Adaptive Grid Resource Selection Based on Job History Analysis Using Plackett-Burman Designs

Cinyoung Hur and Yoonhee Kim*

Department of Computer Science,
Sookmyung Women's University, Korea
{hurcy,yulan}@sm.ac.kr

Abstract. As large-scale computational applications in various scientific domains have been utilized over many integrated sets of grid computing resources, the difficulty of their execution management and control has increased. It is beneficial to refer job history from many application executions, in order to identify application's characteristics and to decide grid resource selection policies meaningfully. In this paper, we apply a statistical technique, Plackett-Burman design with fold-over, for analyzing grid environments and execution history of applications. It identifies main factors in grid environments and applications, ranks based on how much they affect. Especially, the effective factors could be used for future resource selection. Through this process, application is performed on the selected resource and the result is added to job history. We analyzed job history from an aerospace research grid system. The effective key factors were identified and applied to resource selection policy.

Keywords: Resource Selection, Job History, Grid Computing.

1 Introduction

A grid computing[4] is used effectively to support applications of various fields by aggregating computing, network and storage resources. Especially, research capabilities of scientific domains have been increased with grid technology. Although many research grids have been developed, mostly these are limitedly configured for particular domains. To increase the utilization of grid technology by sharing it with various domains, it requires deep understanding of application's characteristics, as well as large-scale computing resources for the applications [7]. Typically, scientific applications take a lot of time to solve problems with enormous size of data, required grid environments based on characteristics of applications in a specific domain. For example, as a CFD(Computational Fluid Dynamics) [9, 13] application usually requires numerical analysis and its visualization of the process, high performance computational grids are needed. If resources are configured without considering these characteristics, grid utilization could be lowered and the efficiency of research be degraded as well.

[*] Corresponding author.

© Springer-Verlag Berlin Heidelberg 2009

It is important to understand and utilize thoroughly about job history [5] of applications executions on grid, as well as characteristics of the applications. Analysis of job history from various angles can help in diverse areas such as research in grid resource management, grid maintenance and operation and grid design. Also research applying various statistical methods on job history analysis is actively conducting. Key features of the methods help to define characteristics of applications and properties of grid as factors and to identify effects of each factor and interaction between factors. Execution of an application considered these factors can use grid efficiently. However, if it is too many, that makes difficult to identify effects and interaction of factors - Statistical approach is popularly used to find out these factors and their interaction from job history [10, 11]. Specifically, we use a Plackett-Burman design [3, 12] which is easy to quantify effects of factors which are significant in grid environments and the execution of applications. In addition, the quantified factors and its effects is useful for resource selection.

To apply job history for resource selection polices, each job execution is identified as a job profile, which includes requirement of an application and resource, the result of the execution, and other features understanding characteristics of applications and status of grid. We apply Plackett-Burman designs for screening the factors' impacts on execution of an application as conditions in resource selection policies. Resource selection policies use the effective factors for choosing resources, reflect gap between actual execution time and referred execution time in job history. Newly executed result is added to job history as a new job profile, and used for next resource selection.

Concludingly, we allocate resource efficiently according to characteristics of applications and the status of grid environment. As job profiles are accumulated to job history, we also understand applications deeply more and guarantee resource selection with better resource stability [1, 2].

The rest of this paper is structured as follows. In Section 2, we explain background and assumption. In Section 3, we present our resource selection algorithm. In Section 4, we describe its design and implementation. In Section 5, we evaluate our approach in many grid scenarios. In Section 6, we address the related work. Finally, in Section 7, we conclude and describe future work.

2 Background and Assumption

2.1 Characteristics of Target Application

Simulation of CFD(Computational Fluid Dynamics)[9], scientific discipline which solves a flow variation around target geometry using numerical methods, consists of three main steps. A mesh targeted to solve is constructed, then numerical analysis is applied on the mesh, and finally, resultant data are analyzed using a visualization software. In CFD simulation, the step of solving a mesh requires huge computing resources because it iteratively computes fluid dynamic equations on a mesh and processes resultant data for visualization. Before starting to solve a mesh, multiple simulations are dynamically created with various conditions and diverse ranges [8]. Since execution time of these simulations depends on the conditions and ranges, we define these as an application's characteristics and the numerical analysis on a mesh as a CFD application in this paper.

In general, a CFD application first transmits input files of simulations such as a mesh, information of parameters for analysis, and solvers of a mesh, to computing resources. Actual systems for CFD application is e-AIRS(e-Science Aerospace Integrated Research System) 2.0 [14], which is an aerospace integrated research system, currently running over Korea VOs(Virtual Organizations) and PRAGMA VOs. Korea VOs include clusters of KISTI(Korea Institution of Science and Technology Information) [15]. PRAGMA(Pacific Rim Application and Grid Middleware Assembly) [16] VOs provides large-scale grids composing of 35 institutional resources.

2.2 Effect of Factors and Statistical Methodology

P. Shivam [11] defines rigorously the notion of factors and their effect on system behavior. Based on his study, we define main causes in the performance of grid applications as factors and then analyze how much each factor impacts on the performance. For example, characteristics of computational applications, CPU speed and memory size could be key factors deciding performance of grid applications (See Table 1). Equation (1) represents factors and the most effective factor on an application is represented as shown in Equation (2).

$$F = \{f_1, f_2, \dots, f_k\} \tag{1}$$

$$f_{max} = \{ f \mid \text{most effective } f \in F \} \tag{2}$$

Before starting to solve a mesh, multiple simulations are dynamically created with various conditions and ranges of parameters for solving. Since execution time of these simulations depends on the conditions and ranges on it, we define these as an application's characteristics and we name this numerical analysis on mesh as CFD analysis in the following

Table 1

Domains	Factors	
Application	Mach numbers (MA)	Local flow-speed divided by local speed of sound
	Reynolds numbers (RE)	Relative importance of inertial and viscous forces in a flow
	Angle of Attack (AOA)	Angle between a reference line and the oncoming flow
	Total iteration (ITMAX)	Number of iteration in application
Network	Bandwidth, Propagation delay, Location of Clusters	
Resource	CPU capability, Memory, Stability	

2.3 Plackett-Burman Designs [3, 12]

Consider a situation of a design with N factors, each of which having one of M values, it is possible to simulate a exhustive experiment consisting of M^N runs for a complete analysis. In case of each factor have only two values, simulating all the possible combinations would require 2^N simulations. Since this number of simulations is extremely high, designs of logically minimal number of simulations is required to estimate quickly the effect of each of the N fators.

Plackett-Burman(PB) designs, a well-established approach of this type, provide a method to logically minimize the number of simulations in cases where interactions of factors are not presented. The PB design measures the effects of k=N-1 factors in N runs, where N is multiple of 4. A design matrix is used for PB designs and each rows in the design matrix corresponds to different experiment configurations. For N = 8, the first rows of the matrixfor PB design, which is used in this paper, is acquired from [3]. The second and further rows are found by rotating the previous row to the right; the final (X-th) row is a series of low values.

An improvement on the PB design is PB designs with fold-over(PBDF). Using the PBDF, we can determine the effect of all of main parameters and selected interactions with only 2N experiments. In the case of PBDF, a further N rows having switched values is included, therefore total number of required experiments equals to 2N.

The columns of the matrix correspond to factors, and the rows correspond to different simulation configurations. There are a series of (+) and (-) values corresponds the high and low values of a factor which should be selected to be just outside the range of normal values. After running 2N simulations, the effect of each factor is computed by multiplying the result for each configuration by the value of the entry for that congifuration and factor combination. Then the results of all those multiplications are summed together to determine the overall effect for that factor. When comparing effects of different factors, only the magnitude is used.

3 Resource Selection Optimization Algorithm Based on Job History

3.1 Job Profiles and Job History

Job profiles contain application's characteristics, required conditions for executing job and information of resources. After the execution of an application, which can be defined as interactions among elements in profiles creates several results such as job execution time, completion ratio, the results are also included in a job profile that is used as a record unit of a executed job. Equation (3) represents a job profile and its information in it. A job profile is defined as j, one of a requested job j is allocated on particular resource(r) and executed during T_j.

$$P = \langle j, r, T_j \rangle \text{ where } j \in J = \{j_1, j_2, \ldots, j_n\},\ r \in R = \{r_1, r_2, \ldots, r_m\},\ T_j > 0 \tag{3}$$

Job history is a set of separate job profiles accumulated for a long time. Job history gets richer as jobs are executed more. From the job profiles, we can deduce common information, execution tendency, and patterns of target execution environment. Not all job profiles are applied for job history. Screening a job profile which is not acceptable due to exceptional execution is carried out with caution. Significant job profiles in job history are chosen for reference as PR(Profile Reference) set. A Plackett-Burman design is applied in the screening procedure.

$$P \in \text{JobHistory} \quad (4)$$

$$PR = \{P \mid r \in R_{fast}, PR \subseteq \text{JobHistory}\} \quad (5)$$

3.2 Resource Selection Optimization Algorithms

Objective of below algorithm is to get most effective factor and reference job profiles and optimal resource for job (See Figure 1). If effective factors does not exists, it assumes that consider PBDF(Plackett-Burman with Fold-Over) is never conducted and then estimates impact of factors on system with Plackett-Burman designs (2~6 lines). Rating the impact in decreasing order and f_{max} is defined the most effective factor related to resources. Using JobHistory(F,j) function, set of reference job profiles which contains information of candidate resources is obtained from job history, giving the first consideration to effective factors of applications (7, 11~16 lines). In each reference job profile, resources elements where job j was executed are gathered in set R(8 line). Estimation of resources based on f_{max} is required (9 line). For example, if f_{max} is stability, the stablest resource is selected. Finally, job is submitted to selected resource (10 line).

```
Given $j \in J, T_j$
 1) $f_{max}$ = null, PR = $\emptyset$, $r_{selected}$ = null
 2) if $f_{max}$ not exists then
 3)     PBFD(F, $T_j$)
 4)     Rating F in deceasing order
 5)     $f_{max}$ = one of F related to resource
 6) endif
 7) PR = JobHistory(F,j)
 8) R = PR(r)
 9) $r_{selected}$ = r of $f_{max}$(R)
10) Dispatch j on $r_{selected}$

JobHistory(F,j)
11) PR = null
12) repeat each Profile
13)     Find P where $C_j == f_{max}$
14)     PR = PR $\cup$ P
15) until P == empty
16) return PR
```

Fig. 1. Algorit*h*m for Referring Job Profiles

```
After finish the j on $r_{selected}$
 1)  $P_{new} = \langle j, r_{selected}, T_j \rangle$
 2)  JobHistory = $P_{new} \cup$ JobHistory
 3)  if $P_{new}$ == Failed then
 4)      decrease Stability of $r_{selected}$
 5)  else
 6)      Err = $|T(P_{new}) - T(P_{selected})|$
 7)      if Err < *Threshold* Then
 8)          Increase Credit $f_{max}$
 9)      else↓
10)          Decrease Credit $f_{max}$
11)          Repeat PBFD(F, $T_j$)
12)      endif
13)  endif
```

Fig. 2. Adaptive Algorithm

Job executed on the selected resource could be succeeded or failed. Above algorithm provides adaptive approach with considering such situations (See Figure 2). New job profile produced after job execution is added to job history (2 lines). In case of job failed, stability of resource where the job is executed on is adjusted (3~4 lines). On the other hand, new job profile is compared with reference profile (5~11 lines). As gap between time elements of two profiles is smaller, the algorithm for referring job profiles is reliable (6 line). Credit of factor is increased when gap is smaller than threshold (7~8 lines), otherwise credit is decreased and PBDF is repeated (10~11 lines). The threshold depends on the user's requirement such as maximum tolerance of factors.

4 Design and Implementation of Job History-Based Resource Selection Optimization System

In this system, users are allowed to apply various policies for their resource selection by identifying characteristics of applications and properties of resources. Heterogeneous distributed resources are constructed as VO(Virtual Organization) through Virtualization. As shown in figure 3, Virtualization provides resource sharing service, resource and workload monitoring service, dynamic resource allocation service. Based on these services, System Middleware supports main features of resource selection with job history.

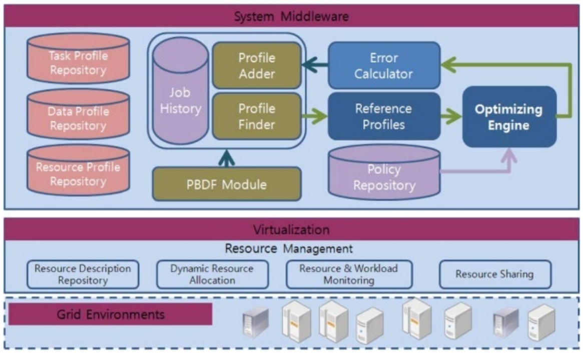

Fig. 3. Architecture of Resource Selection System based on Analysis of Job History

In System Middleware, job profiles are aggregated according to profile's schema which defines characteristics of applications and properties of resources. Aggregated profiles are managed in Profile Repositories (Task ·Data·Resource Profile Repository) and are utilized to analyze information of a particular application and resource with assistance of Profile Finder module. PBDF module, providing Plackett-Burman designs, helps to choose many reference job profiles from Job History component. Resource selection policies in Policy Repository decide resource by Optimizing Engine. Newly generated job profile is added to Job History by Profile Adder, and Error Calculation module periodically conducts estimation for maintaining accuracy of reference job profiles.

5 Experiments and Evaluation

5.1 Analysis of Factors

Prior to evaluation, main factors and job history defined above are as follows. In Plackett-Burman designs, the high and low values of factor should be selected to be just outside the range of normal values. We analyzed high and low values of each factor from job history. Values of MA, AOA are determined by quartile 1 and 3 of its distribution. On the other hand, values of RE and ITMAX is defined by mean.

Among the factors in chapter 2, we select factors of CPU clocks, stability, and network for evaluation and categorize clusters in PRAGMA grid according to selected factors. The information of clusters is referred to PRAGMA grid monitoring service [17]. Values of Network factor are determined with bandwidth and location of cluster. Stability represents ratio of succeed job over whole executed job. Clusters categorized in boundary have values of factor near dividing line, so we utilize both high and low category.

5.2 Preliminary Experiment and Results

In this section, our experiments show that our approach looks for reliable resources for applications using statistical methods and previous job execution history. PBDF

Table 2. Simulation Result of PBDF based on Actual Job History

	Parameters							Execution Time (s)
	CPU Clocks	Stability	Network	RE	AOA	MA	ITMAX	
1	+	+	+	-	+	-	-	1507
2	-	+	+	+	-	+	-	2401
3	-	-	+	+	+	-	+	47889.5
4	+	-	-	+	+	+	-	898.33
5	-	+	-	-	+	+	+	279.5455
6	+	-	+	-	-	+	+	128
7	+	+	-	+	-	-	+	6786
8	-	-	-	-	-	-	-	195.52
9	-	-	-	+	-	+	+	14373
10	+	-	-	-	+	-	+	25267
11	+	+	-	-	-	+	-	1266
12	-	+	+	-	-	-	+	7504
13	+	-	+	+	-	-	-	625
14	-	+	-	+	+	-	-	26517
15	-	-	+	-	+	+	-	363.8
16	+	+	+	+	+	+	+	1811.083
Effect	-61235	-41668.5	-13353	64790.05	71254.74	-94770.3	70264.48	
Rank	1	2	3	4	2	1	3	

design matrix in Table 2 was comprehensively generated from analysis of real job history from the e-Science environment. It indicates that "CPU clocks" is the most effective factor among factors related to resource. Also, we can find out "MA" mainly decides execution time of application. These are used in the experiments.

The first scenario was a situation of submitting an application with specific parameters and completing successfully without any problems. This scenario allowed us to confirm the capability of our algorithms. The other scenario demonstrated cases of adjusting resource selection when a failure occurred. For each scenario, we compared the execution time of an application and computed the gap between actual execution time and referred one.

Following is an example for better understanding for the first scenario. Consider a job with parameters of specific values, e.g. RE:5, AOA:7, MA:0.7 and ITMAX:10000. To select a resource for the job, we first gathered reference job profiles from job history with effective factors identified by PBDF. As a result, Table 3 was obtained through this process and we obtained information of candidate resources of {rocks-52, rocks-153, sakura, nucleus}. When we consider the minimum execution time, sakura met this requirement. However, sakura didn't provide acceptable reliability than other resources would have. Consequently, nucleus is selected as the best candidate resource for satisfying execution time and reliability requirements.

Table 3. An Example of Reference Job Profiles

JobID	RE	AOA	MA	ITMAX	Status	Resource	Execution Time
1000000233	5	7	0.7	10000	DONE	rocks-153	1701
1000000236	5	7	0.7	10000	DONE	rocks-52	2364
1000000238	5	7	0.7	10000	DONE	rocks-52	2289
1000000239	5	7	0.7	10000	DONE	sakura	1458
1000000240	5	7	0.7	10000	DONE	sakura	1491
1000000241	5	7	0.7	10000	DONE	nucleus	1321
1000000245	5	7	0.7	10000	DONE	sakura	851
1000000246	5	7	0.7	10000	DONE	sakura	900

In second scenario, we considered the situation after execution of a job. If a job were completed successfully, information of execution including execution time on selected resource would be added to job history. However, if a job were failed, we would regard the selected resource as unavailable and decrease its stability. For example, nucleus, the selected resource of above example, was crashed and led the job to be failed. Then stability of nucleus was adjusted to lower value than 100.

6 Related Works

NIMO [10] system that automatically learns cost models for predicting the execution time of computational scientific applications executing on large-scale computational grids. Accurate cost models helps to select candidate resources of good performance. NIMO generates training samples of biomedical application to learn fairly-accurate cost models quickly using statistical learning techniques. NIMO's approach actively deploys and monitors the application under varying conditions with no changes to the operating system or application. It obtains training data of applications from passive instrumentation streams.

GWA(The Grid Workloads Archive) [5, 18] is a study of gathering and utilizing job history. It presents a place for exchanging workload data and meeting point for grid users. Also, GWA defines requirements of building a workload archive and describes the approach taken to meet these requirements with the GWA. For example, a format for sharing grid workload information, and tools associated with this format are suggested as these requirements. Using these tools, GWA collects and analyzes data from nine well-known grid environments. Finally, several cases of utilizing GWA are presented in critical grid research and practical areas such as research in grid resource management, grid design, operation, and maintenance.

The theory of *design of experiments*(DOE) [3] is a rigorous theoretical field of statistics that studied planned experiments of factors affecting system behavior. Recently, grid research has applied DOE at improving performance of meta-scheduler [6] on grid environments.

7 Conclusion

In this paper, we investigated the problem of resource selection for scientific grid applications. Also, we presented that resource for a job is selected based on analysis of job history that encapsulates knowledge of an application and grid environments. Under these objectives, we apply Plackett-Burman designs with Fold-Over for identifying characteristics of CFD applications in an aerospace research grid and properties of large-scale networked grids. Since these characteristics are considered as factors that impact on CFD applications execution time, resources are chosen based on these factors. Our approach also allows adaptive resource selection by estimating accuracy of job history after every job execution. Also, since job history gets rich as jobs are executed, reliability of resource selection is improved.

Future work will involve extending our approach to utilize grid with various applications. Algorithms in this paper will be refined by identifying realistic weight of effective factors and making a comparative study of real job history. Also, further research is needed to guarantee efficient execution of various applications by supporting a service for easily defining characteristics of each application. Also, we will apply our strategy on various requirements of users to expand the base of study for improvement of resource utilization and management. These objectives will be evaluated with more precise experiments in the future.

References

1. Hur, C., Kim, Y., Kim, J., Cho, K.W.: A Prediction Strategy of Resource Selection with Pattern of Job History in e-AIRS 2.0 System. In: KIISE HPC Winter Conference (February 2009)
2. Hur, C., Kim, Y.: Prediction Scheduling with Patterns of Job History in e-AIRS System. In: KIISE Fall Conference, October 2008, vol. 35(B), pp. 520–525 (2008)

3. Montgomery Douglas, C.: Design and Analysis of Experiments, 5th edn. John Wiley & Sons, Inc., Chichester (2000)
4. Krauter, K., Buyya, R., Maheswaran, M.: A taxonomy and survey of grid resource management systems for distributed computing. Software—Practice & Experience 32(2), 135–164 (2002)
5. Losup, A., Li, H., Jan, M., Anoep, S., Dumitrescu, C., Wolters, L., Epema, D.H.J.: The Grid Workloads Archive. Future Generation Computer Systems 24(7), 672–686 (2008)
6. Vanderster, D.C., Dimopoulos, N.J., Sobie, R.J.: Improved Grid Metascheduler Design using the Plackett-Burman Methodology. In: Proceedings of the 21st International Symposium on High Performance Computing Systems and Applications (May 2007)
7. Wrzesinska, G., Maassen, J., Bal, H.E.: Self-adaptive applications on the grid. In: Proceedings of the 12th ACM SIGPLAN symposium on Principles and practice of parallel programming, San Jose, California, USA (March 2007)
8. Casanova, H., Obertelli, G., Berman, F., Wolski, R.: The AppLes Parameter Sweep Template: User-level middleware for the Grid. In: IEEE Supercomputing 2000, Dallas, Texas, USA, November 4-10 (2000)
9. Kim, J., Kim, Y., Kim, B.S., Ahn, J.W., Kim, J.-H., Lee, J.-S., Choi, J.G., Lee, G.-B., Cho, J.H., Kim, J., Hur, C., Kang, S.-H., Rye, G.-Y.: Development of e-Science technology for fluid dynamic research in various fields report I-08-GG-05-01R-1. Korea Institute of Science and Technology Information (2008)
10. Shivam, P., Babu, S., Chase, J.: Active and Accelerated Learning of Cost Models for Optimizing Scientific Applications. In: International Conference on Very Large Data Bases (VLDB), Seoul, Korea (September 2006)
11. Shivam, P.: Proactive Experiment-Driven Learning for System Management. Department of Computer Science. Duke University (2007)
12. Park, S.H.: Design of Experiments, Minyoungsa (2005)
13. Computational Fluid Dynamics, `http://www.cfd-online.com/`
14. e-AIRS 2.0,
`http://repository.kisti.re.kr:8080/gridsphere/gridsphere`
15. Korea Institution of Science and Technology Information,
`http://www.kisti.re.kr/`
16. Pacific Rim Application and Grid Middleware Assembly,
`http://www.pragma-gird.net`
17. PRAGMA Grid Monitoring home page,
`http://pragma-goc.rocksclusters.org/scmsweb/`
18. The Grid Workloads Archive, `http://gwa.ewi.tudelft.nl`