Enabling Autonomic Meta-Scheduling in Grid Environments

Yanbin Liu[1], S. Masoud Sadjadi[2], Liana Fong[1], Ivan Rodero[3], David Villegas[2], Selim Kalayci[2], Norman Bobroff[1], Juan Carlos Martinez[2]

[1] IBM Watson Research Center, Hawthorne, NY, USA, {ygliu,llfong,bobroff}@us.ibm.com

[2] Florida International University, Miami, FL, USA, {sadjadi, dvill013,skala001,jmart054}@cs.fiu.edu

[3] Barcelona Supercomputing Center, Barcelona, Spain, irodero@ac.upc.edu

ABSTRACT

Grid computing supports workload execution on computing resources that are shared across a set of collaborative organizations. At the core of workload management for Grid computing is a software component, called meta-scheduler or Grid resource broker, that provides a virtual layer on top of heterogeneous Grid middleware, schedulers, and resources. Meta-schedulers typically enable end-users and applications to compete over distributed shared resources through the use of one or more instances of the same meta-scheduler, in a centralized or distributed manner, respectively. We propose an approach to enabling autonomic meta-scheduling through the use of a new communication protocol that –if adopted by different meta-schedulers or by the applications using them— can improve the workload execution while avoiding potential chaos, which can be resulted from blind competition over resources. This can be made possible by allowing the meta-schedulers and/or their applications to engage in a process to negotiate their roles (e.g., consumer, provider, or both), scheduling policies, service-level agreement, etc. To show the feasibility of our approach, we developed a prototype that enables some preliminary autonomic management among three different meta-schedulers, namely, GridWay, eNANOS, and TDWB.

Keywords: meta-scheduler, grid resource broker, grid interoperability, autonomic workload management.

1. INTRODUCTION

Grid computing supports workload execution across computing resources from cooperating organizations or institutions, which form a virtual organization (VO) [1]. With appropriate management system and polices supporting workload execution and resource usage, the users of such Grid systems can be benefited from increased availability of computing resources while the participating organizations can still maintain their autonomy and fully utilize their own resources, if required. At the core of a Grid system is the management entity, commonly known as a meta-scheduler or grid resource broker, which matches the resources to workload requests for execution based on policies (e.g., workload service objectives, resource usage criteria, etc.).

The need for interoperability among Grid systems reflects the reality that there are numerous organization and institutions that would like to collaborate and share their resources, but still need to operate independently and autonomously. Our interoperable model supports autonomy of organizations and addresses the scalability issues in managing very large numbers of resources, and avoids the complexity of an alternative approach in mapping resources to multiple organizations.

Different architectures have been proposed for these interoperating meta-scheduling systems, including HPC-Europa SPA [2], GridWay [3], Koala [4]. Our architectural design supports schedulers in partnering relations that: (i) can be a hybrid of distributed and hierarchical; (ii) can be dynamically established and changed over time; and (iii) can be of different roles with different policies.

2. PEER-TO-PEER META-SCHEDULING

Our collaborating meta-scheduling architecture consists of multiple resource domain sites that are independently managed and operated. Thus, domains are expected to vary widely in computing and storage capabilities, grid middleware, cluster managers, local schedulers, and policies for accepting and executing jobs. Furthermore, resource availability and policies are not static, and may change within the lifecycle of long running jobs.

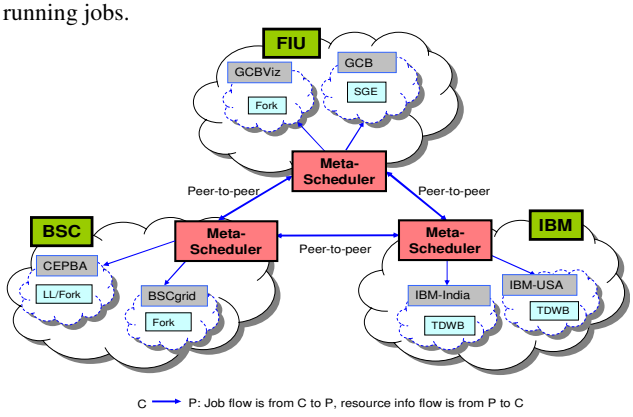

Figure 1. Cooperating meta-scheduling in LA Grid.

Figure 1 shows interconnected meta-schedulers at three diverse institutions of IBM, Florida International University (FIU), and Barcelona Supercomputing Center (BSC). The figure also illustrates the hybrid model such that it follows a peer-to-peer distributed model for interactions between domain meta-schedulers, while it follows a hierarchical model for interactions

978-0-7695-3175-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ICAC.2008.15

between a meta-scheduler and its local schedulers within the same domain. Note that there is no direct interaction between a meta-scheduler from one domain and the local schedulers of other domains.

Though the meta-schedulers have heterogeneous implementations, they adhere to a common set of communication protocols and information encapsulation standards that allow them to interoperate. Table 1 shows the protocols designed and implemented. In addition, we assume that the resource requests expressed by workloads are described in a common language (JSDL [5]).

For the autonomic management of meta-schedulers, the starting point is the negotiation of desirable connection between the collaborating partners. The negotiated parameters include the roles, the rate that heartbeats should be exchanged to monitor connection status, type of authentication, and potentially the quality of service agreements.

TABLE I
List of possible messages in the LA Grid Meta-scheduling Protocol

Connection Messages	Resource Information Messages	Job Execution Messages
openConn()	*requestResourceData()*	*submitJob()*
notifyConn()	*sendResourceData()*	*queryJob()*
Heartbeat()		*notifyJob()*
		cancelJobl()

The possible role of a partner meta-scheduler in our design includes consumer, provider, or peer. A consumer meta-scheduler submits resource requests to provider meta-scheduler that have resources to execute the requests. Two partnering meta-schedulers are peers if they send and execute requests between each other.

Any party - provider, consumer, or peer – can initiate a connection by sending an openConn() message with some suggested parameters. If the remote party agrees with the parameters, it starts sending heartbeats, otherwise it counters with a new openConn() message proposing alternative parameters. Negotiation continues until agreement is reached or the number of rounds exceeds a threshold specified by the initiator, in which case the connection attempt fails. After a connection is established, the notifyConn() message is used to send information about the connection. It is also used to gracefully end the connection. Partners can renegotiate their roles after terminating their current connection.

Once a connection is established, resource information is sent to the consumer meta-scheduler either in pull mode (using requestResourceData()) or push mode (sendResourceData()). The push mode is also used when updates are triggered by dynamic changes in resource capacity, utilization, or availability. Resource updates may be complete or incremental. Complete updates are typically requested in pull mode by the consumer. Incremental updates are generally pushed by the provider when the resource availability/load changes in the domain. If a provider meta-scheduler has connections to multiple other providers, in a simple tree interconnection, for example, it may pass to any attached consumer meta-scheduler the expression of the full range of resources to which it has access according to its policies. In our design, we model a scheduler as a resource that has attributes including the scheduling policies (e.g., priority based, first-come-first-serve based, etc.), the capability (e.g., parallel jobs) and the utilization (e.g., current total number of jobs, mean job turn-around time). Using the resource information of all the provider meta-schedulers and its own local schedulers, a consumer meta-scheduler can intelligently make a decision and distribute its workload over the local and remote resources according to its scheduling, quality of service, and service-level agreement policies.

3. CURRENT STATUS AND FUTURE WORK

The collaborating meta-scheduling architecture and the common protocols were implemented by three LA Grid [6] partners (namely, BSC, IBM, and FIU) using independent implementations of the meta-scheduler, local scheduler, and Web services technologies. BSC's prototype uses eNANOS [7], IBM Research prototype uses IBM product ITDWB [8], and FIU's prototype uses Gridway [3]. Our current implementation achieves the inter-operation among the three meta-schedulers. We will evolve the current prototype in different directions: richer set of meta-scheduling functions and protocols, richer set of local and global polices, separating job/user from resource policies, optimization for job to resources and domain site matching, policy agreement negotiation, sophisticated service-level agreements, etc. Moreover, we are planning to make our meta-schedulers available to our other LA Grid partners to explore the effectiveness of our approach in a variety of application areas such as Hurricane Migration, Bioinformatics, and Healthcare [9].

Acknowledgment: This work was supported in part by IBM and the National Science Foundation (grants OISE-0730065, OCI-0636031, REU-0552555, and HRD-0317692).

REFERENCES

[1] I. Foster, C. Kesselman, editors, "The Grid 2: Blueprint for a New Computing Infrastructure", Morgan Kaufmann Publishers, 2003.

[2] HPC-Europa Web Site. http://www.hpc-europa.org

[3] GridWay: http://www.gridway.org/

[4] A. Iosup, D.H.J. Epema, et al. "Inter-Operable Grids thorugh Delegated MatchMaking", in proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC07), Reno, Nevada, November 2007.

[5] A. Anjomshoaa, M. Drescher, et. al. "Job Submission Description Language (JSDL) Specficiation"; Version 1.0, 2005.

[6] LA Grid Initiative: http://latinamericangrid.org/

[7] I. Rodero, R.M. Badia, et. al. "eNANOS Grid Resource Broker", European Grid Conference 2005, LNCS 3470, Amsterdam, The Netherlands, 14-16 February, 2005.

[8] IBM's Tivoli Dynamic Workload Broker, http://www-306.ibm.com/software/tivoli/products/dynamic-workload-broker/index.html

[9] R. Badia *et. al.* "Innovative Grid Technologies Applied to Bioinformatics and Hurricane Mitigation". High Performance Computing and Grids in Action, IOS Press - Amsterdam, Lucio Grandinetti editor. Nov/Dec 2007.