Predicting Job Start Times on Clusters

Hui Li*† David Groep† Jeff Templon† Lex Wolters*

* Leiden Institute of Advanced Computer Science, Leiden University
The Netherlands
{hli,llexx}@liacs.nl

† National Institute for Nuclear and High Energy Physics (NIKHEF)
The Netherlands
{davidg,templon}@nikhef.nl

Abstract

In a Computational Grid which consists of many computer clusters, job start time predictions are useful to guide resource selections and balance the workload distribution. However, the basic Grid middleware available today either has no means of expressing the time that a site will take before starting a job or uses a simple linear scale. In this paper, we introduce a system for predicting job start times on clusters. Our predictions are based on statistical analysis of historical job traces and simulation of site schedulers. We have deployed the system on the EDG (European DataGrid) production cluster at NIKHEF. The experimental results show that acceptable prediction accuracy is achieved to reflect real site states and site-specific scheduling policies. We find that the average error of our job start time predictions is 18.9 percent of the average job queue wait time and this is around 20 times smaller than the average prediction error using the EDG solution.

1. Introduction

Job start time predictions are useful for resource selections in the Grid, provided that acceptable accuracy is achieved to reflect real site states. However, the basic Grid middleware available today either has no means of expressing the time that a site will take before starting a job or uses a simple linear scale. In the European DataGrid [1], for instance, every computing resource (corresponding to a batch queue) publishes one single job start time, which is based on the user specified wall clock times and the number of queued jobs. This seems to be a reasonable approach, except that experience shows that it cannot deliver acceptable accuracy for the *resource broker* [2] to make proper decisions. Several limitations are found in this approach. Firstly, the user specified wall clock times are generally much larger than the actual job run times, which results in large prediction errors. Secondly, it assumes that FCFS (First Come First Serve) scheduling is used at all sites. This is not valid since sites have different scheduling systems and they are generally more sophisticated than FCFS. Thirdly, every site has its own set of scheduling policies and jobs from different Virtual Organizations (VOs) would most likely have different job start times. Therefore publishing single job start time estimates in the Grid Information Service [3] is not sufficient and we need more sophisticated and detailed job start time predictors.

In this paper, we present a job start time prediction system for clusters. Our system is based on statistical predictions of job run times and simulations of schedulers. We obtain job start time predictions via a chain of steps: 1) historical job information is used to predict execution times of jobs currently running and queued at the site; 2) a scheduler simulation is performed, along with the predicted job run times, to determine how long it will take before a newly-submitted job will start execution; 3) predicted job start times are published to the Grid Information Service in accordance to the scheduling policies defined at the site. We have deployed the system on the NIKHEF EDG production cluster. We find that the average error of our job start time predictions is 18.9 percent of the average queue wait time and it significantly improves the originally implemented EDG solution. We also evaluated our job run time prediction technique on clusters subject to a more diverse workload. The average prediction errors range from 13 to 35 percent of the average job run times.

The rest of the paper is organized as follows: Section 2 describes our technique to predict job run times and it is evaluated using workload traces recorded on three selected clusters. Section 3 describes how we simulate the site scheduling system and our technique to improve the simulation performance. Section 4 describes the system design, the idea of incorporating site scheduling policies and experimental results when deploying the system on the NIKHEF EDG cluster. In Section 5 conclusions are presented and future work is discussed.

Cluster	Location	OS	LRMS	CPUs	Period	Job entries
EDG production	NIKHEF	Linux	PBS	20	01/2003 - 04/2003	11537
DAS-2	VU	Linux	PBS	144	01/2003 - 04/2003	40096
DAS-2	UvA	Linux	PBS	64	01/2003 - 04/2003	5857

Table 1. Characteristics of clusters and job traces (LRMS - Local Resource Management System).

2. Predicting job run times

The first step of obtaining job start time predictions is to predict job run times. This part of work is based on statistical techniques [4, 5, 6], in which predictions are generated by applying statistical methods on historical job traces.

2.1. Related work

In [4, 6], jobs in historical traces are categorized according to their attributes (user name, executable name, etc). The *templates*, which are defined as a set of job attributes, generate categories to which jobs can be assigned. Jobs that fall into the same category are considered similar and statistical methods such as *mean* or *linear regression* are applied to generate run time predictions. Various approaches differ in the set of job attributes used and their template definitions. A comparison of these techniques is available in [6].

Compared with previous approaches, we go one step further towards prediction generation. In [6], among all estimates produced by the set of chosen templates and estimators, the one with the smallest confidence interval is selected as the prediction. In our approach, we evaluate different techniques to select estimates. These techniques include choosing an estimate based on previous prediction errors, or combining the estimates to produce new predictions. Finally, the technique with the smallest average prediction error is selected to implement the job run time predictor.

Our experiments are mainly based on the NIKHEF EDG [7] production cluster. For comparative studies, we also use traces on DAS-2 [8] clusters at UvA (Universiteit van Amsterdam) and VU (Vrije Universiteit Amsterdam). Characteristics of these clusters and workload traces are given in Table 1.

2.2. Template definition and evaluation

The first step of our approach is to define a suitable set of templates and evaluate them quantitatively using historical traces. For our traces, we find that *group name* (G), *user name* (U), *queue name* (Q), *executable name* (E) and *number of CPUs allocated* (N) are key job attributes that can be used for job categorization. With these attributes we can theoretically define 32 (2^5) different templates. Genetic algorithms can be applied to search for templates with the smallest prediction errors, as is investigated in [6]. In our case we define the template space by heuristics, which can be obtained from the statistical properties of the historical traces. This results in the following templates, which forms a representative job classification and categorizes jobs from coarse to fine granularity:

[G], [G, U], [G, U, Q],
[G, U, E], [G, U, E, N], [G, U, Q, E, N].

We also selected two candidate statistical estimators for quantitative evaluation. They are:

WM(n) An AR(n) (Auto Regressive) model with all coefficients set to $1/n$. This predicts the next sequence value to be the average of previous n values, a simple *Windowed Mean*. AR is one of the Time Series Analysis models, which are investigated extensively in Dinda's work for host load prediction [9].

LR(n) *Linear Regression* [10, 11], where n is the number of previous values used for estimation.

We conduct the quantitative evaluation by actually predicting execution times of historical jobs using traces given in Table 1. Results are shown in Figure 1, 2 and 3.

Firstly we evaluate the results on the NIKHEF EDG production cluster. As can be seen in Figure 1, average prediction errors become smaller as the number of previous values used (n) decreases, both for LR and WM estimators. We select two estimators with the smallest prediction errors, which are WM(1) and LR(5). With respect to templates, we eliminate those with the same number or more attributes but produce no better results and keep [G], [G, U] and [G, U, Q] as our templates for generating predictions. This result is consistent with the statistical properties of NIKHEF EDG traces, where attribute E (Executable name) and N (Number of CPUs allocated) provide no extra information for categorization. Template [G] and [G, U] should be kept since they can provide estimations in case that no dedicated historical data is available in finer grain templates (e.g. [G, U, Q]). For the NIKHEF EDG cluster, the predictors are combinations of the selected templates and the selected estimators: {[G, U, Q], LR(5)}, {[G, U, Q], WM(1)}, {[G, U], LR(5)}, {[G, U], WM(1)}, {[G], LR(5)}, and {[G], WM(1)}. It should be noticed that the number of predictors should be kept small to achieve acceptable performance. A maximum predictor number of 8 would be appropriate in practice.

DAS-2 clusters have a wide variety of users and different kinds of applications. In contrast to the EDG cluster, ex-