Adapting master-worker paradigm for high throughput applications in grid environment

Jan Pieczykolan[1], Lukasz Dutka[1], Krzysztof Korcyl[1,2], Renata Slota[3], Tomir Kryza[1,3] and Jacek Kitowski[1,3]

[1] Academic Computer Centre CYFRONET-AGH, Nawojki 11, 30-950 Cracow, Poland
[2] Institute of Nuclear Physics, PAN, 31-342 Cracow, Poland
[3] Institute of Computer Science AGH University of Science and Technology, Mickiewicza 30, 30-059 Cracow, Poland

Abstract. The paper presents a concept and evaluation of Real-Time Dispatcher (RTD) – a system that provides an grid implementation of master-worker paradigm where execution time of data streams is short (O(1s)) with high throughput of data streams. It is based on separation of grid jobs initiation from execution. The architecture, implementation details and current limitations of RTD are discussed. RTD potential application to a High-Energy Physics (HEP) application – an extension of the High-Level Trigger and Data Acquisition (HLT/DAQ) system, a part of the LHC ATLAS project with high throughput requirements is also outlined.

Keywords: grid computing, high throughput computing, grid master-worker paradigm, HEP LHC ATLAS experiment.

1 Introduction

Most of the grid execution models assume off-line grid usage with batch processing paradigm and loosely defined execution time requirements for an application. Therefore, it is hard to use grid environments for on-line processing of data-streams or interaction with humans, where the QoS level, defined as maximum processing time, must be guaranteed.

The Grid, best known as a solution for performing batch computation using jobs, continues to expand. Currently, most research is focused on semantic or knowledge grid, but the full potential of computational grid is still unveiled. The available grid software offers a rich set of services that combined together with a dedicated solution may facilitate the process of delegating computation from an application to the grid infrastructure.

There exist numerous solutions that aid computation delegation either to Network of Workstations or to the Grid. These solutions have been constructed for over 20 years and they have grown mature and increasingly complex over time. Diverse systems such as Condor, Condor-G, Mosix, BOINC or COSM offer rather complete support for managing available resources and provide tools for creating a common problem space for multiple machines. These solutions, constructed and improved over a long period of time, provide a great deal of capabilities at present. A good example of development of interactivity features for grid applications are achievements of the CrossGrid project [1] as well as wide research in the field of interactive grid monitoring [2, 3], adaptive resource brokering or client-server solutions for Grid [4]. Another useful solution is to adapt the existing environments for requirements not considered previously rather than creating a dedicated, complete grid middleware for interactive or data-streaming applications with defined QoS (cf. gLite [5] or Legion [6]) using additional services specific to these kinds of applications. Other examples supporting interactivity, master-worker or computation delegation paradigms in grid environments are Condor MW [7], split execution [8] and interposition agents [9]. Adaptive resource broker [10] to support job migration for controlling interactivity is also possible.

The main goal of this paper is to propose a lightweight implementation of a master–worker paradigm in the grid environment that makes use of grid resources to serve application requests with soft real-time requirements defined on execution of processing streams of data. The Quality of Service can be defined in terms of high throughput. The main feature of the concept is fast and effective delegation of parts of computation to the grid, which consists in sending data to awaiting programs that have been submitted to the grid in advance.

The perspective usage of the proposed concept is grid processing of event data streams from the High-Level Trigger and Data Acquisition (HLT/DAQ) system of the Atlas experiment, which will be performed on Large Hadron Collider (LHC) at CERN. The architecture and initial evaluation results of a software solution that meets the requirements of the concept, the Real-Time Dispatcher (RTD), are presented.

2 The Problem

In grid computing one of the most important factors of the QoS is a delay, related to waiting for job results. The delay is composed of two parts:

- *grid delay* – related with submission and scheduling,
- *job delay* – related with processing time.

The job delay is related to a particular processing algorithm that is used to process input data. Its optimization is possible only by modification of this algorithm, which is an application developer task.

The delay related with the grid environment is a sum of delays introduced by grid services, like Job Submission or Scheduler Service, it is also related with current grid workload. It appears every time when a job is submitted.

An idea of the mentioned solution uses the master–worker paradigm to allow delegation of computation outside of a particular application to the grid without engaging job submission service and scheduler.

3 The System

3.1 System concept

The grid user, using the user interface, creates a pool of jobs called worker processes. These processes register within the RTD and wait until the RTD passes them a request from the application that acts as a master. When an application needs additional computational power it sends a request for a worker to the RTD. When the RTD receives the application's request, it uses the Grid monitoring system to find the most optimal worker and passes it the hostname and port on which the application is listening. As an effect, the worker connects to the application, which acts as a master, as presented in Fig. 1 and it is ready to serve its requests. At this point, the role of the RTD is finished. Passing data, carrying out computation and sending the results is accomplished directly via the created connection between the master and the workers.

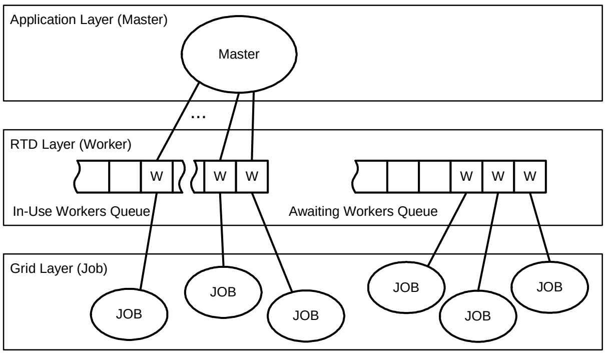

Fig. 1. The RTD placement in grid-based solutions

3.2 System Architecture

The RTD system architecture consists of the following modules (cf. Fig. 2):

- Frontend Interface – module responsible for gateway interface to the system,
- Backend Interface – module responsible for communication with the worker running on the computing nodes on the grid,
- Resource Manager with Resource Registry – module responsible for managing available workers,
- Engine – module that implements task dispatching,
- Monitor – module responsible for gathering data from grid monitoring systems concerning the state of computing nodes on the grid in order to supply the system with information required for optimal selection of the available worker.

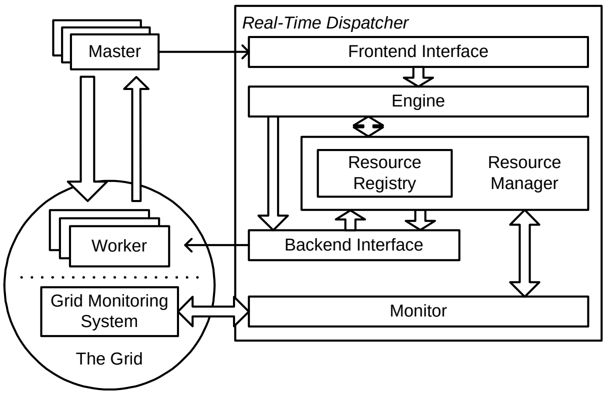

Fig. 2. High-Level Design of the Real-Time Dispatcher architecture

In order to increase reliability and efficiency of the RTD, its modules are constructed in a way that allows placing them on separate cluster nodes. Such a construction of the system is useful for multiplication of module instances and balancing their load, as well as taking over tasks of a damaged instance.

The system is designed according to OOAD and implemented in Java. The choice of the language is based on its multi-platform character, available mechanisms and libraries.

4 HEP application

The High-Energy Physics application is a kind of extension of the software developed within the Atlas project. Currently, the Atlas software sends the events to process to the computing farms located at CERN. The processing consists of analyzing data from detectors installed at the CERN accelerator to indicate if a particular event can be interesting for further, more sophisticated analysis. According to the proposed solution the HEP application is responsible for sending a part of the events to the Grid virtual organization instead to the CERN computing farm. Thus, the analysis of these events is to be performed on the grid infrastructure.

The main characteristics of this application are:

- a large stream of data – data is generated by particles registered by the detectors; after aggregation of data fragments into events and filtration performed at the earlier stages of processing, the resulting data ow, which reaches 3.5 kHz rate of 2MB event size, sets QoS requirements of the application;
- limited computation time of a single event to a few seconds (∼2-3s to be tuned). The events, which computational time is longer, are accepted for further analysis by sending them to a permanent storage system. The rest of the events is neglected (since the physical features which they represent are

not scientifically interested). This limitation of computational time defines soft real-time requirements.

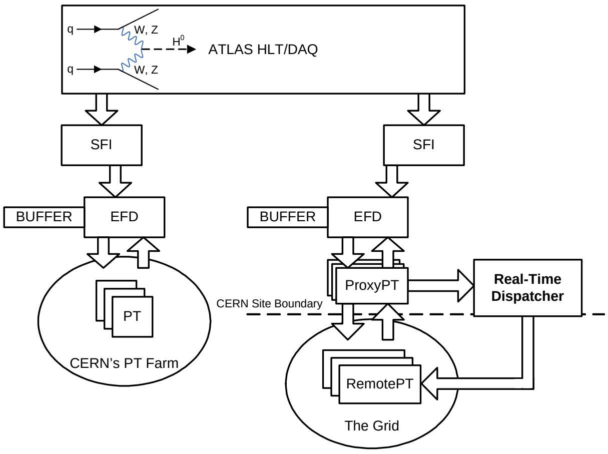

Fig. 3. The architecture of HEP application using grid

Those two characteristics of the application reveal its soft real-time requirements. Contrary to the typical (batch) grid processing, the HEP application requires on-line processing of small tasks due to the stream of event data from the Atlas HLT/DAQ system (cf. Fig. 3). The events obtained from Atlas HTL/DAQ via Sub-Farm Interfaces (SFI) are allocated by Event Filter Dataflow Control Program (EFD) as Processing Tasks (PT). The HEP application requires quick event response, since new events to analyze follow rapidly.

The system evaluation and performance issues will be discussed in the full version of the paper.

5 Summary and further work

A concept of effective delegation of parts of computation to grid resources implemented according to the master–worker paradigm, its implementation and performance evaluation have been presented in the paper. Due to rather high performance of the RTD it is possible to extend the number of master-worker connections on-line, to follow the required computing load of the master.

Future work is focused on making use of communication monitoring using the JIMS software package in order to improve the reliability of the system. Different strategies of worker selection based on data from the monitoring system will be elaborated together with dynamic worker pool sizing using predicted master load.

The final version will be tested for the purpose of the ATLAS HLT/DAQ system within the int.eu.grid project.

Acknowledgment

This research has been done in the framework of EU IST 031857 int.eu.grid project, AGH-UST grant is also acknowledged.

References

1. The CrossGrid Project,http://www.crossgrid.org
2. Steenberg, C., Hsu, S.C., Lipeles, E. and Wuerthwein, F. : JobMon: A Secure, Scalable, Interactive Grid Job Monitor, in Proc. of Computing for High Energy Physics, Mumbai, India, February 13-17, 2006.
3. Bunn, J., Bourilkov, D., Cavanaugh, R., Legrand, I., Muhammad, A., Newman, H., Singh, S., Steenberg, C., Thomas, M., van Lingen, F.: Proposal for a Grid Analysis Environment Service Architecture, internal note, NUST Institute of Information Technology, California Institute of Technology, 2003.
4. Caron, E. and Desprez, F.: DIET: A Scalable Toolbox to Build Network Enabled Servers on the Grid, International Journal of High Performance Computing Applications, 20(3):335-352, 2006.
5. Laure, E. at al.: Middleware for the next generation Grid infrastructure, in Proc. of Computing in High Energy Physics and Nuclear Physics, CHEP 2004, Interlaken, Switzerland, 27 Sep - 01 Oct, 2004.
6. Grimshaw, A.S., Wulf, W.A. and the Legion team: The Legion Vision of a Worldwide Virtual Computer, Commun. ACM 40(1): 39-45, 1997.
7. Goux J-P., Kulkarni, S., Linderoth, J. and Yoder, M.: An Enabling Framework for Master-Worker Applications on the Computational Grid, Conf. Proc. of the Ninth IEEE Symposium on High Performance Distributed Computing, Pittsburg, PA, August 2000, pp. 43-60.
8. Thain, D., Livny, M.: Bypass: A tool for building split execution systems, Conf. Proc. of the Ninth IEEE Symposium on High Performance Distributed Computing, Pittsburg, PA, August 2000, pp. 79-85.
9. Jones, M. B.: Interposition agents: Transparently interposing user code at the system interface, Conf. Proc. of the 14 th Symposium on Operating Systems Principles, 1993, pp. 80-93.
10. Othman, A., Dew, P., Djemame, K. and Gourlay, I.: Toward an Interactive Grid Adaptive Resource Broker, in Proc. of the UK e-Science All Hands Meeting, Nottingham, UK, September 2003, pp. 385-388.