On the Improvement of the Network QoS in a Grid Environment

(Extended abstract)

Agustín Caminero
UCLM. Albacete. Spain
agustin@dsi.uclm.es

Carmen Carrión
UCLM. Albacete. Spain
carmen@dsi.uclm.es

Blanca Caminero
UCLM. Albacete. Spain
blanca@dsi.uclm.es

ABSTRACT
Grid systems are highly variable environments, made of a series of independent organizations that share their resources, creating what is known as virtual organization, VO. This variability makes quality of service (QoS) highly desirable, though it is very complex to achieve due to the large scale of interconnected networks. Entities entrusted with the care of the users’ interests in a Grid, are known as meta-schedulers or Grid Schedulers [2] [1]. These entities usually take into account the power of computing resources in order to decide which resource will run an user’s job, but do not pay attention to the interconnecting network. The provision of network QoS in a Grid environment is the topic of interest of this work, and what we want to do is to develop a meta-scheduler that pays attention to the network when making decisions. This new entity, known as grid network broker, GNB, will be developed at first using a simulation tool. The simulation tool that we have chosen is GridSim [5], because it provides us with a good infrastructure, including an implementation of Differentiated Services [3]. The implementation of differentiated services in GridSim is the Self Clocked Fair Queuing (SCFQ) scheduler [4]. A SCFQ scheduler can provide differentiated service to traffic by changing the weights associated with a certain class of traffic. The higher the weight of a class of traffic is, the better treatment it receives.

Our aim is firstly develop our GNB on a simulator and, when our model is tested, we will proceed with its implementation on a real grid environment.

categories and Subject Descriptors
C.2.4 [Distributed Systems]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MGC ’06, November 27, 2006 Melbourne, Australia Copyright 2006 ACM 1-59593-581-9/06/11... $5.00

1. architecture of the GNB
Our grid network broker (GNB) is entrusted with the resource allocation and admission control, and it will have a global knowledge of the topology of the VO. It will have the architecture shown in Figure 1, whose main parts will be explained the next.

Figure 1: Architecture of our GNB.

- Data base: This module keeps some useful information, i.e., the routing tables of every router of the VO.
- Connection admission control (CAC): This module checks whether a new connection can go through every link of the network path between the user and a computing resource.
- Scheduler: This module decides which computing resource will be the best one to run each of the users’ jobs. To do that, several algorithms could be implemented. The initial algorithm studied here takes into account the power of the computing resource and the network path between the user and each resource. Each of these two parameters is given a weight, which is the significance of this parameter when calculating the quality of a resource.

Taking this into account, the “Scheduler” will order the resources from the best to the worst one, and then the “CAC” module will check the path from the user to the best resource. If the CAC module rejects this connection, then it will try with the second best resource, and so on. If the
CAC module has rejected the connections between a user and all the computing resources, then some of the jobs of that user will not be run at any resource. In this case, when the user receives back any of the accepted jobs, it will ask again for a computing resource for those jobs which were previously rejected. In the end, all the jobs of all the users will be executed.

2. EXPERIMENTS AND RESULTS

We have run some simulations in order to assess the functionality of our grid network broker, which consider one virtual organization with the topology showed in Figure 2. In this topology, User 0 belongs to the class type 2, whose weight is 3, and User 1 belongs to the class type 1, whose weight is 2. This is, user 0 has more priority than user 1 for the SCFQ schedulers. Each user tries to run 15 gridlets (jobs), each one with these features: input/output files sizes are 70MB, and processing power 7000 MI (million of instructions). Two of the computing resources (resources 1 and 2) are more powerful than the other two (resources 0 and 3). The more powerful computing resources are identical, as the less powerful ones. A space shared policy or First Come First Served (FCFS) algorithm is used to compute incoming jobs in our computing resources. The more powerful computing resources have 1 machine made of 4 CPUs with a rating of 1500 MIPS (Million of Instructions Per Second) each one, whilst the less powerful ones have the same number of machines and CPUs, but each CPU has a rating of 750 MIPS. Our links have a baud rate of 2 Mbps, a propagation delay of 10 milliseconds and a MTU of 1500 bytes, and our users request the 20% of the link bandwidth for each job. We also have a GIS (Grid Information Service) entity.

Figure 3 shows some performance results where the GNB uses different configurations of weights, which appear in brackets (power/weight/network weight). It shows the total latencies suffered by each gridlet at each simulation, which is the time each gridlet takes to be sent by the user, reach the resource where it will be run, wait for a CPU, be run, be sent back to the user and reach the user. This statistic does not include the negotiation process between the user and the broker. When there is no CAC, the GNB only performs the scheduling of gridlets to resources, and all the gridlets are sent to their resource at the same time. But when the CAC is enabled, it performs an admission control, allowing only a number of transmissions at the same time along each link, rejecting the transmissions that would overuse each link. When those gridlets that were initially accepted by the GNB are back at their users, users will contact the broker again in order to find a resource to run the gridlets which were initially rejected. Because of this, network load is well-balanced over time, which means that the time users have to wait for a gridlet after it has been submitted to a resource is very much lower when the CAC module is enabled.

3. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a network-aware grid meta-scheduler, also known as grid network broker, or GNB. This entity focuses on providing efficient QoS mechanism inside the network environment. The GNB would be entrusted with the care of the utilization of the links, so that links do not become overused, thus preventing the entire network performance from degrading. Our grid network broker will allow different algorithms to be implemented within itself, so that different scheduling and admission strategies can be studied. We have showed some experimental results supporting the idea that our GNB improves the network QoS that users receive. Looking at the results presented here we conclude that the network performance of a grid system is improved by the utilization of the GNB, as it performs job scheduling and network load balancing that takes advantage of the features the system. We can also conclude that the best results have been those obtained when the broker gave more significance to the network than to the resource power when performing the scheduling.

4. ACKNOWLEDGMENTS

We would like to thank the developers of GridSim, especially Anthony Sulistio, for their valuable help at understanding the functionality of GridSim.

5. REFERENCES


