RESOURCE PREDICTORS IN HEP APPLICATIONS

S. Grinstein, J. Huth, Harvard University, USA
J. M. Schopf, Argonne National Laboratory, USA

Abstract

The estimation of resource needs for data manipulation is fundamental to the operation of the Grid. Situations will arise when it will be necessary to determine which is more expedient, downloading a replica from a remote site or recreating the data from scratch. This paper explores the possibility of predicting the behavior of the ATLAS applications to improve resource usage in a Grid environment by studying the parameters that affect the execution time performance of event generation, detector simulation and event reconstruction.

INTRODUCTION

The ATLAS (A Toroidal LHC ApparatuS) experiment [1] uses a tiered Grid architecture that enables the replication of datasets across the collaboration sites. These subsets may be overlapping, so a single data set may be available at many locations. We envision a time when users or tools will evaluate which is more expedient, downloading a replica from a site or recreating it from scratch. However, this evaluation will only be possible when we can estimate the execution time of the ATLAS applications.

This paper presents a study to predict the application time for three types ATLAS applications: event generation, detector simulation, and event reconstruction. Our results show that we can achieve predictions within $10 - 25\%$ of the execution time (depending on the application).

ATLAS APPLICATIONS AND MODEL

There are two main factors that affect application execution time: the application parameters that the user inputs to determine the exact course of execution an application follows, and the system platform characteristics, such as CPU speed or network connectivity, seen on the resources running the applications.

ATLAS Applications

We examined the execution time for three classes of ATLAS applications: event generation, full detector simulation, and event reconstruction.

Events for different physics processes were generated using the Pythia event generator [2] through ATLAS's software interface, ATHENA [3]. Full detector simulation was performed for different physics processes, using the ATLSIM [3] package. The reconstruction process is built as a combination of many ATHENA's packages. In the results presented in this paper event reconstruction was performed with the full algorithms using the RecExCommon package.

Environmental Factors

The second factor that determines the execution time is the runtime environment. Once a set of parameters that determines execution time path of the application is selected, and we can predict the application behavior on one platform, we need to evaluate if it is possible to predict the behavior on other systems by scaling the prediction according to the characteristics of the system resources.

We used two platforms in our performance studies: the US Atlas Tier 1 Computing Facility (ACF) [4] located at Brookhaven National Laboratory, and the LxPLUS cluster [5] located at CERN. ACF consists of 60 Pentium III running Linux with a 12 TB disk storage system (SAN based RAID arrays) accessed through an NFS server, and a gigabit network connection used to support both the transfer of files from the High Performance Storage System, and the clients access to data. The LxPLUS cluster consists of 1000 P3 and P4 processors running Linux, with AFS access to disk storage.

Our preliminary results showed that CPU speed was the primary influence when estimating the ATLAS application behavior, so that is the only environmental factor we use in our performance predictions discussed in the next section. We found that the integer index of the *nbench* [6] benchmark best scaled the behavior of the ATLAS applications for new platforms.

Application Model

We began our evaluation of the three ATLAS applications by performing an extensive study of their execution times in a controlled environment and varying a wide set of parameters in order to discover which parameter changes affected the execution time.

In general, HEP applications are embarrassingly parallel applications, meaning that there is no inter-process communication. Because of this, the three ATLAS applications all scale linearly with the number of events to be processed.

We found that the event type, or size, also has a large impact on running time. For example, events with a lot of energy in them produce a large number of particles, which increases the data volume of the event, thus increasing the time required to process the event. This is especially true for detector simulation and event reconstruction since the number of CPU cycles used by these applications may in-

crease dramatically with the occupancy of the events. In our approach we use the average number of particles in the Monte Carlo sample being processed as a measure of the size of the event.

The release version of the ATLAS software has a large impact on execution performance, since variations in the algorithm implementations can have an effect on execution time.

The impact of the parameters that steer the operations (for example, the minimum momentum of the tracks), whether dealing with generation, simulation or reconstruction, may also affect execution time. We studied the effect of changing the application parameters that determine the track, jet, muon and electromagnetic object reconstruction. However, when changed within reasonable limits, none of these parameters had a large impact on performance.

The process to obtain the execution time prediction for a certain application is the following. The application behavior is studied in a *benchmarking* environment, *i.e.*, using a certain platform system, Monte Carlo physics sample and release version. The dependence on the number of events (N) of the application execution time is parameterized as $a_1\,N + a_2$. The application is run for different number of events, and the parameters a_1, a_2 that best fit the results are obtained.

The predicted application execution time (T) in a different environment is obtained using the formula:

$$T = (a_1\,N + a_2)\,S\,H\,t \tag{1}$$

where

- S is the ratio between the average number of particles in the current and the benchmarking samples,
- H is a factor that accounts for the system performance (obtained with *nbench*), and
- t is a number that scales the prediction if the executable is not optimized.

For example, if we know the behavior of the reconstruction application in a given environment and we want to make a prediction for the execution time in a different host and using a different sample, we would scale the know prediction by $S = N_p/N_p^0$ and $H = H_p/H_p^0$, where N_p (N_p^0) is the average number of particles in the new (benchmark) sample, extracted from the Monte Carlo information, and H_p (H_p^0) is the *nbench* integer index of the new (benchmark) host. If we were to change the release version of the code the parameters a_1 and a_2 have to be obtained for the new version, by studying it's performance in the benchmark environment. The same applies to the simulation application. Since the generation is a more simple application, it's behavior can be predicted regardless of the data sample being generated.

The benchmark environment consisted of the 1 GHz ACF machines, using a MSSM sample with the ATHENA release version 7.0.0. The results presented below were obtained in the different hosts of the ACF and LxPLUS clusters, for different event samples and code release versions (where indicated).

RESULTS

ATLAS Event Generation Application

We ran the ATLAS generation application on both the ACF and LxPLUS clusters using the ATHENA's Pythia interface (release version $7.0.0$). Six different physics processes were produced: jet production, Z+jet events, H(130)→ZZ→4l, WH(400)→ $\mu\nu$bb, H(400)→hh→bbbb, and $t\bar{t}$ (unbiased decays).

As expected, event generation was found to scale well with the total number of events as the determining parameter. The different physics processes introduced a small variation in the execution performance. The average time per event on the ACF 1 GHz processors was of 0.15 seconds, with an overhead of 1.5 seconds. The number of events and the CPU benchmarks were the only parameters used to predict the execution time of the event generation. The results showed that the prediction was within 6% of the actual execution time, when generating 50 to 10,000 events for the different physics processes at the ACF and LxPLUS clusters.

ATLAS Event Simulation Application

The ATLSIM package was used at ACF to perform full detector simulation. Since detector simulation is very resource consuming, both in terms of CPU usage and disk space, samples of limited size where simulated.

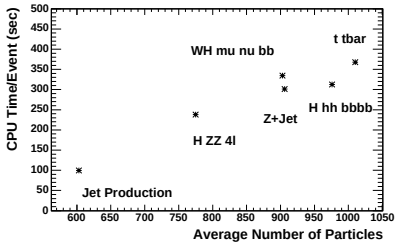

Figure 1: Execution time per event of the full detector simulation as a function of particle multiplicity for different physics processes.

The full simulation of each type of data set produced for the event generation studies was performed, but only subsamples of up to 100 events were simulated. The simulation of the different processes resulted in very different resource usage. In the 1 GHz nodes, the average time required to perform the simulation ranged from 100 seconds per event for the QCD sample, to 335 seconds for the $t\bar{t}$ production sample. The resource usage by the different processes was found to scale with the average number of

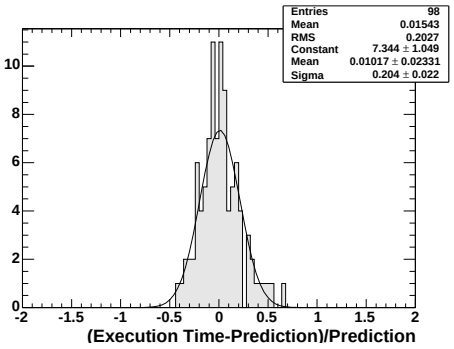

Figure 2: Difference between the actual CPU time per job and the predicted time, normalized to the prediction, for QCD simulation jobs executed at BNL CAS farm. The distribution is fitted with a Gaussian function.

particles in the different samples. Figure 1 shows the execution time per event as a function of the average number of particles per event in the sample. The number of particles was obtained from the generation log files (all particles are counted).

The behavior of the simulation was predicted from the platform information (benchmark of the execution node), the number of events to be simulated, the average number of particles per event in the sample, and the $a_{1,2}$ parameters for the release version. With these parameters the differences between the actual execution time and the prediction normalized to the prediction, in the different hosts of the ACF cluster, for samples with 10 to 100 events, were within 25%. Figure 2 shows the comparison between the predictions and the actual execution times for simulation jobs of QCD events (jet production).

ATLAS Reconstruction Application

Predicting the resource usage of the reconstruction code is difficult due to the large fluctuations in the CPU time required per event. Figure 3 shows the execution reconstruction time per event using RecExCommon (release 7.0.0) and the SUSY sample, on the 1 GHz hosts of the CAS farm. A Landau fit to the distribution is shown, the most probable value (MPV) for the reconstruction time per event is 9.96 seconds, while the mean is 16.38 seconds. Some of the most time consuming events were visually scanned using the Atlantis [7] event display program, but no obvious problem was detected.

The information of the average time per event, together with the platform, optimized code choice, and the release version, was used to predict the execution time. Debugged executables were found to be about 8.5 times slower than optimized code. Figure 4 shows the CPU time of 350 jobs with 10 to 200 randomly distributed events per job as a function of the predicted time. The plot shows that there is very good correlation (0.96) between the actual execution time and the prediction. Figure 5 shows the difference between the CPU time per job and the predicted time, normalized to the prediction. This result is plotted as a function of the predicted time in Figure 6, which indicates that, in most cases, the prediction is within 10% of the CPU time.

ATHENA's reconstruction execution time prediction at LxPLUS was obtained by using the results gathered at ACF (for a_1, a_2, S, r, and t) and the benchmark information of the LxPLUS hosts (obtained with *nbench*). Figure 8 shows the difference between the CPU time per job and the predicted time, normalized to the prediction. Figure 9 indicates that the prediction is within 10% of the CPU time in most cases.

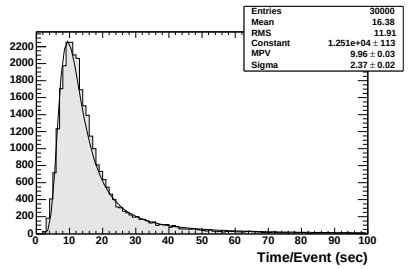

Figure 3: Execution time per event for the full reconstruction chain using RecExCommon (release 7.0.0) and the SUSY sample.

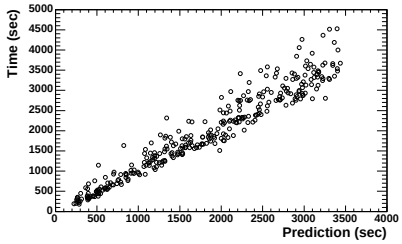

Figure 4: Correlation between execution time and predicted time for jobs with random number of events (between 10 and 200) executed at BNL's CAS farm. Every entry on this plot is independent of every other one. The correlation coefficient of the distribution is of 0.96.

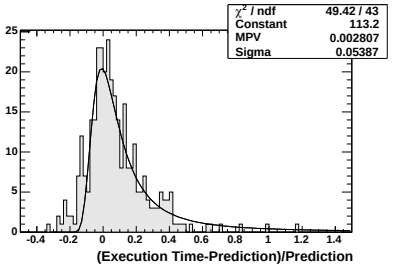

Figure 5: Difference between the actual CPU time per job and the predicted time, normalized to the prediction; for jobs executed at BNL CAS farm, using different input data files. The distribution is fitted with a Landau function, the result shows that 0.003 is the most probable parameter of the density.

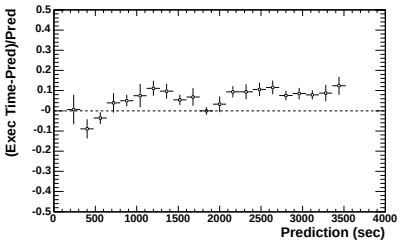

Figure 6: Difference between the actual CPU time per job and the predicted time, normalized to the prediction, as a function of the predicted time.

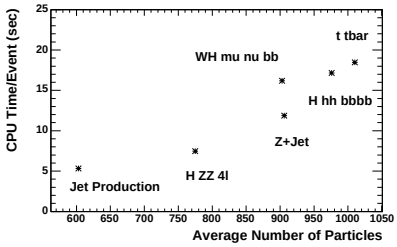

Figure 7: Execution time per event of ATHENA's reconstruction algorithms as a function of particle multiplicity for different physics processes.

To study the effect of different data samples on execution time, the six samples used for detector simulation (see Section) were reconstructed. Figure 7 shows the execution time per event as a function of the average number of particles per event in the sample. As seen in Figure 1, the reconstruction time is proportional to particle multiplicity.

The impact of the version of the reconstruction code on on execution performance was studied. The results obtained with release version $7.0.0$ were compared with versions $6.5.0$ and $7.2.0$. Version $7.2.0$ was found to be 1.4 times slower than $7.0.0$, while $7.0.0$ was found to be 1.6 times slower than $6.5.0$. This is not a surprise, since the ATLAS software is in it's development stage.

CONCLUSION

In this paper we have shown that it is possible to predict the execution time of the ATLAS software (including event generation, full detector simulation and event reconstruction) with an accuracy of 90% (generation and reconstruction) to 75% (simulation). The applications exhibit a largely deterministic behavior, requiring only six parameters to obtain such predictions: the CPU speed of the execution host, the type of executable (optimized or not), a measure of the average size of the events (like the average number of particles per event), and the number of events to process, together with the two parameters that describe the linear dependency of the execution time with the number of events for a given release version. These parameters are sufficient to obtain the degree of accuracy desired for this work, however, new parameters could be added to improve

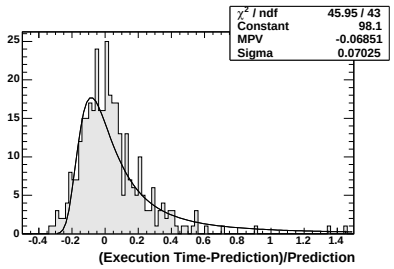

Figure 8: Difference between the actual CPU time per job and the predicted time, normalized to the prediction; for jobs executed at CERN's LxPLUS cluster, using different input data files. The distribution is fitted with a Landau function, the result shows that -0.07 is the most probable parameter of the density.

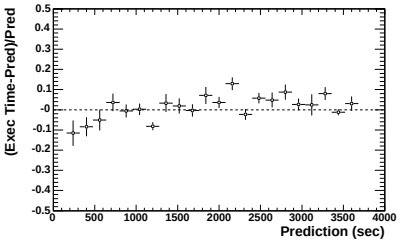

Figure 9: Difference between the actual CPU time per job and the predicted time, normalized to the prediction, as a function of the predicted time (LxPLUS).

the precision of the prediction and adapt to different system configurations and environments.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Number PHY-0218987.

REFERENCES

[1] ATLAS Collaboration, "ATLAS Detector and Physics Performance Technical Design Report, CERN-LHCC-99-14-15" (1999).

[2] T. Sjöstrand, P. Edén, C. Friberg, L. Lönnblad, G. Miu, S. Mrenna and E. Norrbin, Computer Phys. Commun. 135 (2001) 238.

[3] P. Calafi ura, *et. al.*, "The Athena Control Framework in Production, New Developments and Lessons Learned", this proceedings.

[4] "US Atlas Tier 1 Computing Facility", http://www.acf.bnl.gov/

[5] "PLUS Service and Machines", http://plus.web.cern.ch/plus/machines.html

[6] "Linux/Unix nbench", http://www.tux.org/∼mayer/linux/bmark.html

[7] J. Drohan, *et. al.*, "The Atlantis event visualisation program for the ATLAS experiment", this proceedings.