon09].

The proposal considers the workload as a set of jobs to be scheduled. Treating them in packages increases the knowledge of the future availability of the system and provides more scheduling opportunities to perform better allocations.

3.1.1 Contributions to the state of the art

The following paper proposes a Genetic Algorithm technique, called GA-MF for Genetic Algorithm Makespan and Flowtime, dealing with the scheduling problem in heterogeneous environments with the objective of reducing makespan and flowtime (the overall execution time of the jobs). Thanks to the ability of evolutionary algorithms to explore the solution space exhaustively with reasonable execution costs, this algorithm is able to improve the solutions of traditional heuristics, obtaining a close to optimal solution, without the high computational costs implied by the usage of exact methods.

The proposal is implemented to work in a Federated Cluster Environment simulated using the GridSim simulator. The workloads used in the experimentation are extracted from real traces obtained from the HPC2N workload found in the Parallel Workload Archive site managed by Felteison [Fei14].

3.1.2 Paper 1: Multi-Criteria Genetic Algorithm Applied to Scheduling in Multi-Cluster Environments

The paper presented can be found in the following publication:

Authors: Eloi Gabaldon, Josep Lluis Lerida, Fernando Guirado, Jordi Planes

Title: Multi-criteria genetic algorithm applied to scheduling in multi-cluster environments

Journal: The Journal of Simulation

Volume: 9 **Issue:** 4 **Pages:** 287-295

Year: 2015

Impact Index (SCI/SSHI/AHCI): 1.164 **Citations:** 5

Quartile and Subject(SCI/SSHI/AHCI): OPERATIONS RESEARCH & MANAGEMENT SCIENCE, 45 of 82 (Q3)

ISSN: 1747-7778

DOI: 10.1057/jos.2014.41

3.2 Particle Swarm Optimization Scheduling for Energy Saving in Cluster Computing Heterogenous Environments

In this paper, a Particle Swarm Optimization (PSO) meta-heuristic was presented to perform the scheduling of jobs in heterogeneous environments. The proposed meta-heuristic focused on reducing the energy consumed by the system when executing the jobs and facing the goals posed in milestone M3.

The energy consumed by the system has become a critical issue in the last decade for the providers of distributed computing systems, as the reduction of the energy implies a significant reduction in costs plus a reduction in the carbon footprint, making the system less expensive and more sustainable.

In this work, we focused our proposal on the use of another popular evolutive technique, known as PSO. This technique has the advantage of simplicity and effectiveness, making it ideal for dealing with such complex problems as scheduling and task allocation. The algorithm combines a great exploration of the solution space and a fast convergence to near optimal solutions. The proposal also considers both the computational and the communication resources and treats the jobs with packages. Moreover, it introduces the use of a blacklist of forbidden nodes that allows the creation of new opportunities for energy saving allocations, providing a reservation mechanism of computing nodes for future jobs. The blacklist is one of the main contributions of this PhD, as it provided a simplification in the problem representation and the ability to increase the complexity of the problems to be solved and to improve of the results obtained.

3.2.1 Contributions to the state of the art

The following paper proposes a Particle Swarm Optimization technique (PSO) to solve the scheduling problem in Heterogeneous Environments with the aim of efficiently reducing the energy consumed and providing an improvement in the system

maintenance costs. The algorithm also incorporated a blacklist of forbidden nodes in order to provide a reservation mechanism for the future execution of jobs.

Finally, the proposal was evaluated with three real workload traces extracted from the Parallel Workload Archive [Fei14], each with different characteristics, and the results showed the effectiveness of the proposed method, obtaining a high reduction in energy consumption compared with the other literature techniques, on the workloads tested.

3.2.2 Paper 2: Particle Swarm Optimization Scheduling for Energy Saving in Cluster Computing Heterogenous Environments

The paper presented can be found in the following publication:

Authors: Eloi Gabaldon, Fernando Guirado, Josep Lluis Lerida, Jordi Planes

Title: Particle Swarm Optimization scheduling for Energy Saving in Cluster Computing Heterogenous Environments

Conference: 3rd International Workshop on Energy Management for Sustainable Internet-of-Things and Cloud Computing

Citations: 5

Published: IEEE CPS

Year: 2016

DOI:10.1109/W-FiCloud.2016.71

3.3 Blacklist Muti-objective Genetic Algorithm for Energy Saving in Heterogeneous Environments

In this paper, a Multi-Objective Genetic Algorithm was presented to solve the job scheduling problem, focused on optimizing both the makespan and the energy consumption.

Unlike the previous works, where the optimization only focused on one criterium and there is only one optimal solution, in multi-criteria optimization there can be more than one optimal solution, as the best solution obtained for one objective may not be optimal for the other one. For this reason, the algorithm has to take into account both objectives concurrently and search for the optimal solutions that are a combination of both objectives.

In this proposal, the optimization of both objectives was obtained by using a Non-dominated Sorted Genetic Algorithm II (NSGAII) that ensures the optimization of both objectives, returning the Pareto Frontier of the optimal solutions. However, in the experimental study, this frontier was composed of a single solution, as both objectives converged on the same solution.

3.3.1 Contributions to the state of the art

The following paper proposes a Multi-Objective Genetic Algorithm (MOGA) metaheuristic to optimize both the makespan and the energy consumption when scheduling a workload of jobs in a heterogeneous Cluster Federation environment.

This technique uses some of the features of the previous techniques presented, being able to consider the computational and the communication resources by following the model presented by Lerida et al [Mon09], dealing with sets of jobs and incorporating a blacklist of computational nodes used by the previous proposals to better optimize the energy consumption.

The new proposal was evaluated with three real workload traces extracted from the Workload Parallel Archive [Fei14]. The results showed the effectiveness of the

proposed method, providing solutions that improved the performance not only of the makespan objective but also of the energy consumption, finding a near optimal tradeoff between these objectives. However, despite the fact that the complexity of a GA technique was severally reduced thanks to the representation using a blacklist, some experimentation with new evolutionary techniques can be proposed to reduce the computational cost even more.

3.3.2 Paper 3: Blacklist Muti-objective Genetic Algorithm for Energy Saving in Heterogeneous Environments

The paper presented can be found in the following publication:

Authors: Eloi Gabaldon, Josep Lluis Lerida, Fernando Guirado, Jordi Planes

Title: Blacklist Multi-objective Genetic Algorithm for Energy Saving in Heterogeneous Environments

Journal: The Journal of Supercomputing

Volume: 73 **Issue:** 1 **Pages:** 354-369

Year: 2017

Impact Index(SCI/SSHI/AHCI): 1.088

Citations: 4

Quartile and Subject (SCI/SSHI/AHCI): COMPUTER SCIENCE, THEORY & METHODS, 47 of 105 (Q2)

ISSN: 0920-8542

DOI: 10.1007/s11227-016-1866-9

3.4 Energy Efficient Scheduling on Heterogeneous Federated Clusters using a Fuzzy Multi-Objective Meta-heuristic

In previous papers, we presented different heuristics based on GA or PSO evolutionary algorithms. The main difference between them is that, while GA is able to broadly explore the solution space, PSO gives a faster convergence to the solution. Considering this, in this paper a hybrid heuristic was presented that merges the PSO and GA techniques.

The algorithm also includes a Fuzzy logic operator that helps the promotion of the best solutions at each iteration. The optimization heuristic uses a multi-objective PSO algorithm that incorporates GA and Fuzzy methods as one of its evolutionary operators.

3.4.1 Contributions to the state of the art

This paper proposes a Multi-Objective Hybrid Meta-heuristic based on PSO and GA with the objective of optimizing both makespan and energy consumption of the system when executing a workload in a Federated Cluster Environment. The hybrid meta-heuristic was able to find good solutions taking advantage of the broad exploration provided by GA while reducing its computational cost thanks to the simplicity and speed of PSO. The technique also uses the representation of the cluster availability based on a weighted blacklist to consider resource heterogeneity, communication and application requirements that proved very successful in previous proposals. It also includes a fuzzy operator on the allocation process to promote the solutions with better improvements.

The proposal was evaluated with three real workload traces extracted from the Parallel Workload Archive [Fei14] and the results showed the effectiveness of the proposed method, providing solutions that improved the performance compared with other well-known techniques in the literature with low computational costs. This

technique was able to reduce the computational cost, obtaining similar results to MOGA.

3.4.2 Paper 4: Energy Efficient Scheduling on Heterogeneous Federated Clusters using a Fuzzy Multi-Objective Meta-heuristic

The paper presented can be found in the following publication:

Authors: Eloi Gabaldon, Sergi Vila, Fernando Guirado, Josep Lluis Lerida, Jordi Planes,

Title: Energy Efficient Scheduling on Heterogeneous Federated Clusters using a Fuzzy Multi-Objective Meta-heuristic

Conference: IEEE International Conference on Fuzzy Systems

Rank conference: CORE A (2017)

Published: IEEE CPS

Year: 2017

DOI: 10.1109/FUZZ-IEEE.2017.8015589

Chapter 4

Global discussion of results

The following chapter presents the general conclusions and a brief discussion of the results obtained for each paper described in Chapter 3.

4.1 Multi-Criteria Genetic Algorithm Applied to Scheduling in Multi-Cluster Environments

In order to optimize the makespan criteria, a GA-based scheduling meta-heuristic for large-scale multi-cluster environments called GA-Makespan&Flowtime (GA-MF) was presented. The algorithm used an objective function that combines both the makespan and the flowtime metrics in a single objective function in order to obtain a fair scheduling that reduces the whole workload execution time but also without penalizing individual jobs. This technique also considers the computational and communication heterogeneity of the Federated Cluster Environment.

The experimentation was performed using the GridSim Simulator, adapted to Cluster Federated Environments, and real job traces from the HPC2N workloads. The results showed that, by using our proposed GA-MF technique, we can obtain the best flowtime, directly related to the QoS parameter, obtaining greater reduction of the makespan compared with the most common techniques used in the literature. The results confirmed the performance improvement using population-based meta-

heuristics in order to optimize the scheduling problem.

4.2 Particle Swarm Optimization Scheduling for Energy Saving in Cluster Computing Heterogenous Environments

In this paper we focused on the reduction of the energy footprint. When addressing the optimization of energy consumption, a method was proposed to limit the allocation resources available for the jobs in order to create new scheduling opportunities and provide energy savings. Our research resulted in the incorporation into the scheduling technique of a novel blacklist representation for forbidden nodes. The blacklist representation provided the algorithm with a resource reservation mechanism for further jobs in the queue.

In this paper, a model was also presented to predict the energy consumption of the system during the execution of the workload. The model took into consideration the energy consumed by the computing nodes in their idle and computing states.

With these new features, a Particle Swarm Optimization (PSO) was presented, called PSO-EA, which stands for PSO Energy Aware. The performance of the new proposal was evaluated by simulation and compared with other techniques in the literature. The empirical results showed that PSO-EA was able to significantly reduce the energy consumption and also showed a low sensitivity to workload variations, obtaining the solutions with a low computational cost provided by the PSO.

4.3 Blacklist Muti-objective Genetic Algorithm for Energy Saving in Heterogeneous Environments

At this point of the research, two different single objective techniques had been presented, demonstrating that the use of approximated heuristics could provide really good solutions to the scheduling problem. However, one of the objectives of this PhD

was to obtain an optimal solution considering both objectives simultaneously. Thus, a Multi-Objective Genetic Algorithm (MOGA) was proposed. Taking advantage of the previous knowledge, the algorithm also incorporates the blacklist of forbidden nodes representation with the aim of increasing the chances of an energy consumption reduction.

The experimental study was carried out by simulation and was conducted with three real workload traces in a heterogeneous Federated Cluster Environment. The MOGA results were compared with other techniques in the literature and demonstrated its capability to obtain better results while optimizing both makespan and energy efficiency.

This technique was a great contribution to the field of scheduling in Cluster Federated Environments with heterogeneous resources and co-allocation. However, the technique has a high computational cost when experimenting with some environments or workloads, opening up a new research line pursuing similar results with lower computational costs.

4.4 Energy Efficient Scheduling on Heterogeneous Federated Clusters using a Fuzzy Multi-Objective Meta-heuristic

A fuzzy multi-objective hybrid scheduling meta-heuristic was proposed, called MPSO-FGA, which stands for Multi-objective PSO with Fuzzy GA. This technique optimizes both makespan and energy consumption metrics but with a low computational cost. To do so, the new proposal combines the great exploration of GA with the fast convergence of PSO. Additionally, a fuzzy operator was added to the meta-heuristic to promote the best solutions found during the algorithm. This proposal also incorporates the blacklist of forbidden nodes representation proposed in the previous research works.

The proposal was evaluated using three real workloads and using a heterogeneous

federated cluster environment. Although the results obtained were slightly higher than the MOGA algorithm, 10% for energy and 20% for makespan, its computational cost was two orders of magnitude lower and obtained better global results than the other literature techniques, 15% for energy and 10% for makespan.

These results open up a new perspective for the use of more sophisticated multi-objective techniques, providing good results in short time in increasingly complex environments.

Chapter 5

General conclusions and future directions

In this chapter, the research work presented in this PhD is summarized and the main contributions are highlighted. It also discusses open research problems in the area and outlines different future research directions.

5.1 Conclusions

This PhD proposed multiple evolutive techniques to perform the scheduling of jobs in heterogeneous Cluster Federated Environments. These techniques use a job execution model that takes into account both the computation power of the resources and the communication links availability. Our proposals focus on optimizing the execution of sets of applications, taking as objectives the minimization of the global execution time, the minimization of energy consumption or, by using advanced multi-objective techniques, both objectives.

The proposed techniques were tested using the GridSim simulator, adapted to work in Cluster Federated Environments and executing different real traces workloads extracted from the Parallel Workload Archive Site created by Feitelson in [Fei14]. The results showed improvements in the solutions obtained when compared with other literature techniques.

Taking as a reference the milestones presented in chapter 2, below is the list of papers published, together with the goal achieved. The contributions included in Chapter 3 are highlighted in bold:

M2: Use an approximate heuristic to optimize the makespan

This milestone focused on the optimization of makespan, related to user experience and QoS. The first proposal was a Genetic Algorithm with makespan as the fitness function. The technique provided a great reduction in this metric compared with other simple techniques in the literature, demonstrating that the use of evolutionary algorithms could be successfully applied to solve the scheduling problem.

In the solutions obtained, it was observed that some applications were severely penalized, by moving them to the back of the queue, increasing their waiting times. Thus, we proposed GA-MF, which introduces the Flowtime to the fitness function as a weighted objective. This metric is related to the QoS and takes into consideration the reduction in the waiting time, obtaining in this way a fairer scheduling for the applications.

Although our proposals show good results for some of the instances of the problem, they do not work correctly with large packages of jobs and present a high computation cost. The research continued by searching for simpler ways to represent the problem as well as other evolutionary techniques in order to provide faster convergence of the algorithms.

The contributions published in this milestone are described below:

- Eloi Gabaldon, Fernando Guirado and Josep L. Lerida. Genetic Meta-Heuristics for Batch Scheduling in Multi-Cluster Environments. *Proceedings of the 13th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2013*

- Eloi Gabaldon, Josep L. Lerida, Fernando Guirado, Jordi Planes. Slowdown-Guided Genetic Algorithm for Job Scheduling in Federated Environments. *In-*

ternational Conference on Nature of Computation and Communication. ICTCC 2014, volume 144, pp 181-190, DOI: 10.1007/978-3-319-15392-6_18

- **Eloi Gabaldon, Josep L. Lerida, Fernando Guirado, and Jordi Planes. Multi-Criteria Genetic Algorithm Applied to Scheduling in Multi-Cluster Environments. *The Journal of Simulation, 2015, Volume 9, Issue 4, pp 287:295*, DOI: 10.1057/jos.2014.41**

M3: Use approximate heuristics to optimize energy consumption

This milestone focused on reducing the energy consumption of the system. To do so, a method was required to limit the allocation resources available for the jobs in order to create new scheduling opportunities and provide energy savings. With this requirement, and also aiming to simplify the representation, we developed a new representation able to solve both problems.

The new representation had to meet to the following requirements:

- Avoid the representation of the same solutions in different individuals
- Avoid the representation of unfeasible solutions
- Allow a reservation method for the next jobs in the queue

For this purpose, a representation was proposed including a blacklist of forbidden nodes. The new problem representation was simpler to analyze for evolutionary algorithms and provided more chances to save energy.

A Genetic Algorithm (GA-EA) was presented, provided with the blacklist representation, focused on reducing energy consumption. The results showed that the algorithm was able to reduce the energy consumption of the system compared with other techniques in the literature. Also, the computational cost of the algorithm was lower, allowing the technique to work with bigger job packages.

Besides this, in order to explore other approximate techniques that provide less computational cost, a Particle Swarm Optimization (PSO-EA) technique was pro-

posed. This technique was able to obtain similar results to GA-EA, but with a lower computational cost.

The contributions published in this milestone are presented below:

- Eloi Gabaldon, Fernando Guirado, Josep Lluis Lerida and Jordi Planes. Black-List Genetic Algorithm Scheduling for Energy Saving in Heterogenous Environments. *Proceedings of the 16th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2016*

- **Eloi Gabaldon, Fernando Guirado, Josep Lluis Lerida, Jordi Planes. Particle Swarm Optimization Scheduling for Energy Saving in Cluster Computing Heterogenous Environments.** ***2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Vienna, 2016, pp. 321-325. doi: 10.1109/W-FiCloud.2016.71***

M4: Optimize both criteria using multi-objective algorithms

To continue with the research process, we proposed techniques to optimize both, the makespan and the energy consumption, at the same time using multi-criteria techniques.

First, a Genetic Algorithm was presented, based on multi-criteria algorithm NSGAII [DPAM02]. This technique was able to optimize both the Makespan and Energy Consumption criteria, providing solutions very close to optimal. However, as observed in the previous proposals, the GA has a high computational cost.

Following the experience and the great success of the previously presented PSO technique, a new experimental algorithm (MPSO-FGA) was presented, using a hybrid proposal mixing the PSO and the GA techniques. This technique incorporated a set of Fuzzy rules to help with the convergence to the algorithm. The proposal reproduced the great results of the MOGA, reducing its computational cost and opening up a new line of research on the Hybrid algorithms to be studied as future work.

The contributions in this milestone are presented in the following publications:

- **Eloi Gabaldon, Josep Lluis Lerida, Fernando Guirado, Jordi Planes. Blacklist muti-objective genetic algorithm for energy saving in heterogeneous environments.** ***The Journal of Supercomputing, 2017, Volume 73, pp 354:369.*** **doi:10.1007/s11227-016-1866-9**

- **Eloi Gabaldon, Sergi Vila, Fernando Guirado, Josep Lluis Lerida and Jordi Planes. Energy Efficient Scheduling on Heterogeneous Federated Clusters using a Fuzzy Multi-Objective Meta-heuristic.** ***Proceedings of the 2017 IEEE Conference on Fuzzy Systems, FUZZ-IEEE 2017*****, DOI: 10.1109/FUZZ-IEEE.2017.8015589**

5.2 Future work

During the preparation of this PhD and its current contributions to the scheduling of applications in the Cluster Federated Environments field, several open research issues emerged to continue to provide improvements to this area.

Implementation in Real Environments

This PhD work has been developed using the GridSim Simulator [Gar13], which has been highly recognized by the scientific community. The next step in this research could be to implement these proposals in real environments, in order to benefit from their good performance.

Analyze and evaluate the proposals applied to scheduling in Cloud Systems

Nowadays, with the improvements of the high-speed interconnection networks, and the globalization of the use and accessibility of computing resources, Cloud systems have emerged as an opportunity to acquire computational resources without having to invest in physical machines. In these environments, the architecture has many differences to the one studied in this PhD, involving virtual machines (VM) that can

migrate from one physical machine to another during their execution and applications that, instead of being high in resource requirements, are common services that must run for undefined periods of time.

All these characteristics force us reconsider the models of the system and the way in which the scheduling is applied. However, the background problem is the same, making the knowledge acquired in this PhD very useful in this kind of environments.

Use of deep learning to select a scheduling technique

Despite the great results obtained by the optimization techniques proposed in this PhD, sometimes, when the workloads have specific characteristics, some of the proposals were able to obtain the same results, but with different computational costs. For this reason, it could be interesting to generate a deep learning pre-processing of the workload, taking into account the application characteristics and the environment, in order to decide which technique would be the most suitable in each situation.

Consider the machines overhead

This PhD work takes in consideration the energy consumption when performing the execution of applications. However, the proposed techniques try to use the machines intensively and this can produce an overhead in the machine processor that can evolve to a machine failure. It could be interesting to study the possibility introducing a worn out factor into the energy model. In this way, the techniques could also take into account the misuse of the system as an optimization objective thus enlarging the functional life of the machines.

Bibliography

[Adr93] W. Richards Adrion. Research methodology in software engineering. *ACM SIGSOFT Software Engineering Notes, Summary of the Dagstuhl Workshop on Future Directions in Software Engineering*, 18(1):36–37, 1993.

[BB01] Rajkumar Buyya and Mark Baker. Emerging technologies for multicluster/grid computing. In *Proceedings of the 2001 IEEE International Conference on Cluster Computing (CLUSTERí01)*, pages 457–458, 2001.

[BdF12] Héctor Blanco de Frutos. *Clusterización de aplicaciones paralelas para su planificación en entornos de cómputo multi-cluster*. PhD thesis, Universitat de Lleida, 2012.

[BE07] Anca I.D. Bucur and Dick H.J. Epema. Scheduling policies for processor coallocation in multicluster systems. *Transactions on Parallel and Distributed Systems, IEEE*, 18(7):958–972, 2007.

[BGLA12] Hector Blanco, Fernando Guirado, Josep L. Lerida, and Victor M. Albornoz. MIP model scheduling for BSP parallel applications on multicluster environments. In *2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), IEEE*, pages 12–18, 2012.

[BLCG11] Hector Blanco, Josep L. Lerida, Fernando Cores, and Fernando Guirado. Multiple job co-allocation strategy for heterogeneous multi-cluster systems based on linear programming. *The Journal of Supercomputing*, 58(3):394–402, 2011.

[BLGL12] Hector Blanco, Jordi Llados, Fernando Guirado, and Josep L. Lerida. Ordering and allocating parallel jobs on multi-cluster systems. In *12th International Conference Computational and mathematical methods in Science and Engineering(CMMSE)*, pages 196–206, 2012.

[BSB+01] Tracy D. Braun, Howard J. Siegel, Noah Beck, Lasislau L. Bölöni, Muthucumara Maheswaran, Albert I. Reuther, James P. Robertson, Mitchell D. Theys, Bin Yao, Debra Hensgen, and Richard F. Freund. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed Computing.*, 61(6):810–837, 2001.

[BSS+95] Donald J Becker, Thomas Sterling, Daniel Savarese, John E Dorband, and Charles V Packer. Beowulf: A parallel workstation for scientific computation. In *Proceedings of the 24th Internatio- 198 BIBLIOGRAFÍA [SSN99] [STF99] [Sto07] [SvANS00] [SW02] [Tan05] [TD01] [TEF07] [TOP] [TRA+05] Proceedings of the 24th International Conference on Parallel Processing*, 1995.

[BV98] Egon Balas and Alkis Vazacopoulos. Guided local search with shifting bottleneck for job shop scheduling. *Management Science*, 44(2):262–275, 1998.

[CAT+01] Jeffrey S. Chase, Darrell C. Anderson, Prachi N. Thakar, Amin M. Vahdat, and Ronald P. Doyle. Managing energy and server resources in hosting centers. *ACM SIGOPS Operating Systems Review*, 35(5):103–116, 2001.

[CRS15] Alberto Cocaña, José Ranilla, and Luciano Sánchez. Energy-efficient allocation of computing node slots in HPC clusters through parameter learning and hybrid genetic fuzzy system modeling. *The Journal of Supercomputing*, 71(3):1163–1174, 2015.

[CTSB15] M Christobel, S Tamil Selvi, and Shajulin Benedict. Efficient scheduling of scientific workflows with energy reduction using novel discrete particle swarm optimization and dynamic voltage scaling for computational grids. *The Scientific World Journal*, 2015, 2015.

[CXA07] Javier Carretero, Fatos Xhafa, and Ajith Abraham. Genetic algorithm based schedulers for grid computing systems. *International Journal of Innovative Computing, Information and Control*, 3(6):1–19, 2007.

[DPAM02] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Transactions on Evolutionary Computation, IEEE*, 6(2):182–197, Apr 2002.

[EHS+02] Carsten Ernemann, Volker Hamscher, Uwe Schwiegelshohn, Ramin Yahyapour, and Achim Streit. On advantages of grid computing for parallel job scheduling. In *International Symposium on Cluster, Cloud and Grid Computing (CCGRID),IEEE*, pages 39–39, 2002.

[ELvD+96] Dick. H. J. Epema, M. Livny, R. van Dantzig, X. Evers, and J. Pruyne. A worldwide flock of condors: Load sharing among workstation clusters. *Future Generations Computing Systems*, 12(1):53–65, 1996.

[Fei14] Dror Feitelson. Parallel workloads archive. http://www.cs.huji.ac.il/labs/parallel/workload, Accessed: 2014.

[FK03] Ian Foster and Carl Kesselman. *The Grid 2: Blueprint for a new computing infrastructure*. Elsevier, 2003.

[FR90] Dror G. Feitelson and Larry Rudolph. Distributed hierarchical control for parallel processing. *Journal Computer*, 23(5):65–77, 1990.

[Gar13] Saurabh Kumar Garg. Gridsim simulation framework. http://www.buyya.com/gridsim, Accessed: 2013.

[JLPS05] William M. Jones, Walter B. Ligon, Louis W. Pang, and Dan Stanzione. Characterization of bandwidth-aware meta-schedulers for co-allocating jobs across multiple clusters. *The Journal of Supercomputing*, 34(2):135–163, 2005.

[KBK07] Kyong Hoon Kim, Rajkumar Buyya, and Jong Kim. Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters. In *International Symposium on Cluster, Cloud and Grid Computing (CCGRID), IEEE*, volume 7, pages 541–548, 2007.

[KKWZ15] Joanna Kołodziej, Samee Ullah Khan, Lizhe Wang, and Albert Y. Zomaya. Energy efficient genetic-based schedulers in computational grids. *Concurrency and Computation: Practice and Experience*, 27(4):809–829, 2015.

[LAH10] Hongbo Liu, Ajith Abraham, and Aboul Ella Hassanien. Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm. *Future Generation Computer Systems*, 26(8):1336–1343, 2010.

[LLM88] Michael J. Litzkow, Miron Livny, and Matt W. Mutka. Condor-a hunter of idle workstations. In *8th International Conference on Distributed Computing Systems (ICDCS)*, 1988.

[LLQ09] Yu Li, Yi Liu, and Depei Qian. A heuristic energy-aware scheduling algorithm for heterogeneous clusters. In *15th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 407–413, 2009.

[May15] Jordi Vilaplana Mayoral. *Management of Cloud systems applied to eHealth*. PhD thesis, Universitat de Lleida, 2015.

[MBB+16] N Moganarangan, RG Babukarthik, S Bhuvaneswari, MS Saleem Basha, and P Dhavachelvan. A novel algorithm for reducing energy-consumption in cloud computing environment: Web service computing approach. *Journal of King Saud University-Computer and Information Sciences*, 28(1):55–67, 2016.

[Mon09] Josep Lluís Lérida Monsó. *Meta-Planificador Predictivo para Entornos Multicluster no Dedicados*. PhD thesis, Universitat Autonoma de Barcelona, 2009.

[Mos98] Gur Mosheiov. Multi-machine scheduling with linear deterioration. *INFOR: Information Systems and Operational Research*, 36(4):205–214, 1998.

[NLYW05] Vijay K. Naik, Chuang Liu, Lingyun Yang, and Jonathan Wagner. Online resource matching for heterogeneous grid environments. In *International Symposium on Cluster, Cloud and Grid Computing (CCGRID), IEEE*, pages 607–614, 2005.

[OBBB15] Karima Oukfif, Lyes Bouali, Samia Bouzefrane, and Fatima Boumghar. Energy-aware dpso algorithm for workflow scheduling on computational grids. In *2015 3rd International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 651–656. IEEE, 2015.

[OLG08] Anne-Cécile Orgerie, Laurent Lefevre, and Jean-Patrick Gelas. Save watts in your grid: Green strategies for energy-aware framework in large scale distributed systems. In *14th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pages 171–178, Dec 2008.

[PSS15] Surendra Kumar Patel, Anil Kumar Sharma, and Anurag Seetha. Implementing job scheduling to optimize computational tasks in grid computing using pso. In *Proceedings on National Conference on Knowledge, Innovation in Technology and Engineering (NCKITE)*, pages 20–24, 2015.

[SCJG00] Quinn Snell, Mark Clement, David Jackson, and Chad Gregory. The performance impact of advance reservation meta-scheduling. In *Workshop on Job Scheduling Strategies for Parallel Processing*, pages 137–153, 2000.

[SF05] Edi Shmueli and Dror G. Feitelson. Backfilling with lookahead to optimize the packing of parallel jobs. *Journal of Parallel and Distributed Computing*, 65(9):1090–1107, 2005.

[SG16] Neeraj Sharma and Ram Mohana Guddeti. Multi-objective energy effi-

cient virtual machines allocation at the cloud data center. *IEEE Transactions on Services Computing*, 2016.

[SHM97] David B Skillicorn, Jonathan MD Hill, and William F McColl. Questions and answers about BSP. *Scientific Programming*, 6(3):249–274, 1997.

[Sto07] Heinz Stockinger. Defining the grid: a snapshot on the current view. *The Journal of Supercomputing*, 42(1):3–17, 2007.

[TD01] Thyagaraj Thanalapati and Sivarama Dandamudi. An efficient adaptive scheduling scheme for distributed memory multicomputers. *IEEE Transactions on Parallel and Distributed Systems*, 12(7):758–768, 2001.

[TD11] K Thanushkodi and K Deeba. A new improved particle swarm optimization algorithm for multiprocessor job scheduling. *International Journal of Computer Science and Issues*, 8(4), 2011.

[TEF07] Dan Tsafrir, Yoav Etsion, and Dror G. Feitelson. Backfilling using system-generated predictions rather than user runtime estimates. *IEEE Transactions on Parallel and Distributed Systems*, 18(6):789–803, 2007.

[VMT+15] Jordi Vilaplana, Jordi Mateo, Ivan Teixidó, Francesc Solsona, Francesc Giné, and Concepció Roig. An SLA and power-saving scheduling consolidation strategy for shared and heterogeneous clouds. *The Journal of Supercomputing*, 71(5):1817–1832, 2015.

[ZT01] Albert Y. Zomaya and Yee-Hwei Teh. Observations on using genetic algorithms for dynamic load-balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(9):899–911, 2001.