УДК 004.75

ADVANCED BROKERS FOR NORDUGRID ARC USING WEB-SERVICES

A. Petrenko
Professor, head of department*
E-mail: paul.svirin@gmail.com
S. Svistunov
Senior researcher, Ph.D.
Bogolyubov Institute for Theoretical Physics
Metrologichna str, 14-b, , Kyiv, Ukraine, 03680
E-mail: svistunov@bitp.kiev.ua
P. Svirin
Assistant professor*
E-mail: paul.svirin@gmail.com
*System Design department
National Technical University of Ukraine
“Kiev Polytechnical Institute”
av. Peremohy, 37, Kyiv, Ukraine, 03056

Мета даної роботи полягає в розробці стратегій і архітектур для розподіленого брокера в середовищі Nordugrid ARC 2.0 із застосуванням сучасних можливостей даної платформи. Такі стратегії мають бути загального призначення, отже, не мають орієнтуватись на специфічні задачі в Грід-сегменті, де застосовуються дані стратегії

Ключові слова: Грід, Nordugrid ARC, веб-сервіси, брокер, балансуваннянавантаження, відмовостійка багатопроцесорна система

Цель данной работы состоит в разработке стратегий и архитектур для распределенного брокера в среде Nordugrid ARC 2.0 с применением современных возможностей платформы. Такие стратегии должны быть общего назначения, то есть, не ориентироваться на специфические задачи в Грид-сегменте, в котором применяются данные стратегии

Ключевые слова: Грид, Nordugrid ARC, веб-сервисы, брокер, балансировка нагрузки

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 solutions [1 – 2] 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.

UkrainianNational 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 full 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 own 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 (Fig. 1).

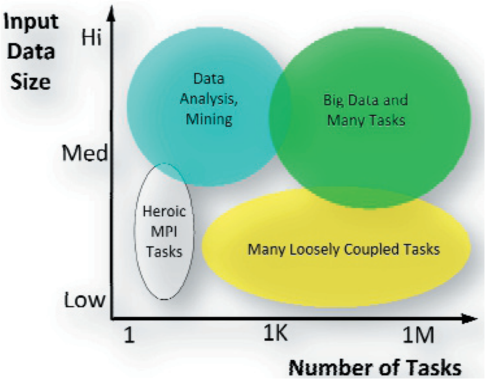

Fig. 1. Many tasks computing

The main tasks that require Grid are the following:

- large number of tasks with low requirements regarding the resources. Such tasks are executed over a short period of time;
- large number of tasks with high requirements regarding the resources that are executed over a long period of time.

- Examples of such tasks are as follows:
- ALICE experiment data processing. Usually such task requires 1 processor; the data are transferred to the resource in the course of calculation, maximum running time is up to 24 hours. Number of such tasks can range up to hundreds of thousands;
- Calculation of molecular dynamics tasks. This category of tasks requires a big number of processors, transfers a small amount of data for calculation, maximum running time of such tasks is up to months. Number of such tasks can range up to thousands.

Hence, the use of a single strategy for distribution of various categories tasks is not efficient. The solutions to this problem are specialized Grid systems such as AliEN Grid, WeNMR. However, the number of tasks categories is extremely important and it is not possible to develop a system for each category of task.

The specifics of Ukrainian Grid infrastructure are the following:

- 38 clusters with low computational performance [4];
- 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 [5 – 8] 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 especially 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. Optimization strategies for broker

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 simulate the strategies suggested we used Alea3 [9 – 11] which is a Grid and cluster scheduling simulator designed for study, testing and evaluation of various job scheduling techniques. This event-based simulator is able to deal with common problems related to the job scheduling as well as the heterogeneity of jobs, resources, and the dynamic runtime changes such as the arrivals of new jobs or the resource failures and restarts. Alea3 is based on the popular GridSim toolkit [12 – 13] and represents the next generation of Alea2 which is a major extension of the Alea simulator, developed in 2007. The main part of the simulator is a complex scheduler which incorporates several common scheduling algorithms working either on the queue or the schedule (plan) based principle. Same as the GridSim, the Alea3 is an event-based modular simulator, composed of independent entities which implements the desired simulation functionality. It consists of the centralized scheduler, the grid resources with the local job allocation policy, the job loader, the machine and failure loader and additional classes responsible for the simulation setup, the visualization and the generation of simulation output.

For the simulation we used a standard file metacentrum.mwf which comes together with Alea3 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

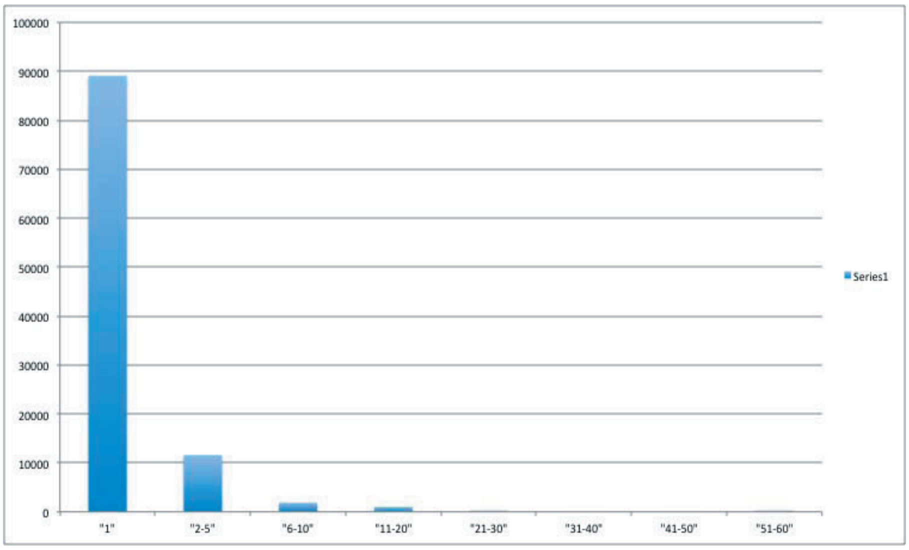

Fig. 2. Task distribution by CPUs count requested

On this figure (Fig. 2) we can see the distribution of jobs with respect to the CPUs count requested. Most of the jobs request only one CPU to run.

On this figure (Fig. 3) the jobs distribution respectively to their estimated length. Most of the jobs are short ones with estimated lengths up to 3600 seconds.

We can notice that the resource load is balanced but still the most powerful clusters (e.g. cluster_11) are running with low load (Fig. 4).

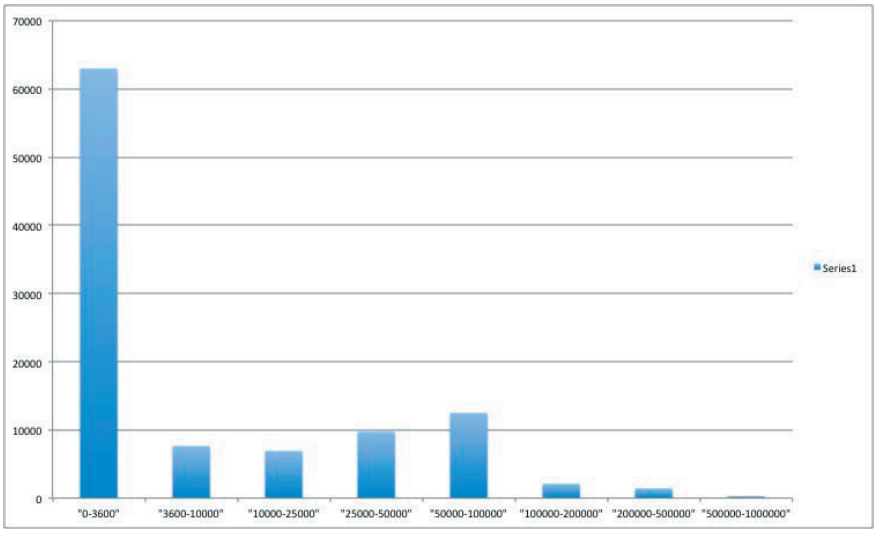

Fig. 3. Task distribution by CPU time requested

Thus, it is possible to combine this approach for load balancing with task rescheduling using receiver initiated strategy when the request for task relocation is issued by a free resource. The idea of the approach is to request the resources with queued tasks for the opportunity to transfer one to a resource that currently is in idle state. In this experiment the last task in the queue was transferred to a free resource (Fig. 5).

The simulation showed that using this approach most of the tasks go to the most powerful clusters and there are many low-loaded clusters in the segment.

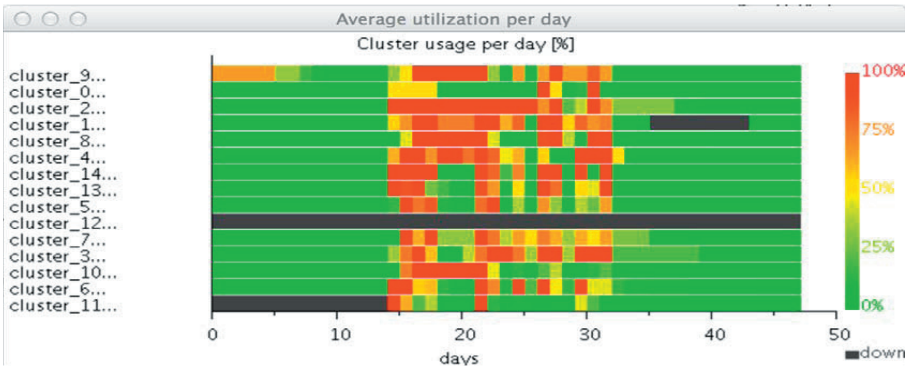

Fig. 4. Resource load for Earliest start approach

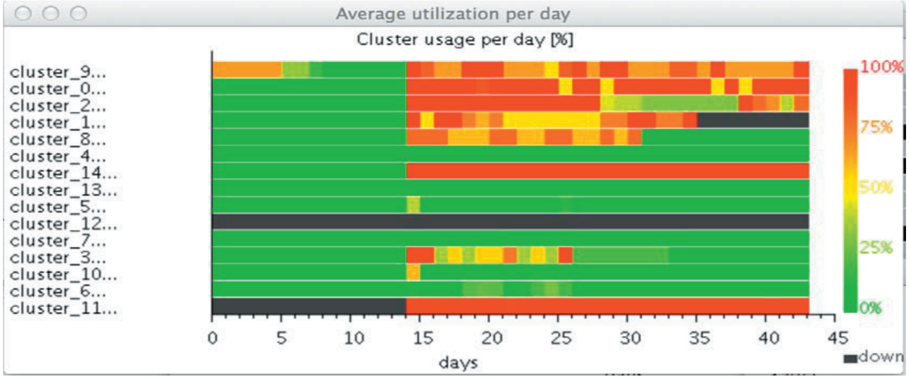

Fig. 5. Resource load for Earliest start approach with rescheduling

However, we can notice decreased makespan when using this approach.

4. Suggested architecture for ARC in UNG

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 service that will store the estimations for the task types for different CPU types present in the Grid segment. If there is no such type stored in the service database the average values is returned in the response. Fig. 6 shows the architecture for Nordugrid ARC with the feature described.

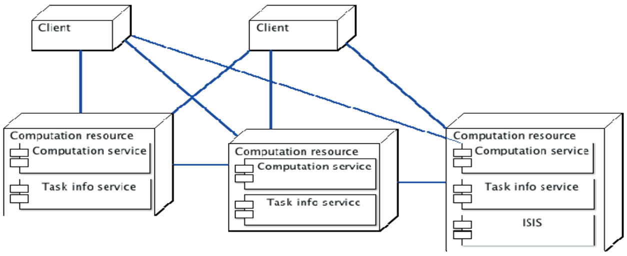

Fig. 6. Suggested architecture for start time estimation feature in Nordugrid ARC environment

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 [12].

In case of using the rescheduling feature it is necessary to implement an intermediate service which will act as a mediator between client ARC software and AREX service which executes the tasks. The implementation of this service can be centralized or P2P like ISIS information service. Client software posts task to a candidate resource and also registers the task URL in this service. In case of relocation of the task the new URL is stored into the service database. When the user queries for task state or tries to retrieve the execution results the client software first queries the service which returns the last URL for the task and then queries the real resource for the data.

5.Conclusions

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 and implement this approach for Nordugrid ARC scheduler. Simulation showed that this method is significantly better that the default one. It had been noticed that many tasks still await in the queue until the resource starts their execution while other resources have finished their tasks and became free. In this situation it is useful to run task rescheduling which can decrease the makespan for a package of tasks.

An architecture also has been suggested to implement this method using ARC platform. It is also possible to extend this architecture with other features.

References

1. Петренко, А. Алгоритм оцінки завантаженості ГРІД-сайту [Текст] / А. Петренко, С. Свістунов, П. Свірін // Матеріали конференції«Системний аналіз та інформаційні технології», 23-28 мая 2011. - г. Киев, Украина - 388с.
2. Chao-Tung, Y. A Grid Resource Broker with Network Bandwidth-Aware Job Scheduling for Computational Grids. [Текст] / Y. Chao-Tung, C. Sung-Yi, C. Tsui-Ting // Advances in Grid and Pervasive Computing – 2007. - Vol. 4459 - pp. 1 - 12.
3. Nordugrid ARC. [Електронний ресурс]. - Режим доступу: http://www.nordugrid.org.
4. Загородний, А. Украинский академический грид [Текст] / А. Загородний, Г. Зиновьев, Е. Мартынов, С. Свистунов // Українсько-македонський науковий збірник, Вип. 4 - Київ: Вид-во Національна бібліотека України імені В.І.Вернадського, 2009. - С.140-150.
5. Read, A. Complete Distributed Computing Environment for a HEP Experiment: Experience with ARC-Connected Infrastructure for ATLAS. [Електронний ресурс] / A. Read, A. Taga, F. Ould-Saada, K. Pajchel, B. H. Samset, D. Cameron. - Режим доступу: http://www.nordugrid.org/documents/chep07-atlas.pdf.
6. Kennedy, J. ATLAS Production System. [Електронний ресурс]. - Режим доступу: http://www.etp.physik.uni-muenchen.de/dokumente/talks/jkennedy_dpg07.pdf
7. Werner, J. Grid computing in High Energy Physics using LCG: the BaBar experience. [Електронний ресурс]. - Режим доступу: http://www.gridpp.ac.uk/papers/ahm06_werner.pdf.
8. Boyanov, L. On the employment of LCG GRID middleware. [Електронний ресурс]. / L. Boyanov, P. Nenkova. - Режим доступу: http://ecet.ecs.ru.acad.bg/cst05/Docs/cp/SII/II.11.pdf.
9. Петренко, А. Гібридний алгоритм брокера для Nordugrid ARC 2.0. [Текст] / А. Петренко, С. Свістунов, П. Свірін // Матеріали конференції HPC UA, 8-12 жовтня, 2012, Київ, Україна - с. 275.
10. Raicu, I. Towards Data IntensiveMany-Task Computing. Data Intensive Distributed Computing: Challenges and Solutions for Large-Scale Information Management. [Текст] / I. Raicu, I. Foster et al. -IGI Global Publishers, 2009 - 352 pages.
11. Klusacek, D. Alea - Grid Scheduling Simulation Environment. [Текст] / D. Klusacek, L. Matyska, H. Rudova. // Lecture Notes in Computer Science – 2008. - pp. 1029 – 1038.
12. ARC peer-to-peer information system. [Електронний ресурс]. - Режим доступу: http://www.nordugrid.org/documents/infosys_technical.pdf.
13. Buyya, R. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. [Текст] / R. Buyya, M. Murshed. // Concurrency and computation: practice and experience – 2002. - Vol. 14, No.13 - pp.1175—1220.