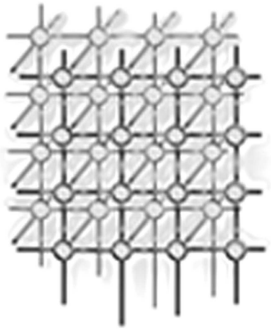
Price-sensitive resource brokering with the Hybrid Pricing Model and widely overlapping price domains



Rosario M. Piro[1,*], Andrea Guarise[1] and Albert Werbrouck[1]

[1] *Istituto Nazionale di Fisica Nucleare (INFN) - Sezione di Torino, Via P. Giuria 1, 10125 Torino, Italy.*[†]

SUMMARY

The *DGAS-Sim(ulator)* simulates the components of the grid middleware that are involved in the job scheduling process, reflecting the architecture of the Workload Management System of the European DataGrid project. Its purpose is to study the impact of different resource pricing schemes and resource brokering strategies on workload balancing. In this paper we present further simulation results for a *price-sensitive* utility function (the Resource Broker always chooses the most economic Computing Element) and the *Hybrid Pricing Model* (fixed base prices and linear price adjustments within fixed variation limits to reflect the current workload of the Computing Element). We specifically consider a Grid economy scenario with *widely overlapping* domains of the prices of the single resources and compare the outcome with our previously presented results.

KEY WORDS: Grid computing; economic brokering; resource pricing; load balancing

Introduction

DGAS-Sim, presented in [5], was designed to evaluate the performance of different pricing schemes for the *DataGrid Accounting System (DGAS)* [4], an infrastructure for the accounting of resource usage in computational grids that was developed for the Workload Management System [9] of the *European DataGrid (EDG)* project [1]. DGAS provides the possibility to dynamically adjust resource prices and thus can enable a Resource Broker (RB) to base scheduling decisions on economic models.

Our main purpose for applying economic principles to the resource selection problem — in addition to eventually reaching a market equilibrium where the computational energy (defined in the following

*Correspondence to: R. Piro, Istituto Nazionale di Fisica Nucleare (INFN), Via Pietro Giuria 1, 10125 Torino, Italy
[†]E-mail: {piro,guarise,werbrouck}@to.infn.it

section) requested by the users equals the computational energy provided by the grid resources — is to balance the workload *among* the available resources in order to improve the grid's throughput. This requires the adoption of pricing schemes that determine *local* prices for the single resources based on their workload, instead of computing *global* equilibrium prices (one for each resource type) as proposed by much related work.

In [5] we presented the first simulation results for the *Hybrid Pricing Model (HPM)* combined with a simple *price-sensitive resource brokering* strategy — where the Resource Broker (RB) selects the cheapest resource that matches a given job's requirements — and compared them with a "worst case" and a "best case" scenario, concluding that applying the HPM with a single base price for all resources could approximate the best case. In the following sections we will summarize the simulation models and the previously published results. We then present additional simulation results that emphasize the importance of widely overlapping price domains for the purpose of load balancing with the HPM, since widely overlapping price domains allow to obtain similarly effective load balancing even if base prices differ to reflect the resources' different "qualities" (in terms of CPU performance, etc.).

DGAS-Sim

DGAS-Sim [5] simulates the different EDG middleware components involved in the job scheduling process, such as the Grid Information System (GIS) that provides information on the available Computing Elements and their characteristics, the DGAS Price Authorities (PAs) that set the resource prices and the Resource Broker (RB)[†] that matches the requirements of a given job to the available Computing Elements and selects the most appropriate resource based on the information provided from the GIS and the single PAs.

The computing power furnished by a Computing Element (CE) — that in DGAS-Sim for simplicity represents a single-processor grid resource with an FCFS (first-come-first-serve) queue — is measured in *Units of Computational Energy (UCE)* per time unit, where we define computational energy as the product of a performance factor p and resource usage u (e.g. the product of a benchmark for CPU performance and the CPU time) in analogy to physics [4]. The computational energy required by a job should ideally be independent of the resource that is executing it. [‡] Different job types (see Section "Simulation configuration") and the time interval between job submissions are among the simulation parameters.

Each simulated PA uses a specific Pricing Module. The simulations presented in [5] and in this paper consider different pricing configurations (in terms of base prices and price variation limits) of a module that implements the *Hybrid Pricing Model (HPM)* [4, 5] described in the following Section.

At present, only processing power is being priced. Although our simulations do not consider data location, network traffic and other factors that influence workload balancing on geographically

[†] The EDG architecture supports an arbitrary number of distributed RBs in order to improve scalability. For simulation, however, a single RB is sufficient.

[‡] The resource usage expectations of a job might be provided by the user's job description or eventually by predictions based on historical information, as for example described in [8] and [2]. For simplicity, however, DGAS-Sim assumes the exact consumption of computational energy to be known.

distributed grids, the results can help to predict the behavior in more realistic grid settings. Furthermore, the EDG Resource Broker (RB) already considers data location during the matchmaking phase — before price information would be included into the resource selection process — by classifying all Computing Elements (CEs), that match a given job request, according to the number of input files that reside on Storage Elements (SEs) "close" to the CE (see [3], Annex 7.7). Moreover, scheduling computing tasks to sites on which the input data is already present while actively replicating popular datasets "decouples" computation scheduling and data movement (in this case replica management) [7]. We therefore believe that a price-sensitive brokering strategy (based only on processing power) may be applied for selecting a resource among a set of CEs that have been matched to a given job by a "data location-aware" RB, as is the case for EDG.

The Hybrid Pricing Model

In the *Hybrid Pricing Model (HPM)*, proposed in [4], PAs *dynamically* adjust prices within *static* limits to balance the workload on the basis of the queue wait times (QWT), that in DGAS-Sim represent the queue lengths. The computation of a resource's price requires only local information about the state of the given resource. Prices are expressed in Grid Credits/UCE and determined as follows:

$$price = P_0 + \Delta P \frac{W - \frac{1}{2}W_{max}}{\frac{1}{2}W_{max}} \qquad 0 \leq W \leq W_{max}$$

Where W is the current QWT and W_{max} the queue's characteristic maximum QWT. P_0 is a fixed base price for the specific queue and ΔP defines the limits of its price domain. Since price adjustments are proportional to the variations in queue length, only results concerning QWTs will be presented.

The HPM offers the advantage of relative price stability, since the fixed price interval ensures that prices do not diverge in the long term. Further advantages are the minimal implementation complexity and the low computational overhead required for price adjustments. Moreover, the fact that the HPM requires only local information significantly limits the communication overhead.

Simulation configuration

The results presented in this paper have been obtained using the same simulation configuration as in [5]. The following briefly summarizes the configuration that models a small grid environment with a total of 50 CEs each of which is associated with one of 5 *queue types* (10 CEs for each queue type) that are characterized by a maximum Queue Wait Time (QWT), expressed in time units (1 t.u. = 1 minute) and ranging from 120 min. (for "*very short queues*") to 1440 min. (for "*very long queues*").

The CEs of each queue type furnished a computing power between 800 UCE/min and 2400 UCE/min, where for simulation purpose we assume the CPU clock speed to be a sufficiently good measure of CPU performance and define the UCE as 1 MHz $\times$ 1 time unit = 1 MHz$\times$min. A total computational power of 80000 UCE per minute was furnished by the 50 CEs.

For each queue type there were three corresponding PA types (see Table I.b): "cheap", "medium" and "expensive" that reflected approximately the CPU performance. The prices were adjusted only every 10 minutes and only if the price difference was greater than or equal to a threshold of 2 Grid

Table I. Job types and computational energy consumed (a) and PA base prices for simulation runs 1 and 3 (b).

(a) job type	comp.en. [UCE]	min.CPU [GHz]	queue types
0	2840–4260	–	very short, short
1	2840–4260	1.6	very short, short
2	9230–12070	–	very short, short
3	9230–12070	–	short, medium
4	9230–12070	1.2	short, medium
5	9230–12070	2.0	short, medium
6	9230–12070	–	short, med., long
7	17750–24850	–	short, medium
8/9	17750–24850	– / –	medium, long
10/11	17750–24850	1.2 / 2.0	medium, long
12/13	39050–46150	– / 1.6	medium, long
14/15/16	39050–46150	– / 1.2 / 2.0	long, very long
17	99400–113600	–	long, very long
18	106500–113600	1.6	long, very long
19	198800–227200	–	long, very long

(b) queue type	P_0 [Grid Credits]	
very short	1175	(cheap)
very short	1200	(medium)
very short	1225	(expensive)
short	1075	(cheap)
short	1100	(medium)
short	1125	(expensive)
medium	975	(cheap)
medium	1000	(medium)
medium	1025	(expensive)
long	875	(cheap)
long	900	(medium)
long	925	(expensive)
very long	775	(cheap)
very long	800	(medium)
very long	825	(expensive)

Credits/UCE. Since in the HPM price adjustments are proportional to the variations in queue length, the QWTs shown in Figures 1 and 2 were updated only when prices were adjusted.

The Resource Broker simply submitted jobs to the cheapest CE that matched the job requirements. Matchmaking was simulated by restricting job submissions to specific queue classes and requiring a minimum CPU performance. Approximately every 30 seconds one of the job types listed in Table I.a was randomly chosen for submission. Since all job types were equiprobable, some of them were similar in order to characterize a particular job distribution. Having different levels of demand for different queue types allows to study the load balancing through price-sensitive brokering with the HPM in a differentiated market situation (due to the imposed queue type requirements we could expect an *over-demand* for the longer queue types — where the computing power requested by the jobs exceeds the computing power furnished by the CEs — and an *under-demand* for the shorter queue types — where the computing power requested by the jobs is lower than the computing power furnished by the CEs).

Previous simulation results

In [5] we discussed two simulation runs that were executed with arbitrarily chosen initial QWTs for the single CEs in order to verify the capability of the HPM approach to balance initially unbalanced queues. Run 1 was done assigning different base prices P_0 to the single PAs (see Table I.b) in order to reflect the different "qualities" of the Computing Elements (in terms of processing power and maximum QWT). The variation limit ΔP was 50 Grid Credits/UCE for all PAs. Run 2 was done with all PAs having a common base price P_0 of 1000 Grid Credits/UCE and ΔP being 50 Grid Credits/UCE.

Figures 1a, 1b, 2a and 2b show, as examples, the results of both simulation runs for the CEs with *medium queues* (max. QWT of 600 min) and *very long queues* (max. QWT of 1440 min); for the other

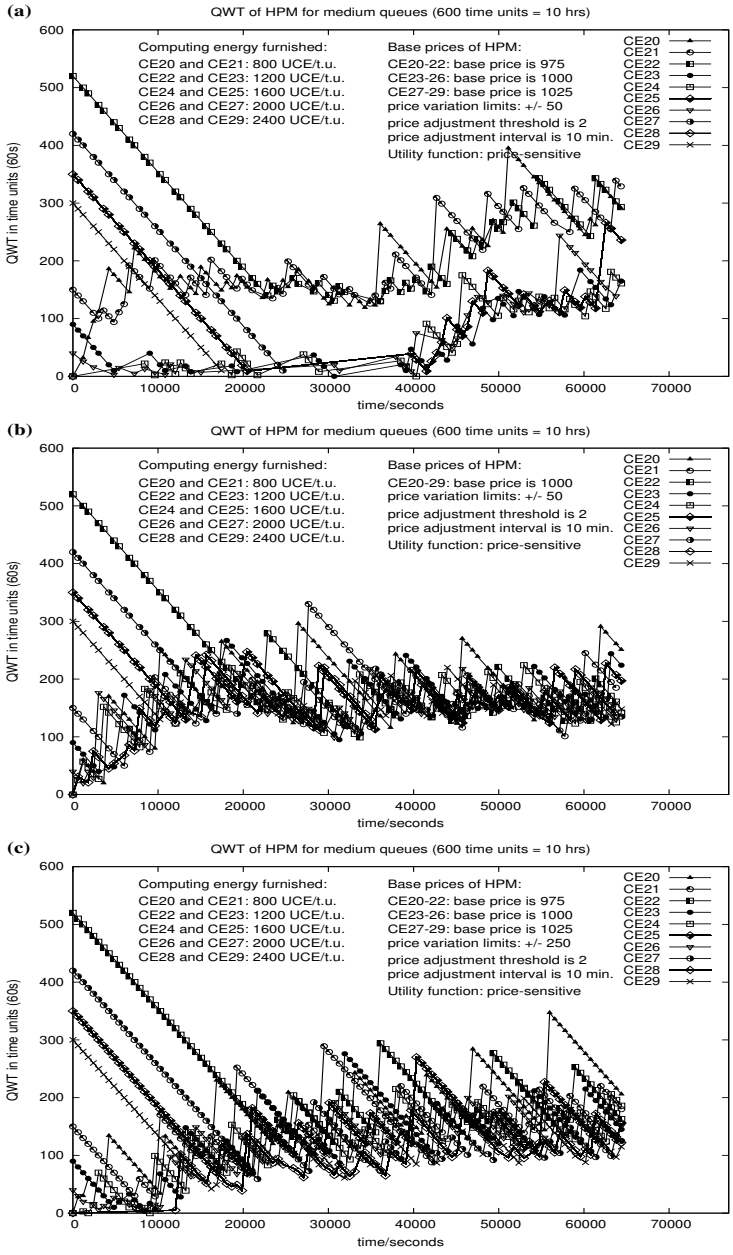


Figure 1. QWTs of *medium queues* of simulation run 1 (a), run 2 (b) and run 3 (c).

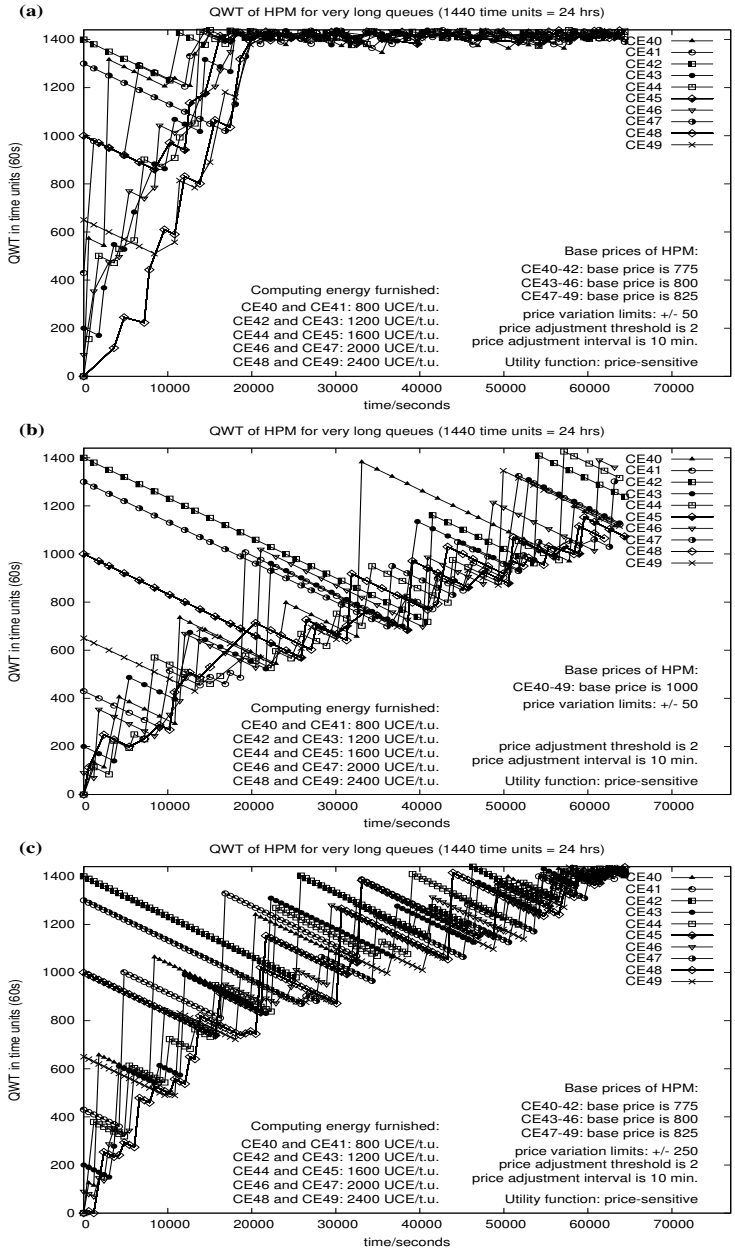


Figure 2. QWTs of *very long queues* of simulation run 1 (a), run 2 (b) and run 3 (c).

queue types see [5] and [6]. Although the average computational energy requested per minute in run 1 was only about 98.7% of the 80,000 UCE per minute furnished by all 50 CEs, a high request denial rate of 4.3% was measured due to a fast saturation of the longer (usually cheaper) queues and the fact that some jobs had requirements (e.g. queue types) that consequently could not always be satisfied without extending the QWT of the matching CEs beyond their maximum values. In run 2, instead, all job submission requests could be satisfied (no request denials) since longer queues were not necessarily cheaper and thus did not saturate within the given simulation time.

For comparison two simulation runs with the same configuration but non-pricing based resource brokering strategies have been executed: one using a *Random* utility function as "worst case" — each job is submitted to a randomly chosen matching resource — and one using a *Minimum Completion Time* utility function as "best case" — each job is submitted to the matching resource that offers the lowest completion time (queue wait time plus job execution time). As examples, the QWTs for medium and very long queues resulting from these two brokering strategies are shown in Fig. 3 and 4.

We concluded that although it seems fair to price computing resources according to their "quality" parameters (e.g. processing power and maximum queue lengths), this might lead to a suboptimal resource selection, at least if a purely price-sensitive brokering strategy is used and the price domains of the different queue classes do not overlap sufficiently as is the case for simulation run 1 [5]. The adoption of a single base price for all CEs (simulation run 2) shows better results that approximate the best case (completion time-based brokering). The medium queues (see Fig. 1b) were more or less balanced and showed a constant mean QWT, since demand and supply for these queues were comparable. Although the longer queues had increasing mean QWTs (see Fig. 2b) due to the expected over-demand, they still became more or less balanced, apart from the high fluctuations for low performant CEs. These can be explained by basically two effects: First, jobs take longer to execute on CEs with low performance, increasing their QWTs more than the QWTs of high performant CEs. Second, the price adjustment interval of 10 minutes implied that the currently cheapest resources were chosen for the next 10 minutes to satisfy the incoming job requests, leading thus to a much higher QWT on the next price computation. Hence the amplitudes of these fluctuations depend in a significant way on the time interval between price adjustments. A shorter interval will lead to smaller fluctuations of the single QWTs, but also to a higher computational overhead.

Applying a set of different prefixed base prices with non-overlapping or only slightly overlapping price domains (relatively small ΔP), as in sim. run 1, may lead to "subgrids" with eventually different levels of workload, as can be seen for example in Fig. 1a, where the low performant and thus cheaper CEs (CE20-22) received most of the jobs that were submitted to medium queues and no job was submitted to the high performant CEs (CE27-29). Although this might also be a desirable grid setting, where users pay more for empty or slightly loaded queues if their applications require an immediate processing (e.g. for interactive jobs), it may result in less effective load balancing and higher request denial rates since generally cheap resources become overloaded more rapidly (see Fig. 2a).

A more detailed analysis of the two simulation runs can be found in [5].

Different base prices and widely overlapping price domains

To consider a grid economy scenario with base prices differentiated according to the resource characteristics but *widely overlapping* price domains we executed a further simulation run (run 3),

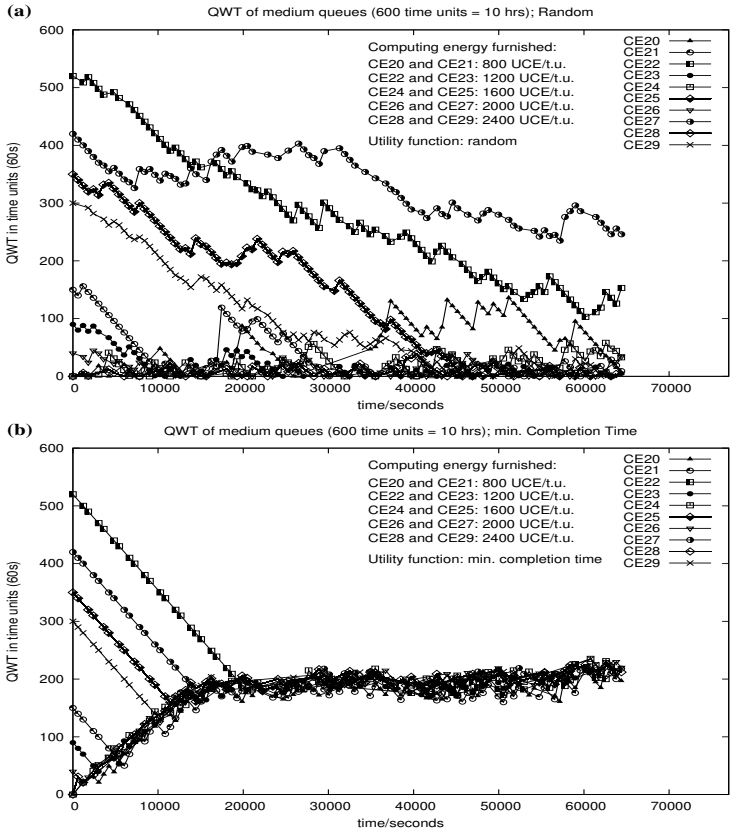


Figure 3. QWTs of *medium queues* for utility functions Random (a) and Minimum Completion Time (b).

having the same configuration (including the base prices) as run 1, except for a price variation limit ΔP of 250 Grid Credits instead of only 50 Grid Credits. Hence, the price domains of the different queue classes overlapped significantly.[§]

Figure 1c shows the QWTs of the *medium queues*. As in run 2 (Fig. 1b) the demand and supply for computing resources with medium queues was more or less balanced, apart from the high fluctuations

[§] In other words, the possibility that queues of distinct classes had similar prices was much higher. In run 3 even the price domains of the cheapest resources (775±250 Grid Credits) and the most expensive ones (1225±250 Grid Credits) overlapped, while in run 1 only the price domains of adjacent queue classes overlapped (e.g. very short and short queues or short and medium queues).

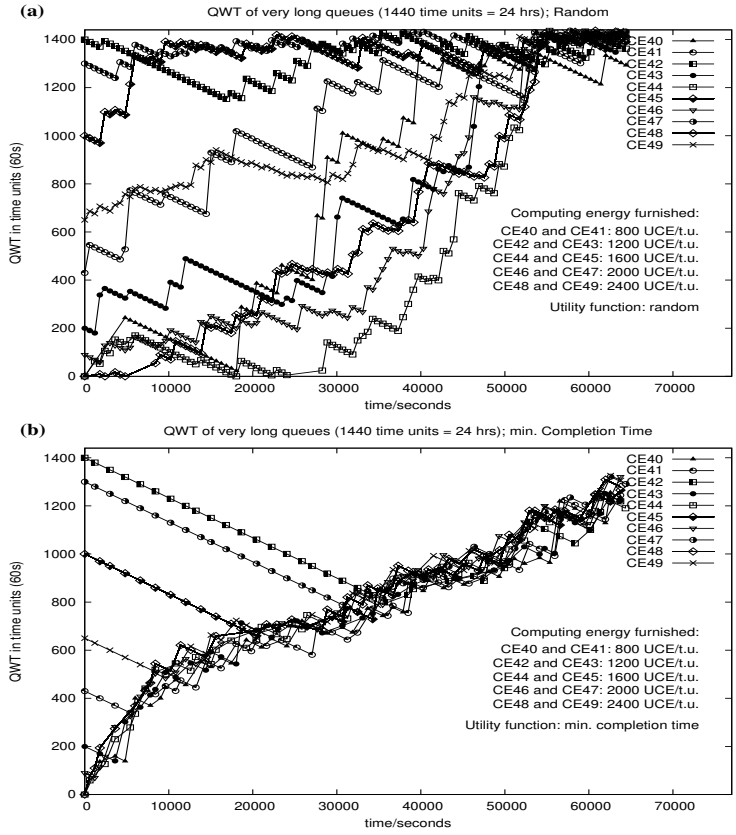


Figure 4. QWTs of *very long queues* for utility functions Random (a) and Minimum Completion Time (b).

for low performant CEs (see previous section). In contrast to run 1 (Fig. 1a), all medium queues received job submissions since the differences of the base prices (up to 50 Grid Credits, see Table I b) were relatively small compared to the high price variation limit ΔP.

The results for *very long queues* for run 3 (see Fig. 2c) are comparable to the results obtained for run 2 (Fig. 2b), although due to the slightly lower base prices the over-demand for this queue class was higher and thus the mean QWT increased faster. Since the price domains of CEs with very long queues, however, significantly overlapped with those of CEs with other queue types, the incoming jobs were better distributed without being submitted mainly to CEs with very long queues. Thus the mean QWT increased much slower than for run 1 and the resulting request denial rate was below 0.05%.

The other queue classes, as well, show results similar to those of the simulation run without differentiated base prices (run 2), with the exception of the *very short queues* that received no single

job submission as for simulation run 1. In both run 1 and run 3 the very short queues were always the most expensive ones, since due to the expected under-demand the short queues never received enough job submissions to become more expensive than the very short queues.

The adoption of a common base price (run 2) and the adoption of differentiated base prices with sufficiently overlapping price domains (run 3), are both able to approximate the best case (completion time-based brokering, see Fig. 3b and 4b), while having the advantage of an economic approach by providing an incentive for grid users to delay less urgent job submissions to periods with lower congestion (and thus lower prices).

Relative Standard Deviation of QWTs

The efficacy to balance the workload, i.e. the QWTs, within the different queue classes can be represented by the *relative standard deviation (RSD)*, that is, the ratio of the standard deviation of the QWTs to their mean value.[¶]

Figures 5 and 6 show the relative standard deviations for the medium, long and very long queues of all simulation runs. The relative standard deviations for very short queues are not shown since they drop to zero[‖] as soon as the queues are completely emptied (very short queues for run 1 and 3). The relative standard deviations for short queues (and very short queues for run 2) show very high and instable values since the small number of job submissions (due to the high under-demand) causes relatively high fluctuations of the QWTs compared to their very low mean value.[**] The under-demand case is thus less appropriate for comparing the general efficiency of different HPM configurations for the purpose of load balancing.

The initial RSD is quite high for all queue types and simulation runs, given by the fact that initial QWTs are chosen arbitrarily in order to verify the capability of the HPM to effectively balance initially unbalanced workloads.

As can be seen in Fig. 5a the RSD of *medium queues* is highest for simulation run 1 (differentiated base prices P_0 and low price variation limit ΔP), since the low performant and thus cheaper CEs (CE20-22) received most of the jobs that were submitted to medium queues, and no job was submitted to the high performant CEs (CE27-29), as can be seen in Fig. 1a. The medium queues become balanced best in run 2 (equal base prices, see Fig. 5b and compare to the best case in Fig. 6b), but the results of run 3 (see Fig. 5c) show a similar performance for widely overlapping price domains.

The performance for *long* and *very long queues* seems to be best for run 1 (see Fig. 5a), since the RSD drops to nearly zero after about 35000 seconds (9.7 hrs) and after about 20000 seconds (5.6 hrs) respectively. This however is not the result of a balanced distribution of the workload, but simply depends on the fact that these queues reach their maximum QWT (see for example Fig. 2a), and thus the variation in their QWTs becomes negligible. The same holds for very long queues after

[¶] The relative standard deviation was used instead of the standard deviation in order to be able to compare the behavior for the different queue types.

[‖] For the case in which all queues are empty (e.g. very short queues) the RSD is set to zero (zero standard deviation divided by zero mean QWT).

[**] The relative standard deviations for very short and short queues can be found in [6].

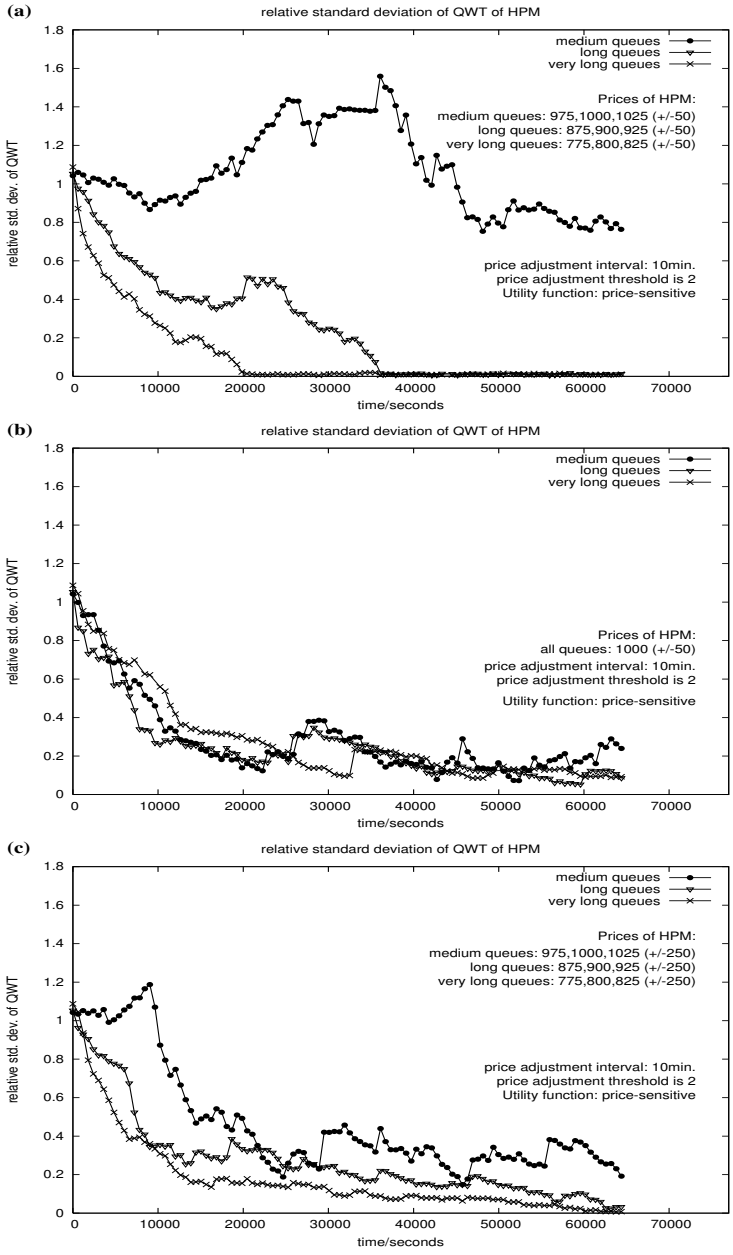


Figure 5. Standard deviation of QWTs relative to the mean value for medium, long and very long queues of sim. run 1 (a), sim. run 2 (b) and sim. run 3 (c).

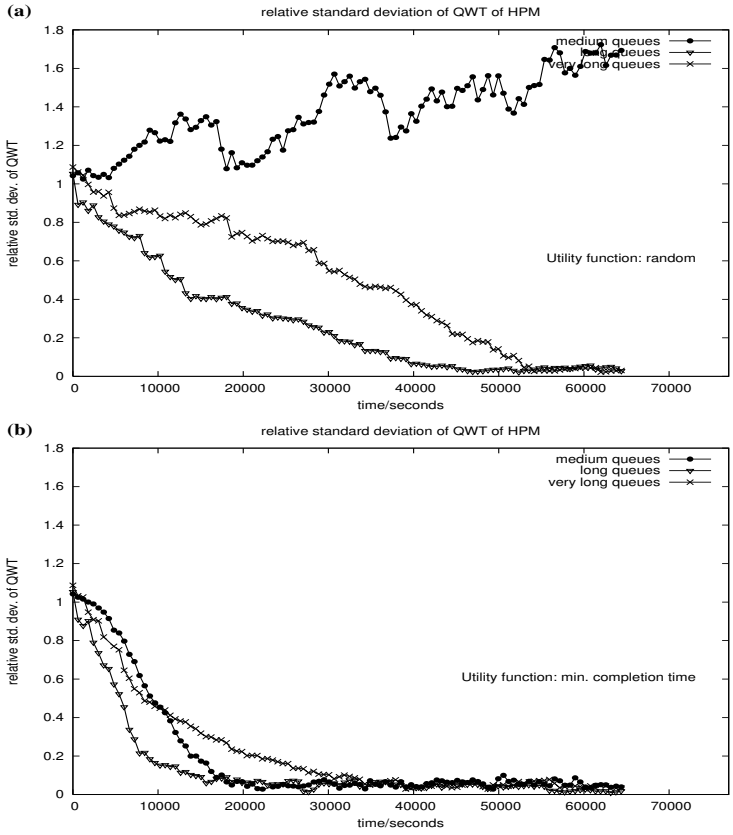


Figure 6. Standard deviation of QWTs relative to the mean value for medium, long and very long queues of Random utility function (a) and Completion Time utility function (b).

about 60000 seconds (16.7 hrs) in run 3 (see Fig. 5c and 2c). Apart from that, the results for run 2 and 3 are comparable. Both runs show results that are closer†† to the best case (completion time-based brokering, see Fig. 6b) than to the worst case (random brokering, see Fig. 6a), while the relative standard deviations for the case with differentiated base prices with not sufficiently overlapping price domains (run 1, Fig. 5a) are comparable to the worst case (Fig. 6a).

††Due to the high fluctuations explained in one of the previous sections, however, the RSD is higher and less stable than for completion time-based brokering.

Conclusion

For a purely price-sensitive brokering strategy a single base price, independent of CPU performance and maximum queue length, is the best choice. However, for sufficiently large price variation limits ΔP, that guarantee widely overlapping price domains with respect to the base prices, similarly effective load balancing may be obtained even if base prices differ to reflect the resources' different "qualities" (in terms of CPU performance, maximum queue length, etc.).

The adoption of a single prefixed base price would have to be agreed upon by all participating Virtual Organizations (VOs) not withstanding their freedom to define their own policies, one of the principles of grid computing. An approach with differentiated base prices but widely overlapping price domains may be the preferable choice, since it is more flexible while offering a comparable performance.

If applying a single base price for all CEs even a relatively small price variation limit ΔP will be sufficient to effectively balance the workload [6], but this will significantly reduce the advantages of an economic approach: If price variations are insignificant, the HPM can balance the workload only in the short term (by "just-in-time" scheduling of the incoming workload), but it will not offer sufficient incentives for users to delay job submissions in times of high congestion on the system (in order to reach a market equilibrium), thus failing to balance the workload on a longer time scale.

The local scheduling policies of grid sites may be more sophisticated than FCFS and may lead to different job start times for jobs from different VOs or users [2]. This might require to base the pricing algorithm on job start times instead of the QWT in order to reflect the real state and queue status of a Computing Element. Basing prices on job start times, however, is conceptually different since a price would depend not only on the resource but also on the job to be submitted. Since the results presented for the HPM are limited to a grid setting with FCFS queues, further investigation is necessary to evaluate whether a similar model may be effectively adopted to grid settings that require to consider job start times instead of queue wait times.

Additionally, future work has to verify the stability of the results over a wider range of parameters.

REFERENCES

1. European DataGrid (EDG) project website. `http://www.eu-datagrid.org/`
2. Li H et al. Predicting job start times on clusters. *4th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid2004)*, Chicago, USA, April 19-22, 2004.
3. Pacini F. WP1 - WMS software administrator and user guide. Technical Report, DataGrid-01-TED-0118-1_2, Datamat S.p.A., Rome, Italy, November 2003.
4. Piro RM, Guarise A, Werbrouck A. An economy-based accounting infrastructure for the DataGrid. *Proc. of the 4th International Workshop on Grid Computing (Grid2003)*, Phoenix, Arizona, USA, November 17th, 2003.
5. Piro RM, Guarise A, Werbrouck A. Simulation of price-sensitive resource brokering and the Hybrid Pricing Model with DGAS-Sim. *Proc. of the 13th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2004)*, track on Emerging Technologies for Next Generation Grid (ETNGRID 2004), Modena, Italy, June 14-16, 2004.
6. Piro RM. *Simulation of Economy-based Load Balancing in Computational Grids for Large-scale Scientific Applications*. Laurea Specialistica thesis, University of Turin, Italy, April 2004.
7. Ranganathan K, Foster I. Simulation studies of computation and data scheduling algorithms for data grids. *Journal of Grid Computing* 2003; **1**: 53-62.
8. Smith W, Taylor V, Foster I. Using run-time predictions to estimate queue wait times and improve scheduler performance. *Proc. of the IPPS/SPDP'99 Workshop JSSPP*, Puerto Rico, USA, 1999.
9. EDG Workload Management System website. `http://www.infn.it/workload-grid/`