INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Flexible Management and Dynamic Control of Network Quality of Service in Grid environments: the QoSinus approach

Pascale Vicat-Blanc Primet[2] — Johan Montagnat[1] — Fabien Chanussot[2] — Mathieu Goutelle[2] —

[1] CNRS UMR5515, CREATIS, INSA de Lyon, 69621 Villeurbanne, France
[2] INRIA-ENS LIP, RESO, 46, allée d'Italie, 69 007 Lyon, France

N° 5083

January 2004

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--5083--FR+ENG

Flexible Management and Dynamic Control of Network Quality of Service in Grid environments: the QoSinus approach*

Pascale Vicat-Blanc Primet[2] , Johan Montagnat[1] , Fabien Chanussot[2] , Mathieu Goutelle[2] ,

[1] CNRS UMR5515, CREATIS, INSA de Lyon, 69621 Villeurbanne, France
[2] INRIA-ENS LIP, RESO, 46, allée d'Italie, 69 007 Lyon, France

Thème 1 — Réseaux et systèmes
Projet RESO

Rapport de recherche n° 5083 — January 2004 — 20 pages

Abstract: Grids generally rely on a complex interconnection of IP domains that may exhibit changing performance characteristics and may offer different quality of service facilities. However the end to end performances the heterogeneous mix of grid flows get from the network may affect the overall grid infrastructure performances, utilization level and impact each individual application's performance. In this paper we examine the case of the eToile french grid testbed based on a research network infrastructure that proposes a set of Differentiated Services with various end to end performances. We present the QoSinus service that aim to dynamically allocate the network resources to Grid flows in order to match their specific requirements under different load conditions. The QoSinus service is based on the *programmable network* approach. We show how a biomedical application distributed over the grid may benefit from such an advanced and controlled network service.

Key-words: Grid network service, Grid QoS, DiffServ, programmable network service, Medical Application

* This work was funded by the RNTL eToile project of the French ministry of research, by the European Commission program IST-2000-25182 through the EU DataGrid Project, the INRIA project RESO, and the french ministry for research ACI-GRID project.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 Montbonnot-St-Martin (France)
Téléphone : +33 4 76 61 52 00 —Télécopie +33 4 76 61 52 52

Gestion flexible et controle dynamique de la qualité de service réseau dans un environement grille: l'approche QoSINUS

Résumé : Les grilles sont généralement construites sur des interconnexions complexes de domaines IP, qui peuvent avoir des performances différentes et proposer différentes fonctionnalités de qualité de service. Cependant, les performances de bout en bout que le réseau assure aux différents types de flux dans une grille affectent les performances globales de la grille, son niveau d'utilisation, et les performances de chacune des ses applications. Ce rapport étudie le cas de la plateforme de grille francaise eToile. Elle est basée sur une infrastructure de réseau expérimentale qui propose un ensemble de services DiffServ aux performances de bout en bout variées. Le service QoSINUS est présenté. Celui-ci cherche à allouer dynamiquement les ressources réseau aux flux de la grille de manière à assurer leurs besoins dans les différentes conditions de charge du réseau. Le service QoSINUS est basé sur l'approche *réseau programmable*. Les bénéfices d'un tel service réseau avancé et controllé pour une application biomédicale distribuée sur une grille sont également démontrés.

Mots-clés : Service réseau pour la grille, QoS Grille, DiffServ, service réseau programmable, application médicale

1 Introduction

The purpose of Computational Grids is to aggregate a large collection of shared resources (computing, communication, storage, information) to build an efficient and very high performance computing environment for data-intensive or computing-intensive applications [6]. But the underlying communication infrastructure of these large scale distributed environments is a complex interconnection of multi IP domains with changing performance characteristics and different quality of service facilities. Consequently *the Grid Network cloud* may exhibit extreme heterogeneity in performance and reliability that can considerably affect the global application performances.

There are many possible utilizations of a grid, not only intensive computing but also interactive usage with remote access to expensive instruments or huge scientific data bases exploration for instance. Consequently, many data transfer patterns will occur in such an environment. Bulk data transfer from one location to some other locations and time constrained message exchanges of fine-grained parallel applications may have to coexist. In Grids, the quality of service (QoS) the heterogeneous mix of individual flows receive may affect the overall grid infrastructure performances, utilization level and impact each individual application performance.

An important effort to provide QoS architecture in the IP networks has been done during the last decade. The Internet community has defined QoS services architectures that scale, while enabling a wide range of QoS guarantees. One contradiction of today's retained IP QoS approaches like DiffServ [1] is they are *pure in network* solutions that care on packets (level 3 building block) and aggregates while end to end users pay attention to the performances of individual flows (level 4 abstract unit). Users have no way to specify in advance and control the service they expect at flow level. To enable Grid users and applications to access the QoS capabilities of the IP networks we argue that a service is required that abstracts and interfaces the QoS objective of the flows and the packet level QoS services offered by underlying networks. In this paper, we present the QoSinus service that has been designed to manage the DiffServ classes at the edge of the network and to allocate them to Grid flows in order to match the specific requirements of individual grid applications. The characteristics and the behaviour of the QoS capabilities of the French grid testbed eToile are presented. The performance requirements of a Grid biomedical application and the results obtained over the eToile network are examined.

2 QoS solutions for Grids

2.1 Requirements

Several analysis [2, 12] have shown that the QoS requirements spectrum of Grid flows is broad comprising needs like the guaranteed delivery of a complete huge data file, the predictability of the TCP throughput, the bounded delivery time of a file transfer, a stability-level for data-delivery.

The network performance model we consider, in the rest of this paper, is a set of quantitative QoS characteristics. These characteristics are bit rate, delay, jitter or delay variation, loss rate and error rate.

Distributed applications generally rely on TCP protocol running over IP for wide area communications. IP is a connectionless protocol and therefore it has no inherent mechanisms to deliver guarantees according to traffic contracts. In the IP routers along a path from sender to destination, the rate at which aggregated traffic is directed to an interface may exceed the rate at which the interface can forward the traffic onward. First, the exceeded traffic will be queued in the device's memory until the congestion subsides, then the traffic will be discarded to alleviate congestion. Consequently, end to end flows will experience varying latency (queues on interfaces) or traffic loss that can damage dramatically the end to end throughput. The increasing interest for Grid Computing make the needs for performance control and optimisation in IP networks more and more vital and challenging. The DiffServ approach is incrementally deployed over IP infrastructures. The Grid network cloud is a complex aggregation of heterogeneous domains offering various performances guarantees and QoS strategies that Grid designer cannot control. Two services are generally provided: Expedited forwarding for time sensitive traffic and Assured Forwarding for rate sensitive traffic.

In the next sections, an example of an high performance Grid network cloud is presented.

2.2 The eToile VTHD testbed and the network performances study

The e-Toile project [13] is an experimental wide area grid testbed. The e-toile [1] has three complementary objectives:

- to build an experimental high performance grid plate-form that scales to France.
- to develop original Grid services to fully exploit the services and capacities offered by a very high performance network. The e-toile middleware integrates the most recent and relevant works of the french computer science laboratories (INRIA, CNRS) focused on enhanced communication services.
- to evaluate the deployment cost of choosed computing intensive and data-intensive applications and to estimate the performance gain they may obtain over the grid.

The VTHD (vraiment très haut débit) network infrastructure interconnects the grid nodes with access link of 1 to 2 Gbps. The e-toile middleware relies on existing middleware (Globus [5], Grid Engine...) and integrates new building blocks. The e-Toile middleware aims to fully exploit the power of the networking infrastructure. One of the original aim of e-Toile is to focus on the High Performance Grid Networking dimension and to evaluate the benefit that grid middlewares and applications can get from enhanced networking technologies.

The performance problem of the grid network are studied in e-Toile from different but complementary points of view:

[1] e-toile is a RNTL project (réseau national de recherche en logiciel) funded by French Ministry of Research

- Measuring and monitoring the end to end performances helps to characterize the links and the network behaviour. Network cost functions and forecasts, based on such measurement information, allow the upper abstraction level to build optimisation and adaptation algorithms.
- Evaluating the advantages of differentiated services, like Premium or Less than Assured Services, offered by the network infrastructure for specific grid flows.
- Creating enhanced and programmable transport protocols to optimise heterogeneous data transfers within the grid.

A Grid based on a controlled very high speed network such as e-Toile permits to study the limits of the existing communication services and protocols and to validate more efficient approaches that aim to carry the gigabit performance to the grid user level and take into consideration the specific needs of grid flows. In this paper we concentrates on the differentiated services evaluation and on the estimation of the performance gain they can offer to Grid application flows.

The Gigabit VTHD [14] Backbone provides four DiffServ classes (EF, TCP AF, UDP AF and BE). The principle of the DiffServ configuration in VTHD is that a bandwidth share is statically provisioned in the edge routers and allocated to the four DS classes. Each access point has to control and to shape the traffic injected in each class. Table 3 gives the absolute rate allocated to each DiffServ class.

A first experimentation has been done, to evaluate the raw performances of the DiffServ classes obtained from one edge to another. Best Effort, EF and AF TCP performances have been measured in five network loads conditions: no background traffic, 267Mbits/s, 535Mbits/s, 840Mbits/s and 1000Mbits/s UDP background traffic. Figure 1, 2 and 3 show the performance (TCP Throuput and RTT) of Best Effort, EF and AF resspectively.

This simple experiment shows that the EF trafic is correctly protected.

The Best Effort performances are very affected when the link is congested. The TCP throughput is divided by 10. This results corresponds to the TCP/IP behaviour. AF performances are protected from Best Effort traffic, but impacted by EF traffic, as expected. A service that permits to control and allocate the DiffServ classes to grid flows is exposed in the next section.

3 QoSinus objective and design principles

One contradiction of today's retained IP QoS approaches like DiffServ is they are *pure in network* solution that care on packets (layer 3 building block) and flow aggregates while end to end users pay attention to the performances of individual flows (layer 4 abstract unit). User have no way to specify in advance and control the service level they expect. In DiffServ standard, packet marking is done by the ingress router. We argue that one reason for the slow deployment of the DiffServ technology is due to the absence of end to end individual flow's performances guaranties. The approach we propose aims to enable Grid users and

applications, that may strongly benefit from QoS guaranties, to access simply to the various QoS capabilities offered by IP domains. This approach combines application aware and infrastructure aware components activated within the network, just at the interface of the Grid computing domain and the Grid long distance networks. A Grid oriented QoS API and a programmable QoS service has been designed and developed within the context of the e-Toile project to introduce flexibility and dynamic in the management, the control and the realization of end to end QoS in Grid context. Such an approach increases slightly the complexity at the frontier points, but leaves the core network and the grid applications simple. The aims are to:

1. provide an extensible API to allow heterogeneous Grid flows to specify their QoS objectives,
2. map these objectives with the existing IP QoS services provided at the edge of the core inter-networks for improving the individual packet performances,
3. realize a dynamic and appropriate adaptation according to the real state of the link, the QoS mechanisms configured and the experienced performances.

The first issue to solve is to allow grid applications to specify and control their QoS. This issue is addressed by an API that provides the user the ability to characterize the flow needs in terms of qualitative or quantitative end-to-end delay, end to end throughput, end-to-end loss rate or in terms of relative weight of these three main metrics. This API permits to define SLS (end to end service level specifications) in XML.

The second issue is addressed by a service architecture that combines flow aware and infrastructure aware components to map and dynamically adapt the QoS specification of the flows to the QoS facilities offered by the network.

We have identified four types of component for flexible QoS programming and control:

- programming component,
- performance measurement component,
- adaptive control component,
- enforcement component.

The QoS programming component is invoked for initiating, propagating and storing the flow QoS goals in the programmable nodes of the path. The QoS performance measurement is responsible for the characterisation of the specific paths. In a DiffServ context, this component can measure the performances of each DS class on each grid path with active out of band probes or filter and gather flow performances in band. The adaptive control components is responsible for class mapping and for adapting the packet marking influencing the forwarding performances regarding the QoS goals and the current state of network classes. It monitors and updates the allocated bandwidth of each class and acts as a decision component that decides which class has to be attributed to the packets when the performance

measurement component indicates some change. The QoS enforcement service, intercepts and adapts the data flows when it is necessary. It realizes the packet marking and conditioning functions. Other components, that are not designed in this architecture, may be added for status report exchange between programmable nodes for example. Various tables and soft states are associated to this service: Class allocation table that stores the status of the allocated class bandwidth, *QoS goal* structure that maintains the flows objectives and *QOS flow status* that monitors the flow. The number of states managed in each programmable node is limited to the number of active flows going out the local grid site and depends on the size of the cluster. In the worst case, it will be in the order of a few hundreds.

The infrastructure elements of this model are the performance measurement component and the class allocation table that indicate the state of the programmable nodes and the network performances. The flow aware components are the programming component and the enforcement component. The adaptive control component is flexible, extensible, replaceable and it is simultaneously dependant on the infrastructure QoS properties and on the flow properties. It represents the kernel of the service. It decides the adaptation according to the performance experienced by the packets of each particular flow. With DiffServ, this adaptation means dynamic rate shaping or packet remarking.

The QoSinus API and programmable service [10] implements this approach in a DiffServ context and interfaces the application QoS specification with an adaptive packet marking strategy at the Grid domain frontier. The QoSinus class diagram is presented in figure 4.

>From the user point of view the QoSinus service is invoked in two phases. The first one is called the programming phase: the Service Level Specification is transmitted to the QoSinus service. A session code is returned back to the sender. Then the sender transmits data packets, identified by the attributed session code. The QoSinus decision component choose dynamically the appropriated DiffServ class to cross the DS domain, trying to optimize the class usage and the flow performance.

In the e-Toile context, the QoSinus decision component is responsible for the flexible mapping of end to end high level QoS needs of grid flows to the actual QoS provided by the network cloud. When a DiffServ class is allocated by the decision component, the corresponding available bandwidth parameter is updated. The flow QoS enforcement component of QoSinus implements token buckets to insure that the flows remain in profile. Different mappings and adaptive algorithms have been designed. The simplest one, is not adaptive and not optimised. It performs the allocation decision on static thresholds. For example, if a delay constrains is expressed and EF bandwidth is still available and sufficient, an EF bandwidth corresponding to the required ones will be allocated. All packets belonging to this flows will be marked EF. The adaptive algorithms attempt to fulfil the requirements of the flows while optimising the classes utilization. The classes are ordered by level of performance they offer. The adaptive mapper monitors the performances of the flows and dynamically allocate class resources according to some kind of earliest deadline first algorithm.

4 Experiments considering real usecases

In this section we introduce a *content-based query of medical image databases* application with strong QoS requirements. Medical imaging is data intensive due to the size of medical images. For instance, a CT-scanner usually produces 3D images made of 50 to 100 slices. Each slice is a 512^2 image. Each voxel intensity is encoded on 16 bits. This results in a 25 to 50 Mb image. A typical 3D Magnetic Resonance Image is 256^2 per slice, 128 slices, 2 bytes per voxel resulting in a 16 Mb image. Content-based query of images on a grid requires the distribution of large data set over the network.

4.1 Application description

Digital medical images are stored in databases. The simplest data structures are flat files: one file for each (2D or 3D) image. Meta-data on image files (e.g. patient name, acquisition type, etc) are usually stored in relational tables and used for querying the available data. Beyond simple queries on meta-data, physicians want to search for images close to a sample they are interested in, so that they can study cases similar to the image they are looking at. This involves an analysis of the image contents to determine images from a database that are similar to a sample image.

Several similarity measures may be used to measure the differences between images [8, 11, 9]. Although these computations are not very intensive, the comparison of a sample image against a complete database may become intractable due to the size of medical databases. However, the computation can be easily distributed on a computation grid since the comparisons of the sample image with each image of the database are independent.

Figure 5 illustrates a content-based query on a computation grid. A sample image is available on the physician's computer. A complete database is available from a storage site. The images are distributed over the computation grid to perform similarity measurements. The result returned to the user interface is a score for each image in the database. The physician can then select the highest score images for visualization.

4.2 Performance analysis and Network requirements

The application does not exhibit real-time constraints. However, computations should be fast enough for a reasonable use in clinical practice (fast means minutes in this context, 30 minutes at most). The computation time of the similarity measures is rather low (from few seconds to 15 minutes on a 1GHz Pentium processor, depending on the measures estimated and the input image size). Therefore, the network transmission of the images to the computing sites may become a bottleneck.

Let S the total search time, n the image number, T the transfer time of an image I between two nodes and C the computing time of image I with another image J. Assuming, all computations are done on one processor and the database is located in the same site, the total search time S_1 will be: $S_1 = nC$.

If the Database is located on a remote site and if we assume that the effective rate r (TCP goodput) on links $N_A - N_B$, $N_A - N_C$ and $N_B - N_C$ are stable and equivalent, given site N_a the location of the end user, site N_b the location of the DataBase, site N_c the location of a computing node. The transfer time T of an image is constant and depends on the size l of the image I: $T = \frac{l}{r}$.

If the operations are dispatched on k processors and the database cut in k equal partitions, S_k the total search time will be:

$$S_k = \frac{nC}{k} + \frac{n}{k}T + kT$$

with $\frac{n}{k}T$ being the time to transfer the k partitions to the k processors and kT the time to transfer the image I to the k processors.

We denote SpeedUp the speedup obtained with a search deployed over the grid.

$$\text{SpeedUp} = \frac{S_1}{S_k} = \frac{nkC}{(nC) + (n + k^2)T} \quad (1)$$

Consequently we have:

- If $\frac{C}{T}$ is low, the speedup goes to 0. In this condition, it is better to adopt a centralised approach
- If $\frac{C}{T}$ is high, the speedup goes to k. It is then very interesting to split the database and distribute the computing load.
- If $\frac{C}{T} = 1$, the speedup goes to$\frac{nk}{2n+k^2}$, depending on the ratio$\frac{n}{k}$

If T is not constant, going from lower that C to higher, what may be the case on a classical TCP/IP best effort congested link, performances will be difficult to predict. The appropriate choice of k becomes very complex and the global application performance impossible to determine. We conclude that, on a grid, it is very important to have an accurate evaluation of the data transmission and a stable throughput because it can impact the distribution strategy and the overall performance.

4.3 Expected effect of the Network QoS on the application performance

This application may benefit from different types of QoS services provided by the network: A service that can guarantee the stability of a given TCP throughput will be the best for the job scheduler. The value of the required stable throughput depends on the image size, processing time and database size. We use 3 databases for this application:

- A small database made of 124 2D images, 140 kb each.
- A medium database made of 238 3D images, from 7 Mb to 15 Mb each.

- A large database made of 456 3D images, about 14 Mb each.

The 3 databases studied are made of images as summarized in table 4. Each images grey level range may be undersampled prior to processing. This loss of precision brings an increased computation time. Following experiments are therefore using undersampled version of the original images to 8 and 12 bits when possible.

This application is completely scalable. The network is stressed with transport of data from a few Mb (comparison of few 2D images) to 6.04 GB (complete largest database). Table 5 gives the theoretical transfer time the application will obtain according different typical end to end throughput.

For DB1, an end to end throughput of 100kb/s is not sufficient. For DB2 and DB3, at least 200Mb/s are necessary to satisfy the global 30 minutes constraint. As we have seen earlier, TCP Best Effort bandwidth drops dramatically when concurrent traffic loads the network, and can get as low as 72Mb/s. Multi-streaming or well TCP tuning may help to get better performances and will be required to satisfy the application constraints.

4.4 Results analysis

In this section we evaluate the gain an application can expect within the e-Toile context when using different QoS classes under three types of conditions: empty network (less than 25%), loaded network (between 50% and 75%) and congested network (around 95%). We calculate the gain when using the different QoS services and different distribution strategies.

The medical application involves similarity measure computations over image volumes. The complexity for all similarity measures depends on the size and the dynamic range of medical images. Furthermore, the complexity of computations varies from on measure to another (see [8] for details). Table 6 summarizes the measured computation time (in seconds) for the similarity measures (excluding image I/O and undersampling) on pairs of the 3 above mentionned dabase images, the 6 similarity measures proposed, and every possible undersampling to 8, 12, and 16 bits. The times where measured on a 800MHz Intel pentium III processor.

Table 8 shows network file transfer time measured in the five network load conditions studied in section 2 for Best Effort, EF and AF. This correspond to the measurement of $\frac{n}{k}T$ for DB2 (3227MB with $k = 1$, 807MB with $k = 4$, 323MB with $k = 10$, and 32MB with $k = 100$). This roughly corresponds to the data transfers when distributing the database from the source to one of the k processors ($k \in \{1, 4, 10, 100\}$).

Table 7 shows the time that would be required to transfer a single image of each database in the same network conditions (T). Those results are calculated using the TCP bandwidth measured previously.

According to table 6, the mean appliction execution time (C) is 0.8s (for 8 bits undersampled images), 15.37s (for 12 bits undersampled images), or 734.1s (for 16 bits images). Given that $n = 238$, it is possible to estimate the speedup using equation 1 for various network conditions. Table 1 shows the speedup obtained for congested traffic on an EF class, while table 2 shows the same result on a BestEffort class network. These tables demonstrate

	k		
C (s)	4	10	100
0.8	3.228213	7.603834	9.863241
15.37	3.950837	9.838626	67.766197
734.1	3.999896	9.999657	99.900540

Table 1: Application speedup using $k \in \{4, 10, 100\}$ processors on a congested EF class network.

	k		
C(s)	4	10	100
0.8	1.208275	2.210355	8.386187
15.37	3.570598	8.450010	63.750815
734.1	3.998993	9.996162	99.881129

Table 2: Application speedup using $k \in \{4, 10, 100\}$ processors on a congested BestEffort class network.

that (i) EF class networks significantly improve the speedup for shorter computation times where the network overhead is most damaging the computation time, and (ii) an optimal number of processors can be estimated depending on the application execution time and the network status. Constrained grid application may be very sensitive to network performances and can improve their behaviour by dynamically using DiffServ classes like EF or AF.

4.5 Discussion

Another QoS approach could be to collect information about the achievable throughput and about the stability-level for data-delivery between two points in the network and report this information to the grid users and application. From this information they can directly adapt their use and expectation of the network. Network and throughput measurement services are central to the Grid Network performance measurement architecture that is designed in the European DataGRID project [3] or other Grid projects. A NetCost Estimation Service has also been developed over this monitoring infrastructure to present middleware component a simplified and compact view of the network performance. The aim of such a service is also to simplify the Network performance management the grid application developers will have to care on.

Another key component that helps for end to end QoS management and global performance optimisation in Grid is the dynamic generation of Network performance forecasts. For example, the Network Weather Service (NWS) [15] periodically takes measurements of the load of resources it has to monitor and uses them to generate performance forecasts. One of the NWS sensors is the network sensor [7] whose aim is to take measurements that

represent the network quality in term of latency and bandwidth. The different components of the NWS are distributed on monitored hosts.

The GARA (Globus Architecture for Reservation and Allocation) [4] make a different choice. It provides advance reservations and end-to-end management for quality of service on different types of resources (network, storage and computing). This architecture remains a traditional solution, in the sense that the processing task associated with transport purpose are performed on the end systems (reservation and adaptation).

5 Conclusion

This paper presented the problem of mapping the QoS requirements of grid flows to the QoS services offered by IP network services. The results obtained on an experimental grid testbed show the benefit the Grid application can expect from IP QoS services available in current academicals network infrastructures. A medical image analysis application corresponding to real usecases has been exposed and its theoretical performances analysed. This application has been selected as it is strongly dependent on the network capacity and representative of grid application flow performance requirements. A flexible service named QoSinus that has been designed for dynamically manage the QoS classes of an IP domain to serve the needs of grid application is proposed. The use of this service by grid application can offer a very flexible, transparent and efficient QoS control to grid application. An adaptive marking algorithm is proposed and is very promising. In the future we will compare the performances obtained with the different mapping algorithms under the same utilization and load conditions. The programmable network approach and the modular architecture of QoSinus permit to easily deploy and improve such an algorithm and to apply this principle with other QoS approaches like bandwidth on demand.

References

[1] Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An architecture for differentiated services. Internet Request For Comments RFC 2475, Internet Engineering Task Force, December 1998.

[2] F. Bonnassieux, P. Primet, and P. Clarke. Network provisioning for the datagrid testbed1. Technical report, EU DATAGRID report Deliverable D7.1 - Approved by the EC January 2001, 2001.

[3] European datagrid ist project. `http://www.edg.org/`.

[4] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy. A distributed resource management architecture that supports advance reservations and co-allocation. In *Proceedings of the International Workshop on Quality of Service*, 1999.

[5] Ian Foster and Carl Kesselman. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, July 1997.

[6] Ian Foster and Carl Kesselman. The grid : Blueprint for a new computing infrastructure. *Morgan Kaufmann Publishers Inc.*, 1998.

[7] Benjamin Gaidioz, Richard Wolski, and B. Tourancheau. Synchronizing network probes to avoid measurement intrusiveness with the network weather service. In *Proc. 9th IEEE Symp. on High Performance Distributed Computing*, 2000.

[8] J. Montagnat, H. Duque, J.-M. Pierson, V. Breton, L. Brunie, and Magnin I.E. Medical Image Content-Based Queries using the Grid. In *Healthgrid'03*, pages 138–147, Lyon, France, January 2003.

[9] G.P. Penney, J. Weese, J.A. Little, P. Desmedt, D.LG. Hill, and D.J. Hawkes. A Comparison of Similarity Measures for Use in 2D-3D Medical Image Registration. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'98)*, volume 1496 of *LNCS*, pages 1153–1161, Cambridge, USA, October 1998. Springer.

[10] P. Primet and F. Chanussot. Spécification du service actif qosinus. Technical Report RR-0287, INRIA, 2003.

[11] A. Roche, G. Malandain, A. Ayache, and S. Prima. Towards a Better Comprehension of Similarity Measures Used in Medical Image Registration. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'99)*, volume 1679 of *LNCS*, pages 555–564, Cambridge, UK, September 1999. Springer.

[12] P. Vicat-Blanc/Primet. High performance grid networking in the datagrid project. *special issue Future Generation Computer Systems*, January 2003.

[13] P. Vicat-Blanc/Primet, G. Romier, and M. Soberman. Le projet e-toile: développement et mise en oeuvre d'une grille haute performance. In *Proceedings of "Journées Nationales du RNTL 2002"*, 2002.

[14] Vthd. `http://www.vthd.org/`.

[15] R. Wolski. Dynamically forecasting network performance using the Network Weather Service. *Cluster Computing*, 1(1):119–132, 1998.

DS Class	Share
Expedited Forwarding	10%
Assured Forwarding for TCP	30%
Assured Forwarding for UDP	30%
Best Effort	30%

Table 3: The statical bandwidth share of VTHD DiffServ classes

	DB1	**DB2**	**DB3**
Number of images	124	238	456
Dimensions	256×256	181×217×181	181×217×181
Size	65536	7109137	7109137

Table 4: Test images size and greylevel ranges.

	Size	100kbps	1Mbps	50Mbps	1Gbps
DB1	7.75 MB	23mn	2mn	28s	1,4s
DB2	3.15 GB	80h	8h	1h30	5mn
DB3	6.04 GB	144h	15h	3h	8mn

Table 5: Database transfer time evaluation for Content Based search Medical application

n	undersampling	SD	SSD	CC	RC	Woods	MI
DB1	8 bits	0.046	0.050	0.063	0.060	0.061	0.070
n=65536	9 bits	0.104	0.110	0.139	0.133	0.134	0.150
DB2	8 bits	0.795	0.798	0.810	0.808	0.810	0.813
n=7109137	12 bits	13.79	14.46	16.17	16.05	15.87	15.92
	16 bits	697.3	697.4	700.3	702.6	802.7	804.3
DB3	8 bits	1.038	1.046	1.062	1.057	1.058	1.092
n=7109137	12 bits	13.97	14.54	18.00	17.62	17.61	18.89

Table 6: Computation times for similarity measures computation.

		image size		
Class used	Concurrent traffic	72KB (μs)	13.5MB (s)	10MB (s)
Best Effort	0 Mbits/s	782.26	0.14	0.11
	267 Mbits/s	869.95	0.16	0.12
	535 Mbits/s	1417.85	0.26	0.20
	840 Mbits/s	5084.69	0.94	0.70
	1070 Mbits/s	8158.01	1.50	1.13
EF	0 Mbits/s	780.19	0.14	0.11
	267 Mbits/s	869.95	0.16	0.12
	535 Mbits/s	934.74	0.17	0.13
	840 Mbits/s	934.74	0.17	0.13
	1070 Mbits/s	934.74	0.17	0.13
AF	0 Mbits/s	783.30	0.14	0.11
	267 Mbits/s	809.09	0.15	0.11
	535 Mbits/s	966.92	0.18	0.13
	840 Mbits/s	1276.68	0.23	0.18
	1070 Mbits/s	1750.22	0.32	0.24

Table 7: Estimates of network image transfer time in different traffic conditions.

		transfer (s)			
Class used	Concurrent traffic	3227MB	807MB	323MB	32MB
Best Effort	0 Mbits/s	36.0	9.0	3.7	0.4
	267 Mbits/s	39.9	10.0	4.0	0.4
	535 Mbits/s	65.2	16.3	6.5	0.6
	840 Mbits/s	234.5	58.4	23.2	2.2
	1070 Mbits/s	358.1	109.3	65.4	3.8
EF	0 Mbits/s	36.1	9.1	3.7	0.4
	267 Mbits/s	39.9	10.0	4.0	0.4
	535 Mbits/s	42.9	10.7	4.3	0.4
	840 Mbits/s	42.9	10.7	4.3	0.4
	1070 Mbits/s	42.9	10.7	4.3	0.4
AF	0 Mbits/s	36.1	9.1	3.7	0.4
	267 Mbits/s	37.7	9.4	3.8	0.4
	535 Mbits/s	45.1	11.3	4.6	0.4
	840 Mbits/s	60.1	15.1	6.0	0.6
	1070 Mbits/s	82.1	20.4	8.1	0.7

Table 8: Network transfer time in different traffic conditions, corresponding to Database transfers.

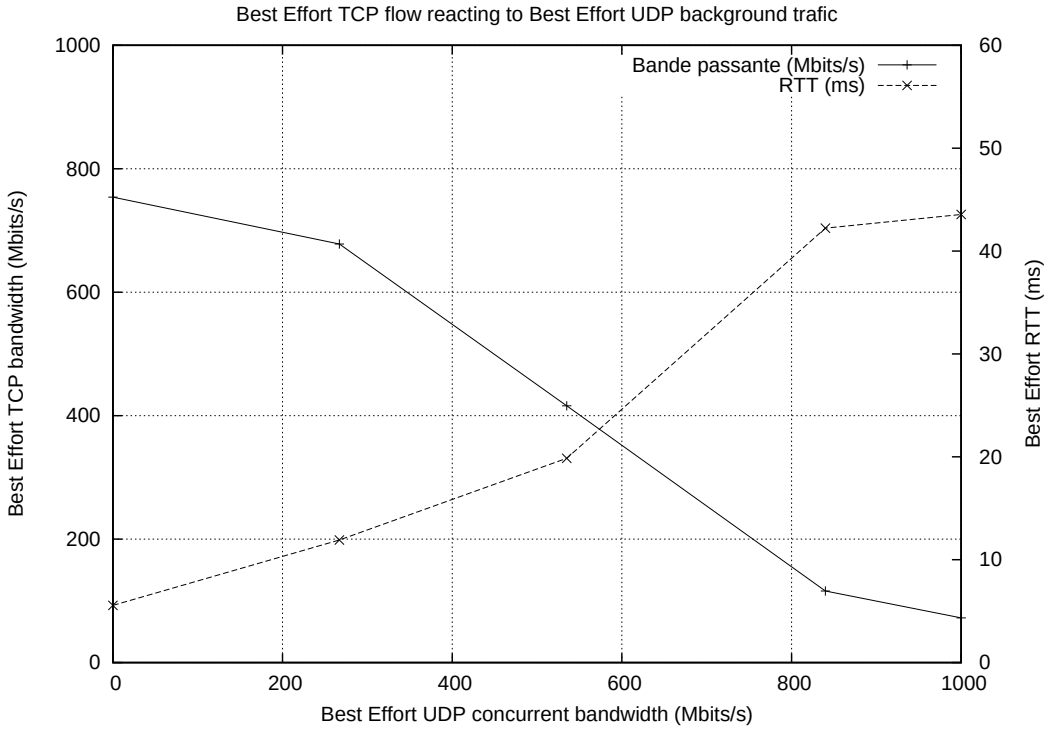

Figure 1: Best Effort performances

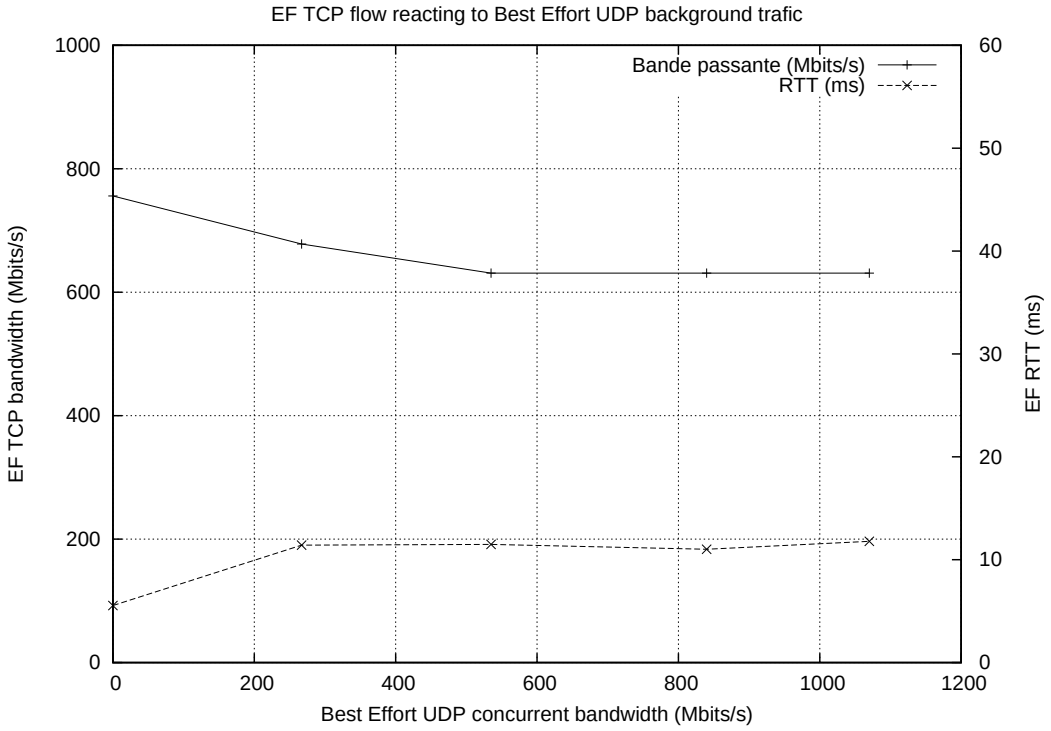

Figure 2: EF performances

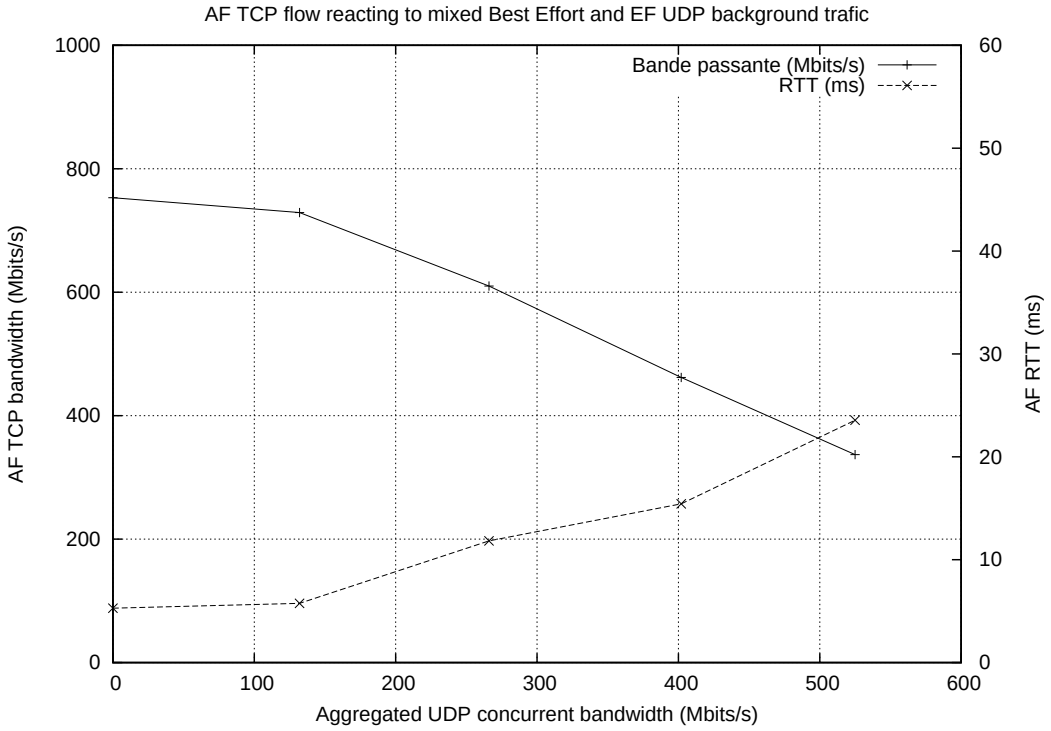

Figure 3: AF performances

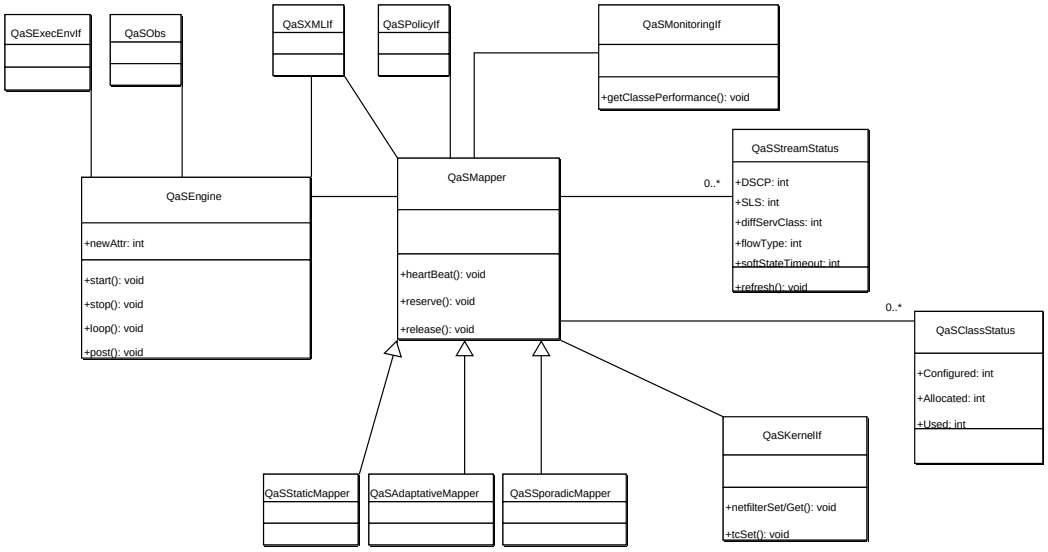

Figure 4: The QoSinus class diagram. The QaSEngine corresponds to the programming component, the QaSMonitoringif corresponds to the performance measurement component interface, the QaSMapper is the adaptive control component that can be a static mapper, an adaptive mapper or a sporadic mapper, QaSkernelif represents the interface to the enforcement component.

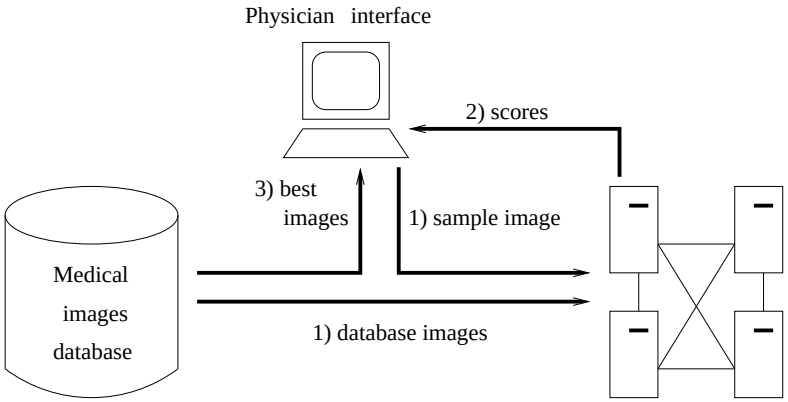

Figure 5: Content-based query on medical images

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)

Unité de recherche INRIA Futurs : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399