Distributed Grid Brokers Architecture Using Web-Services

Anatoly Petrenko[1], Sergiy Svistunov[2], Pavlo Svirin[1]

[1] *System Design department, NTUU KPI, 37 Peremohy ave., Kyiv, Ukraine*
[2] *Bogolyubov Institute for Theoretical Physics, 14-b, Metrolohichna str., Kyiv, Ukraine*

petrenko@cad.kiev.ua, svistunov@bitp.kiev.ua, paul.svirin@gmail.com

Abstract. *In order to provide users with performance and task execution effectivity GRID has to implement an effective brokering algorithm. The main goal of such load balancing in GRID is to decrease the overall execution time and make utilization of the computing resources effective. In this work the modern approach to building metaschedulers in Grid segments is discussed.*

Keywords

Grid, load balancing, scheduler, broker, web-services.

1 Introduction

Despite the load balancing algorithms in computing resources in GRID being studied for a long time and despite the availability of many ready algorithmic decisions as well as software implementations, the intensive development of GRID technologies and improvement of middleware constantly actualizes the problem of load balancing and the interest towards research activities in this area is not decreasing. The main purpose of such load balancing in GRID is to decrease the overall execution time for the user's task and ensure utilization efficiency of the computing resources.

Ukrainian National GRID (UNG) infrastructure is made by the use of ARC (Advanced Resource Connector) middleware also known as project NorduGrid [3].

In ARC both 0.8 version and new ARC 2.0 version use maximum decentralization as the main principle therefore the personal broker is installed on every workbench of the GRID network user. The broker's function is to opt for the best resource to execute the user's task in the GRID network.

Currently in UNG the random selection of the resource is used, however it does not take into account the current state of the existing resources. For more efficient distribution of load among the resources personal brokers which take into account both the current state of the resources and the load balancing policy should be developed. It should be emphasized that Nordugrid ARC package contains the simplest policies therefore the suggested methods can be used not only in UNG but also for other segments and virtual organizations having specific and general tasks.

2 Problem definition for UNG

The situation in UNG can be defined as Many-task computing paradigm. This paradigm aims to bridge the gap between two computing paradigms, high throughput computing and high performance computing. Many task computing denotes high-performance computations comprising multiple distinct activities, coupled via file system operations [11].

The specifics of Ukrainian GRID infrastructure is the following:

- 38 clusters with low computational performance [13];
- Only 2 high computational performance resources are available;
- All resources are managed by ARC;
- Various calculation subjects: molecular dynamics, physics, chemistry, astronomy etc., a high number of virtual organizations.

Specifics of brokers in Nordugrid ARC:

- Availability of only simplified policies for tasks distribution

- The system is targeted at ATLAS experiment data processing with prevailing short tasks having small amounts of data. The broker that draws a conclusion regarding the target resource taking into account the amount of required data in the computational resource cache was developed specifically for this experiment. In such way it decreases the data transfer time.

Therefore Ukrainian segment lacks brokers suitable for efficient distribution of tasks of all categories.

In reality the tasks that require 10-30 processors are sent to the cluster of the Cybernetics institute and they await for days in a queue to be executed. Shorter tasks can also be directed there and also wait in queue.

Hence the goal of the optimal broker for UNG is:

- Directing shorter tasks to weaker resources
- Directing longer tasks to more powerful resources.

3 Algorithm

In order to optimize the task distribution we've used resource selection using earliest start criteria. The similar broker that uses resource queue length is already present in Nordugrid ARC. Unfortunately it cannot predict the approximate start time.

The algorithm steps are the following:

1. Query a service which returns estimation start time for a task being scheduled;
2. Deliver a task to a resource with the closest start time.

In order to test the algorithm we use a standard file metacentrum.mwf which comes together with Alea3 simulator [9] and is a real-life workload. In order to simulate the distributed scheduler used in Nordugrid ARC via centralized scheduler we use First-Come-First-Served queue processing policy.

The characteristics of this file are the following

Average CPUs count requested for a job	1.553253068
Average estimated runtime	20976.14668
Minimum CPUs count requested for a job	1
Maximum CPUs count requested for a job	60
Minimum estimated runtime	1
Maximum estimated runtime	2592130
Number of jobs	103656

The following figure represents the simulation of Random broker algorithm which is default for UNG. This algorithm shows poor task distribution among the resources which results in a very long makespan (more than 400 days).

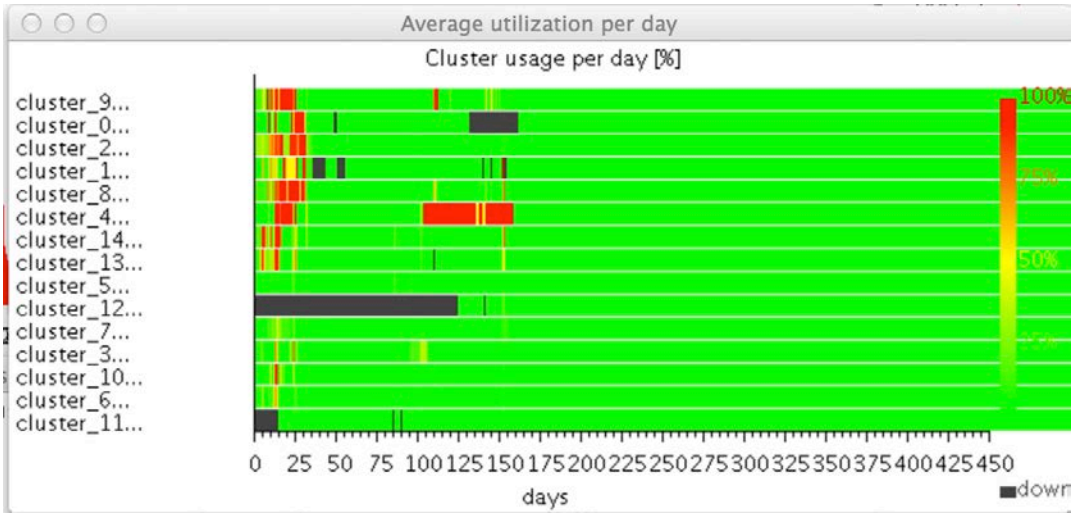

Figure 1. Resources load rate in time for Random broker algorithm

The suggested Earliest Start algorithm showed better results: about 47 days with more balanced load between the resources.

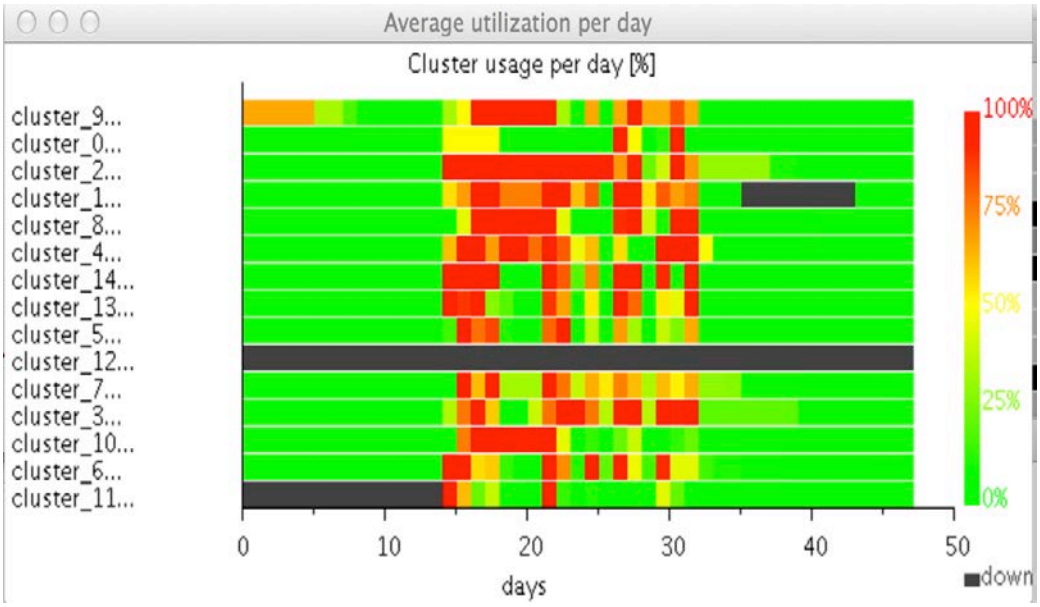

Figure 2. Resources load rate in time for Earliest Start broker algorithm

Concerning the algorithm described we could suggest an architecture for UNG that will implement this method.

Using the ARC platform service feature it is possible to implement a web-service that will store the estimations for the task types for different CPU types present in the Grid segment. If there is no such task type stored in the service database the average values is returned in the response.

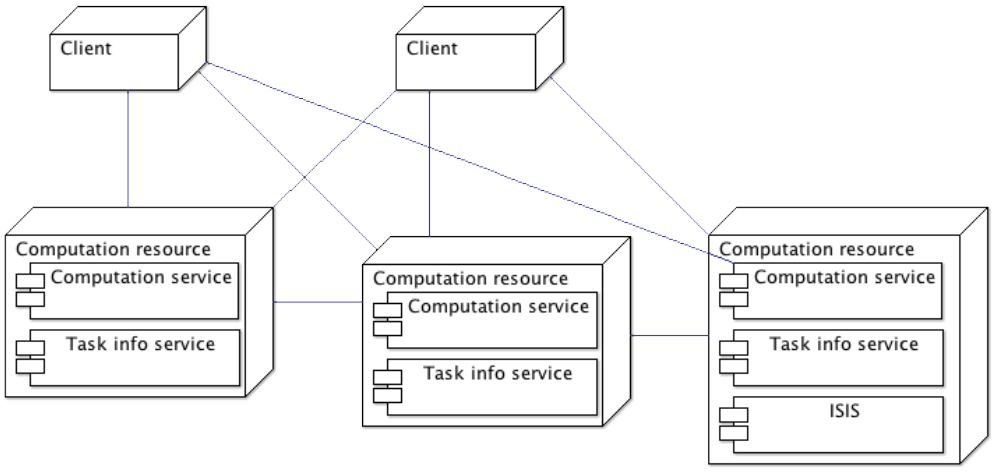

Figure 3. Suggested architecture for start time estimation feature in Nordugrid ARC environment

The following architecture can be implemented either as centralized when we have the central database of tasks or decentralized. In this case on the computational resource side there is a task information peer-to-peer service deployed. These services exchange information between themselves like it is implemented for ISIS information service [10]. Task information service stores information about task type execution length on a certain type of CPU:

$$L = f(T_t, T_{cpu})$$

where T_t – type of task, T_{cpu} - processor type on a cluster where the service is deployed. If there is no such CPU type specified the average execution time for the task type is returned. In case when no such task type specified the average time for all records is returned.

4 Conclusion

The article reviews the modern approaches to building brokers for Nordugrid ARC as well as the state of task scheduling in the Ukrainian Grid segment.

Here we represented a method on how to predict the earliest start time for a task in the Nordugrid ARC scheduler. Simulation showed that this method is significantly better that the default one. An architecture also has been suggested to implement this method using ARC platform.

In this algorithm we still do not take into account network bandwidth for a selected resource if the data amount is quite big. Our future work will be dedicated to make use of resource network status while making a decision about a resource as a candidate to execute a task being scheduled.

References

[1] A. Petrenko, S.Svistunov, P.Svirin. The Algorithm of Load Evaluation for A Grid Site. Proceedings of 13-th International Conference «SAIT-2011», p. 388.

[2] Y. Chao-Tung, C. Sung-Yi, C. Tsui-Ting. A Grid Resource Broker with Network Bandwidth-Aware Job Scheduling for Computational Grids. Advances in Grid and Pervasive Computing. – 2007 - Vol. 4459. - pp.1 - 12.

[3] Nordugrid ARC website. http://www.nordugrid.org

[4] A.Zagorodniy, G. Zinovyev, E. Martynov, S. Svistunov. Ukrainian academic Grid: Ukrainian-macedonian scientific compilation, 2009, No. 4., pp.140-150.

[5] A. Read, A. Taga, F. Ould-Saada, K. Pajchel, B. H. Samset, D. Cameron. Complete Distributed Computing Environment for a HEP Experiment: Experience with ARC-Connected Infrastructure for ATLAS. http://www.nordugrid.org/documents/chep07-atlas.pdf

[6] J. Kennedy. ATLAS Production System. http://www.etp.physik.uni-muenchen.de/dokumente/talks/jkennedy_dpg07.pdf

[7] Grid Monitor. http://gridmon.bitp.kiev.ua/

[8] A. Petrenko, S. Svistunov. P. Svirin. Hybrid broker algorithm for Nordugrid ARC 2.0. 2012, Proceedings of 2nd international conference «High-Performance Computing», p. 275.

[9] Klusacek, D., Matyska, L., Rudova, H.: Alea-Grid Scheduling Simulation Environment. Lecture Notes in Computer Science 4967 (2008) 1029

[10] ARC peer-to-peer information system. http://www.nordugrid.org/documents/infosys_technical.pdf

[11] Ioan Raicu, I. Foster et al. Towards Data Intensive Many-Task Computing. Data Intensive Distributed Computing: Challenges and Solutions for Large-Scale Information Management, IGI Global Publishers, 2009.