GCViR: Grid Content-based Video Retrieval with work allocation brokering

Pablo Toharia[1], Alberto Sánchez[1*], José Luis Bosque[2] and Oscar D. Robles[1]

[1] *E.T.S. de Ingeniería Informática, Universidad Rey Juan Carlos, Madrid, Spain*
email: {pablo.toharia,alberto.sanchez,oscardavid.robles}@urjc.es
[2] *Facultad de Ciencias, Universidad de Cantabria, Santander, Spain*
email: joseluis.bosque@unican.es

SUMMARY

Nowadays TV channels generate a large amount of video data each day. A very huge number of videos of news, shows, series, movies, and so on have to be stored with the aim of being accessed later on. Moreover, channels have a clear need of sharing videos so as to settle a real collaboration among them that minimizes the cost of information acquisition. These features demand a huge storage capacity and a sharing information environment.

Both requirements can be solved by using grid computing. It provides both computing and storage capacities to store that great volume of data required, as well as the resource sharing capabilities for the cooperation of different TV channels.

This paper presents a video retrieval system that covers these needs and suggests a work allocation broker to improve the performance of video accesses. An evaluation shows the feasibility and the scalability of this approach showing the benefits of the work allocation made to store and retrieve large video data.

KEY WORDS: *Video retrieval, grid computing, Zernike invariants*

*Correspondence to: alberto.sanchez@urjc.es

1. Introduction

As a consecuence of Globalization, information is no more the heritage of a few, since it flows between geographically remote locations for its quick and instant mass consumption.

The media, and quite specially TV Channels and multimedia broadcasting corporations, have the need of storing and managing large amounts of multimedia information, that heavily increases day by day. It requires a high storage capacity, as well as an efficient and distributed management, since either resources or users may belong to different organizations.

Also, quite specific software and hardware resources are required for a preprocessing stage that includes a complex process of upload, tagging and storage. Some of these operations (e. g. tagging and classification) are currently done manually (see Section 3). Some others (as segmentation in sequences or shots) are not done at all, since it is a quite time-consuming operation to be done manually. For all these reasons, whole videos are stored as they arrive, something that becomes a penalty during an information retrieval operation on long ones.

Moreover, events follow one another in different locations throughout the world. It is hence quite clear the need of managing physically distributed resources. TV channels have head offices or partnership agreements with local channels, which sometimes belong to the same business corporation. It is quite common for them to share resources as digital archives, but specially videos about local events.

Additionally, it can be pointed out the need of accesing information almost in real-time, since the events happening loose their interest from one day to the following. This instantaneity poses certain problems when such a huge volume of information has to be managed. For example, a journalist needs some images to mount a news video in a quite short period of time, as he is in a rush and cannot wait for hours. In this way, when an event takes place the audience wants to watch things as they happen. This demand cannot be always satisfied, specially when unexpected events take place in locations without regular TV coverage.

All this problems come then with the need of efficiently sharing, storing and managing a huge amount of resources, both hardware (computing and storaging) and software (applications and videos), at geographically remote locations. Sharing them also means to apply security policies so as to decide what to share and to whom share that information. In addition, approaches adopted should be highly scalable since the amount of information to process increases exponentially in the course of time. To conclude, a fully distributed control is needed to ensure an efficient operation not only nowadays, but in the future.

On the other hand, grid computing [19] is becoming nowadays a feasible solution for computer applications with high levels of computational power demand. This is due to the good price/performance ratio offered by the networks that compose this type of systems and because of both the high flexibility and availability offered by this computation paradigm. Furthermore it enables resource sharing among different institutions and individuals [3]. All of this makes grid a suitable infrastructure for retrieving multimedia information, making the cooperation and resource sharing among different TV channels easy.

This paper presents a grid system for providing video upload, storage and content-based retrieval of video in grid environments, following the suitable combination of grid computing and multimedia retrieval systems. The proposed system supports all the above different video storage and retrieval services. But, beyond that, a work allocation broker has been proposed to improve the performance of the video retrieval stage. The broker [48, 54] is vital for any grid infrastructure since its operation and performance determine the required user experience and the use of the environment. Thus, the system capacities depend on the way the broker selects resources. It must be noticed that as far as the authors know there are no other brokers designed for these tasks in the bibliography. The proposed improvement affects the whole expected system performance since journalists demand to retrieve video data in a interactive way.

The system achieved with all these elements takes response time of the retrieval process almost to real-time, greatly improving its performance. Also, this approach will probe its scalability, both in terms of storaging capacities and in terms of retrieval response time. In this way, the integration of new equipment into the system is completely independent from its architecture or its operating system. Also, this approach makes it possible for different corporations to work in a completely secure environment, as well as to define their own privacy, sharing and data trading policies. Finally, this system allows a collaborative work, since resources and information can be shared or even hired when it is appropriate.

The rest of this article is organized as follows. Section 2 describes some previous related works with the problem herein described. Section 3 presents the system funtional description, focusing on how nowadays video storage and retrieval processes are made in a TV channel. Section 4 talks about the retrieval primitive used in the system. Section 5 discusses the architecture of the system and the proposed work allocation model to provide an efficient video retrieval in grid environments. Section 6 shows the evaluation of the proposal comparing it with other possibilities of decision making. Section 7 presents the main conclusions and outline ongoing and future work.

2. Related work

The easiness of creating big collections of multimedia data has been enforced by different factors, such as technological advances of consumer goods along with their lower prices. But advances in technology related to communication systems and the results yielded in different fields (e. g., signal processing, computer vision or databases) has enforced the use of information systems to store and manage those large amounts of multimedia information [61, 47].

To manage these multimedia databases, it is crucial to extract the most relevant information in order to efficiently access it. It can be considered as mandatory when dealing with unestructured multimedia data collections. Content-based Video Retrieval (CBVR) systems were raised to cover the needs created when these large collections contain digital video. They provide a very useful help to users whose aim is to introduce a query in the system and retrieve from the available datasets those shots or videos that look more similar [2, 10, 1, 31].

The work on this field has significantly grown [27], ranging from using low-level visual data describing properties like color, shape, etc, that compose an individual keyframe [51], to the addition of techniques to merge these different types of data so as to assist the system on giving an answer as much closer as possible to the user's information need [58], to the attempts to automatically determine the presence or absence of semantic features from video shots [45]. It all has come to the rise of the TRECVID annual benchmarking campaign [46], with 54 groups from all around the world along 2007 and 142 during 2008 campaign.

High performance computing fits in a natural way in application areas where large volumes of data are required to be managed or processed.When large volumes of data are considered, as it is very often the case of multimedia databases, it may become necessary to look for parallel solutions in order to process, store and gain access to the available items in an efficient way [44, 49, 28, 6]. Multiprocessor shared-memory architectures are being used for these purposes since a long time ago. During decades, they were the first scientists' choice when dealing with parallel implementations [30, 26, 39]. Nevertheless, the last decade has settled grids as a feasible distributed solution for applications where higher levels of scalability and lower costs are required.

Thus, one application area where the concept of grid computing suits perfectly in is multimedia information retrieval. A content-based image retrieval cluster-based implementation was first introduced in [7], and extended to a grid in [42]. This approach showed a good cost/performance ratio as well as an appreciable degree of fault tolerance, as also states an application of the OurGrid middleware to medical image retrieval [8].

As far as we know there is only one work somehow related to video retrieval in grid by Ewerth *et al.* [13], that talks about a video segmentation service. The implementation presented in this paper includes a video segmentation stage as a part of a more complex service that uploads a new video into the system and extract its shots. But beyond that, it also includes a service that allows a user to retrieve and play shots from the datasets, according to the topics the user is interested in, as it is explained in the following sections.

Moreover, different brokers [22, 11, 9, 57] have been developed to identify, characterize, evaluate, select and reserve the most suitable resources or services for generic problems. There are two tendencies related to the brokering phase:

- Client-broker. This kind of broker is designed to comply with client needs about services selection. It follows client policies without the need of having a global vision of other clients requests. An example of a client-broker is GridWay [24].

- System-broker. In this case, the broker has a complete and global vision of resources and clients of the grid. Thus, it can select the most suitable resources for client jobs with the aim of improving the performance of the whole grid.

Nevertheless, no broker has been previously designed specifically for video storage and retrieval. The specilization of the proposed broker improves the video access performance since it is adapted to these needs. Thus, we have developed a work allocation algorithm for improving the video retrieval stage since it should be interactive. Although the algorithm is specialized for video retrieval, it can be used in other fields.

3. System description

Nowadays, the management of all multimedia information collected by a TV channel is done by means of three basic types of operations: upload, retrieval and sharing. Interviews made to different experts yielded the following description of the processes involved in these three operations [53].

During the upload process, a whole video containing all the information a cameraman has recorded (called rush) is dumped in the storage system. Documentalists tag and classify it manually. It means to assign it a textual description as well as several keywords. They are stored with the video (without any shot or sequence segmentation) in the digital archive. This way, retrieving information can be a quite time-consuming task when videos are very long.

The retrieval process begins when a journalist asks for any piece of video related to some news, using a textual description of what it is being looking for. Using the keywords given by the journalist, a documentalist queries the database and gives the journalist back a list of names of files containing the videoclips that have been found. The journalist will select then the files he thinks are going to be useful and asks for them to the documentalist. This way, the chosen files are send to the server the journalist is working with. This is a long process of around two hours if videos are short (it must be remembered there is no shot or sequence segmentation). Longer videos require more time. Additionally, sometimes it happens that any of the videos satisfies the journalist needs but it is too late to begin the process once again.

National TV channels do not have usually offices throughout a whole country. On the other hand, local channels are not able to send videos via satellite. What they usually do is to send a tape in a car (which has a great bandwidth but also a great latency) to the national channel. It can be seen the whole process presents two serious problems: one related do data sharing among different corporations and another one dealing with data storage, management and retrieval. It all means that many times unexpected events (specially at a local level) can not be properly covered.

The system herein proposed solves these problems. It proposes a solution with response times that allows doing the whole process interactively, setting up a collaborations and sharing data while defining security and privacy policies. Details on all these aspects are explained in the following sections.

3.1. Functional description

It has been explained the normal operation of a TV channel nowadays: how multimedia information is acquired, stored and retrieved when it is necessary. The GCViR system herein described follows that process, so this section presents a functional description that captures the main issues to cover. That is, there will be an upload process as well as a retrieval one.

The *video upload* process in the distributed system can be broken down into three phases: segmentation, keyframe extraction and signature computation, and storage.

- Segmentation is the step that allows to isolate the minimum units with meaning from a video: shots. On one hand both beginning and end of each shot are delimited, on the other hand video contents are managed and processed more efficiently. To achieve this, differences between consecutive frames or groups of frames are computed. These differences are calculated based on some low level primitive that captures information about color, shape, texture, etc. In this case, Zernike invariants presented in Section 4 have been used (see details in [52]).
- Storage is completely distributed over the accessible nodes, since each of the sequences or shots obtained from the segmentation stage can be independently stored in the distributed system. This data distribution is done at two levels. First, shots, keyframe identification and their signatures are distributed throughout the grid.
  Each node in the grid can be a single computer, cluster with multiple nodes, an array of disks and so on. Then, a second level of distribution of data among the physical storage devices available could be performed. In terms of performance and scalability two important advantages are achieved. First, efficient data accessing is provided by using parallel access [38, 43]. Then, in terms of scalability, system's growth does not harm performance but satisfies storage needs by adding new nodes or storaging devices.
- Once the video is temporally segmented, one frame per shot is selected as keyframe (i.e. the one containing representative information from that shot). Details can also be found in [52]. A signature based on Zernike invariants is then computed from each keyframe, as explained in Section 4.2. This task is carried out by each node on the shots it is going to store.

The *video retrieval* process can be broken down into the following stages:

a) Input parameters introduction. The user first selects an image to be used as a query example. Additionally, the user may specify which nodes of the grid environment are going to take part in the retrieval process. Finally the user must specify the number p of results that the system should display as a response for the query.

b) Distributed query. The query example is sent to the grid nodes, that will compute its signature. Details about the retrieval techniques involved in the system can be found in [51, 50]. This step is transparent to the user. This way, a remote job is sent to every grid node. The client remains waiting for all the remote executions to finish.
c) Query and database (DB) image's signature comparison and sorting. The signature obtained in the previous stage is compared with all the DB shots' signatures using a metric based on Euclidean distance. The identifiers of the p most similar images are extracted. Should it become necessary to incorporate a new signature to the group of the best p, the one with the worst ranking within the group would be discarded and the set then newly sorted. A bubble sorting algorithm with *O(np log(p))* order has been used for this purpose, being n the number of images. This step is performed in each of nodes independently and each of them write a file with the local results.
d) Final results selection. Each node generates a single file with its local results. When the client receives the results it makes a new arrangement and actually selects the p most similar signatures of the whole system.
e) Results display. The system provides the user a dataset with the p shots considered most similar to the query one. Then the user can select a sequence, clicking with the mouse, and the system plays the video (a fee can be required acoording to agreements among TV channels). If the result does not satisfy users, they can choose one of the selected shots or enter a new one that presents some kind of similarity with the required image returning to stage a).

This system covers the problems stated in section 3 about uploading and retrieving videos. As a result of the way information is managed and stored, it also completely satisfies all the needs explained about not only resource sharing but remote access to information. It can be said that it keeps the privacy of each organization so this system could be the basis for a model oriented to buy information or rent resources.

4. Retrieval primitive

Retrieving information from videos requires the use of low-level primitives to extract from them color, shape, texture or many other features. Section 3.1 states when it is done within both video upload and retrieval processes. In this work, Zernike invariants have been selected because of its demonstrated good performance in object recognition problems [4, 29] as well as the results achieved in previous experimentation on content-based image and video retrieval [51]. Next section provides a brief mathmatical introduction, whereas section 4.2 presents the computation of the signature using them.

4.1. Zernike Invariants

In 1934, Zernike [62] presented a set of complex polynomials $V_{nm}(x, y)$ that were defined inside a unity radius circle $(x^2 + y^2 \leq 1)$ in the following way:

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho)\, e^{jm\theta} \tag{1}$$

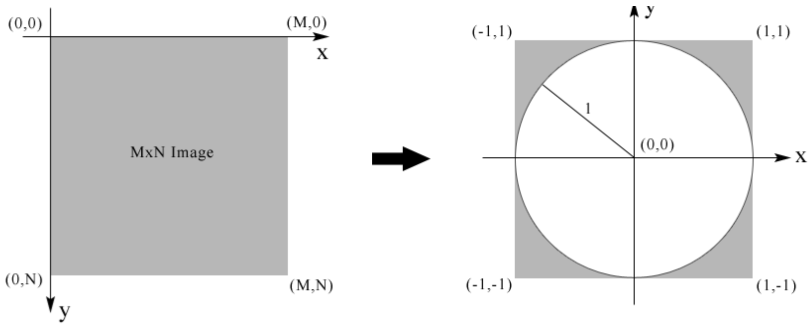

Figure 1. Inscribing Zernike's polynomial definition domain into rectangular images.

where V_{nm} is a complete set of complex polynomials, n is a positive integer value $n \geq 0$ that represents the polynomial degree and m is the angular dependency, ρ and θ are the polar coordinates of the Cartesian coordinates (x, y) and R_{nm} is a set of radial polynomials that have the property of being orthogonal inside the unity circumference. The values of n and m have the following relation:

$$(n - |m|) \bmod 2 = 0 \quad \text{and} \quad |m| \leq n \tag{2}$$

and the radial polynomials have the following expression:

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{m-|n|}{2}} (-1)^s \frac{(m-s)!}{s!(\frac{m+|n|}{2} - s)!(\frac{m-|n|}{2} - s)!} \rho^{m-2s} \tag{3}$$

Starting from Zernike polynomials and projecting the function over the orthogonal basis composed by the polynomials, the moments can be generated in the following way:

$$A_{mn} = \frac{m+1}{\pi} \iint_{x^2+y^2 \leq 1} f(x,y) V_{nm}^{*}(x,y) dxdy \quad \text{with} \quad x^2 + y^2 \leq 1 \tag{4}$$

The discretization needed to work with digital data can be done straightforwardly:

$$A_{mn} = \frac{m+1}{\pi} \sum_{x} \sum_{y} f(x,y) V_{nm}^{*}(x,y) dxdy \quad \text{with} \quad x^2 + y^2 \leq 1 \tag{5}$$

From these functions, we compute the modulus to obtain the p different invariant values for each considered case. The invariant values are used to create a vector of p elements ZI_i that collect the shape information of an image i. For example, in the case of polynomials of 10^{th} degree, p would be 36.

4.2. Signature based on Zernike Invariants

The visual contents of the images are transformed into a vector of some features, named signature, that aims to collect some discriminant information of the original data.

A primitive based on the Zernike Invariants extracted from the original image has been implemented. This way, the signature is generated concatenating the invariants extracted until the maximum polynomial degree considered. A simple vector of scalar values is obtained [51]. As it can be deduced from Equations 3 and 5, the computation of invariants is a very high demanding task from a computational point of view. [51] shows how an interesting trade-off is to use polynomials up to order 10.

An interesting issue faced in the implementation is mapping the images' rectangular domain to the circular space where radial polynomials are defined (Equation 1). The unity radius circle has been inscribed into the image, so its corners have been discarded under the assumption that they do not usually contain relevant information about the scene (see Figure 1).

5. GCViR

Problems shown in Section 3 fit perfectly the definition of a grid environment. The most commonly accepted definition of grid computing is "*to provide flexible, secure, and coordinate resource sharing among dynamic collection of individuals, institutions and resources*" [21]. These collections of individuals, institutions and resources are grouped following the way they share computing resources in virtual organizations (VO) [21]. According to this concept, a TV channel or comercial alliance can be defined as a VO aiming to share its resources (video data, applications, computing and storage resources) collaborating with the rest of TV channels in the way shown in Figure 2. Also, this approach makes it possible for different corporations to work in a completely secure environment, as well as to define their own privacy, sharing and data trading policies.

Following this idea of collaboration among several TV channels, we present a Grid and Content-based VIdeo Retrieval (GCViR) system for providing a flexible and scalable video sharing environment. GCViR offers a good cost/performance ratio to select the most suitable grid resources for data storage in order to store and retrieve large video data among different TV channels.

GCViR has been built following the current grid service oriented architecture. In 2002, the grid community changed its orientation to a services model [20]. This new architecture, named Open Grid Services Architecture (OGSA) [18, 36], proposed to converge the grid computing technology towards Web services providing both creation and maintenance of the different services offered by VOs. Finally, grid services have been fused with Web services (WS) in an single research line named WS-Resource Framework (WSRF) [23]. WSRF provides standard

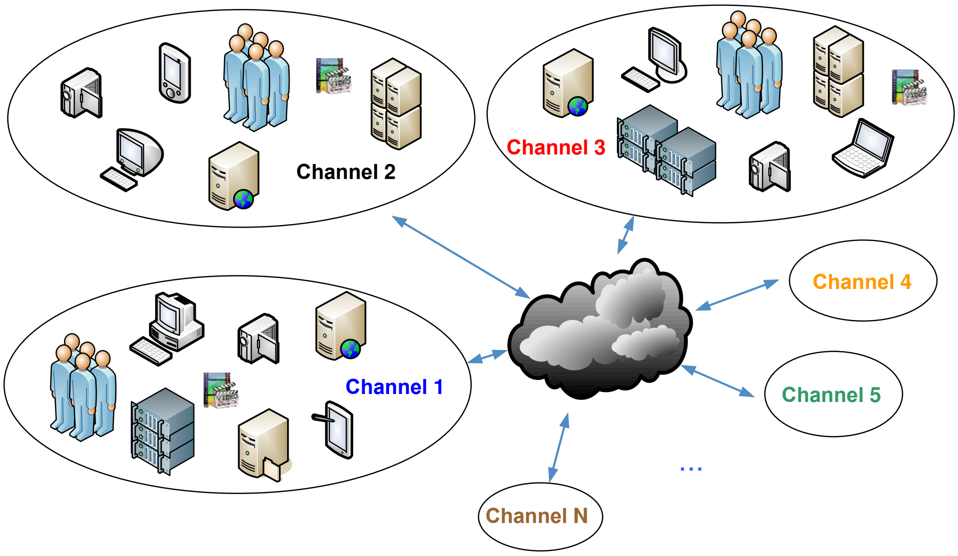

Figure 2. Example of a possible TV channel sharing environment.

access to stateful resources through Web Services.

GCViR is an OGSA and WSRF compliant approach that creates grid services specifically designed for the requirements of TV channels. As underlying infrastructure, a WSRF implementation is required. Globus Toolkit [14, 16] has been chosen for GCViR since it is the most comprehensive and known grid middleware.

5.1. Architecture

GCViR presents a three-tier level architecture based on Globus.

1. Lower layer: it is represented by means of basic grid services, such as WS Grid Resource Allocation and Management (WS GRAM) [59] for running jobs on grid resources, Monitoring and Discovery System (MDS) [33] for resource discovery and monitoring and Reliable File Transfer (RFT) [41] for data management. These services are built upon Grid Security Infrastructure (GSI) [17] and can be combined and used as a basis to deploy and build any high-level services.
2. Middle layer: specialized services used as building blocks to construct high-level video retrieval services. Most of them uses WS GRAM as building block. Although more

services can be defined, the most important services for updating and retrieving videos are the following:

- *Shot boundary detection (SBD)*: It detects shot boundaries in a given video sequence. In this case, the SBD algorithm used is based on the non compressed domain so a first decoding step is needed. This is done using the VD service.
- *Video decoding (VD)*: It is focused on getting a non compress video sequence from a compressed one.
- *Video encoding (VE)*: It allows to compress video sequences using as input a non compress video or a set of images.
- *Shot extraction (SE)*: It takes as input a video with its shot boundary detection output and return one coded video sequence for each shot. This may involve an encoding stage or not depending on the used codecs. VD and VE services may be used in the affirmative case.
- *Keyframe extraction (KFE)*: It extracts a keyframe that represents the content of a given shot.
- *Feature extraction (FE)*: It represents the content of a shot with a small amount of data. This information is also called the signature of the shot. In our case the signature of a shot will be built using its keyframe.
- *Feature comparison (FC)*: It compares two signatures and returns a measure of similarity between them.

Parallel algorithms have been developed to take advantage of compound resources, such as clusters or supercomputers. In this way, parallel versions, [52] and [5], have been used to provide the SBD and FC services respectively. These algorithms are based on MPI [35] enabling its use on both shared and distributed memory machines.

3. Higher layer: lower and middle layer services can be composed to form high-level services, like video upload and retrieval. GCViR provides the capability of service composition, that is, the ability to create workflows, which allows several services to be scheduled in a flexible and efficient manner. Service composition can be performed in two ways [25]. Firstly, vertical composition in which several services whose purposes are different can be combined to build up a high-level service. Secondly, horizontal composition in which several services that have the same functionality can be run in a parallel way can be combined to enhance the performance.
GCViR combines both models to construct a hybrid service composition oriented to improve the performance building complex high-level services. A broker is required in order to efficiently provide this service composition. Next section describes the brokering architecture of the proposed approach. The high-level services for video retrieval are:
    - *Video Upload (VU)*: This high-level service allows the user to upload new videos into the system. The VU service is made regarding specialized and basic grid services. Figure 3 shows the construction of the VU service by means of a joint vertical and horizontal service composition by using the broker where VU service is running.
    - *Video Retrieval (VR)*: This high-level service allows the user to retrieve similar shots using as input an example clip or image. Figure 4 shows how the VR service

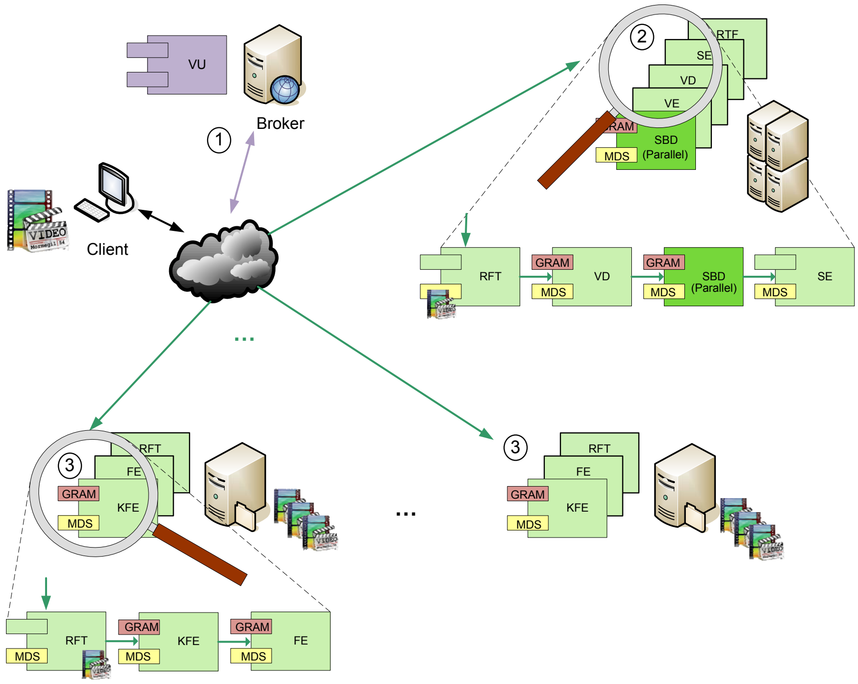

Figure 3. Video upload service. *The client contact the broker to upload a video (step 1). Then, the broker selects the suitable services taking into account both the vertical and horizontal composition. In a first stage the video has to be temporally segmented (step 2). Different services are combined to do this operation. The combination of RFT, SBD and SE services provide, segment and extract shots from the input video data, respectively. Then the video shots are parallelly stored in its final location by using a hybrid service combination (step 3). Not only the vertical combination of services RFT, KFE and FE allows storing videos into the system but also the horizontal composition of each one of these services improve the performance of the operation thanks to the parallel execution of the services in different grid resources.*

works. The output will be a list of the N most similar shots to the given example where N can be defined by the user before querying.

One of the most important aspects of this system is security and the ability to define collaboration or privacy policies for data and resources. To do that, the user access the GCViR system through a web-based GUI interface. GCViR is in charge of providing the required security for this scenario. Firstly, users must prove their identity. To do that, GSI can be used. GSI can be configured to guarantee privacy, integrity and authentication. It is possible to restrict users access by means of a *gridmap* list in each resource of authorized users who can access grid elements. If users appear in the list, they are admitted to the grid with a

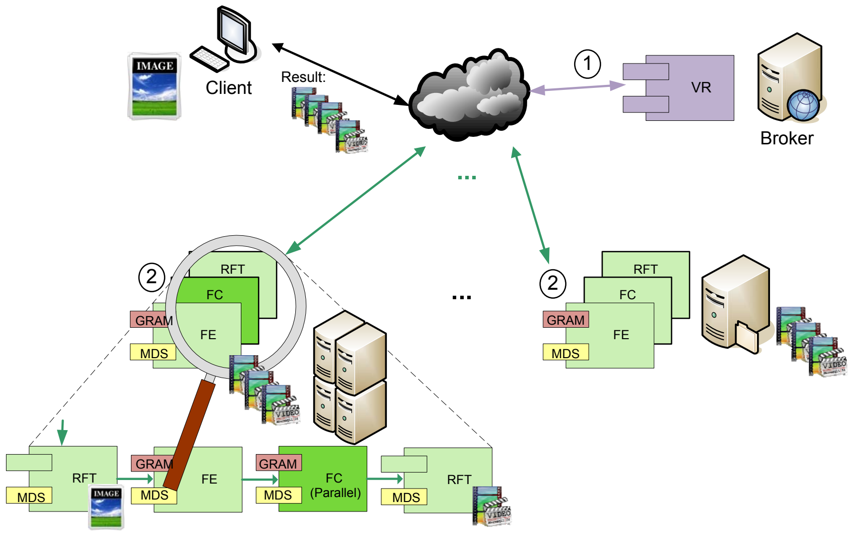

Figure 4. Video retrieval service. *The client has to contact the broker in a similar way than the access to the VU service (step 1). Then, this service involves accessing specialized and grid services FE, FC and RFT following the proposed hybrid service composition (step 2). The FE service is needed to extract the features of the input in order to compute the similarity between the query example and the shots stored in the nodes databases. RFT is used to parallelly send and receive shots from the client to the grid resources. And FC is used to compare the features of the shots stored in the grid resources and the input data in order to obtain a similarity value that will be used to select the most similar shots.*

certain user account and they obtain the corresponding user's rights. Nevertheless, this option can not specify fine-grained access control policies. Community Authorization Service (CAS) [37] solves this problem delegating fine-grained access control policy management to the user community itself. Each TV channel maintains authority over its resources and grants their use to any other institution as a whole. Each TV channel is who restricts their users the rights in a fine-grained way. In this way, GCViR uses CAS and GSI as security infrastructure.

Nevertheless, it is still required to develop policies to assure an environment of confidence and reliability. In this work, it means the application of a set of policies that makes it possible to store video data in a set of resources in which clients trust, e.g. those belonging to their corresponding commercial alliances regarding data ownership. In this way, a client can fix a list of TV channels (VOs) and/or resources as requirement to the broker, that should choose among them.

However, security has also to be provided in a different sense. TV channels can limit access to their resources because of different reasons, such as resource overload or the existence of service level agreements (SLAs) with other TV channels. Thus, it is required to define the system allows the definition of policies to state the way clients can access video data (paying for this service in case it applies). In order to allow a TV channel to control client access, a list for all video data is created in which the access rights of every VO and/or client can be defined following the SLAs among different TV channels. The same public-key certificates used in GSI are used for the purposes of identification. The access permission list will be exported together with video data to the broker. When the broker receives a request of a client who does not have rights to access video data in a certain resource, this one will be directly removed from the candidates list to carry out the operation.

Logically these two protection levels can work simultaneously. In the first step, elements would be restricted according to client requirements. Later on, user rights to carry out the operation in each selected storage element will be checked. Then, the usual decision process continues only with the elements fulfilling both restrictions.

5.2. A broker-based approach

Grid computing does not follow a centralized approach. Focusing on the specific problem, each TV channel has a local administration and security policy and controls the access to their own resources. As a consequence, this shared environment is highly dynamic. Nevertheless, video retrieval services should be distributely accessed across several TV channels. To maintain a global vision of the resources, a broker is needed. Although several generic brokers [22, 11, 9, 57] have been developed in the state-of-the-art, no one has been designed for video storage and retrieval.

The broker is very useful in any grid infrastructure in order to determine to which extent the user requirements are met and how efficiently the underlying resources are utilized. It is possible to take advantage of the service-based vision of grid technology with the aim of making transparent the interaction between clients and grid elements. Specifically, in WSRF, WS are stateless, but they explicitly manage stateful *resources*. The *resource* state is defined by different parameters named WS-Resource Properties in WSRF's terminology. WS-Resource Properties can be published using MDS for each grid element. Any broker, in the case that has rights to access its MDS published information, can know and discover these *resources* and access them by using its EndPoint Reference (EPR) [40]. The EPR uniquely identifies a particular *resource*, including the location of the grid element where it is.

Thus, if an intrinsic association between video shots and *resource* are established, this method enables the discovery of video shots by means of MDS. WS-Resource Properties can be used to store meta-information about the video shot.

Therefore, all shots can be discovered using WSRF mechanisms:

1. The meta-information of each *resource*, which represents a video shot, is published by the grid element.

2. *Resources* are automatically discovered by the broker thanks to MDS. In this way, the broker can know all the EPRs associated with video shots at a given moment. It is possible to access the shots using these EPRs.

In this way, the inherent characteristics of grid environments, like heterogeneity, distributed resources, scalability, and the involvement of multiple administrative domains, are solved by the broker. However, a single broker could be a system bottleneck and a single point of failure due to the large number of grid resources that may be involved. A peer-to-peer protocol [34, 15] is used to address this problem. Firstly, it is possible to achieve reliability by means of broker redundancy. The replication of the broker in other machines provides fault-tolerance, following the idea of a reliable peer-to-peer topology [60]. Nevertheless, this redundancy involves a cost: the network bandwidth decreases because of the higher number of messages transmitted to inform brokers and the workload increases because of processing required to send and receive these messages. Thus, our proposal uses a broker hierarchy due to the shown drawback. This hierarchical structure allows increasing the system scalability by reducing the number of brokers. This idea is gathered from a peer-to-peer point of view in [32].

Once the external broker operations have been characterized, we define its internal design in depth. A broker must contain three components [56]:

1. A resource discovery system: it is aware of available services and current state of grid elements.
2. A matchmaking system: it is in charge of selecting the available candidate elements that match the client's requirements according to the state of the resource.
3. A decision making system: the broker chooses the best grid elements from the previously selected candidates regarding certain algorithm.

GCViR is a system-broker approach as it can be seen in Figure 5. Following its design, the decision making logic its included into the broker instead of into the clients. This means more client simplicity. They can take advantage of the grid, but the broker hides them its complexity. This way, most of the operations are performed by the broker instead of the client. Among other operations, the broker carries out both vertical and horizontal compositions. As a consequence, the developed broker has a matchmaking and decision making system involving two dimensions: vertical and horizontal. In this sense, the broker selects the appropriate services to do each necessary upload process depending on the client needs. Furthemore it points out some services that have the same functionality to be accessed in a parallel way in order to improve the performance.

On the other hand, the retrieval process requires a better efficiency since journalists demand access video data at once to analyze it. Thus, this process takes higher relevance than the upload one. The broker aims to distribute the number of video shots among heterogeneous grid elements according to the their features to provide an efficient retrieval. In this way, all elements should take a similar time to make a video retrieval regardless of their computing and communication capacities. The idea is that any grid element has to wait for the rest of elements providing an efficient access. Therefore, the aim of the decision making is to minimize

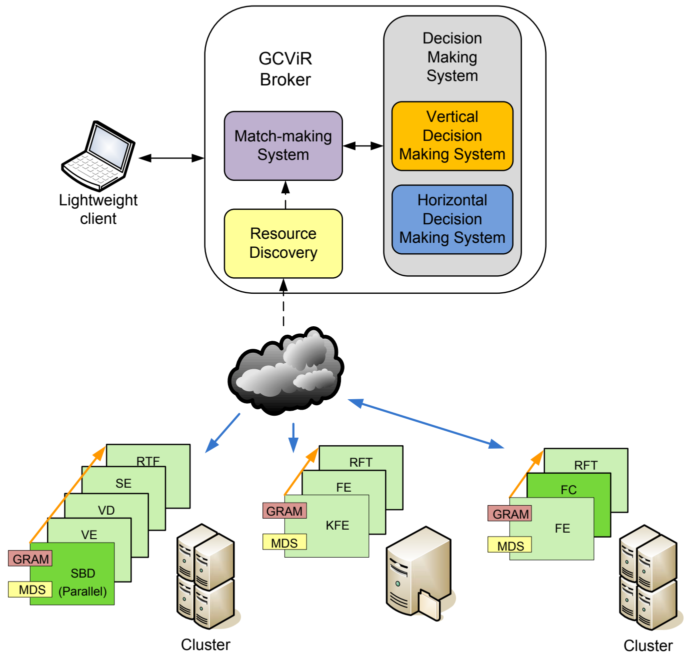

Figure 5. Broker architecture

the response time of video retrieval avoiding situations where resources are overloaded in comparison with other elements.

Thus, the broker defines a proper distribution of video shots or work allocation for each video upload that takes into account the system heterogeneity. The distribution of shots during a video upload is optimal if there are no significant changes in the behavior of each element of the system. Since it is expected that this kind of storage and retrieval system does not change its usual behavior because of the way documentalists use it, the grid element parameters can be *a priori* obtained†. In this case, shots are distributed according to the mathematical model explained in the next section. On the other hand, in case the behavior changes, the parameters have to be recalculated. This means an automatically redistribution of the video shots among grid elements fulfilling the new optimal distribution.

†A priori models work with a previous learning phase where the system is analyzed. These methods try to infer the model that manages the system behavior and the approximated values of the control variables.

5.2.1. Work allocation model

Let $G_1...G_n$ be a set of grid heterogeneous elements in which it is possible to obtain the following parameters:

- Computing power (P_i). It is defined as the expected number of video shots per unit time that a grid element G_i can process. P_i includes CPU time (to compare signatures and sort data), disk access time and service execution time[‡]. Instead of aggregating values, an accurate way to measure this parameter is through the sequentially monitoring of the video retrieval operation over the resource taking an average value. The total computing power of the system is the sum of all of them:

$$P = \sum_{i=1}^{n} P_i$$

- Communication time (C). It is defined as the communication capabilities of the environment. Since all elements transmit the same amount of information to the broker (signatures), it is possible to experimentally obtain C by evaluating the communication between the broker and the rest of grid elements. In this way, $C = C_1, ..., C_n$ represents the average time to send a certain number of signatures from G_i to the broker.
- Workload (W). The workload is represented by the number of video shots that are stored in the system. Each grid element has a workload proportional to its computing power and communication capability named W_i. It is stated that:

$$W = \sum_{i=1}^{n} W_i$$

- Storage capacity (S). This is a limiting factor representing the number of video shots that a grid element G_i can store. The storage capacity available on each node is stored in a vector $S = S_1, ..., S_n$

A work allocation model can be fixed according to the parameters shown above. The objective of this model is to obtain the optimal distribution or work allocation X_i so every grid element G_i spends the same time T to process a video retrieval request of K shots. The following equations are raised (where W_i are the shots stored by G_i and K is the number of shots that are going to upload to the system):

$$T = \frac{W_i + X_i}{P_i} + C_i$$

$$K = \sum_{i=1}^{n} X_i$$

[‡]The use of Globus as infrastructure,e.g by using WS GRAM, implies an overhead.

P_i, C_i and W_i can be obtained using the proper monitoring and known by the broker using MDS that provides information about the available resources in the grid and its state. The unknown quantities are determined by X_i and T. The equation system of $n+1$ unknown quantities and $n+1$ equations is solved for obtaining the suitable X_i.

Finally, it is required to consider the storage capacity checking that each node can store the assigned video shots ($X_i \leq S_i$). If so, S_i is updated. If that not be the case, it is necessary to solve again the system equation setting the value X_i to the storage capacity, i.e. $X_i = S_i$.

Anyway, it is important to notice that the system is able to keep on optimally distributing after non-permanent faults on resources, appearance of elements and so on (in some cases it is required to obtain a new optimal distribution).

6. Evaluation

This section analyzes in depth the performance and different benefits of the proposed approach. This analysis aims at proving the viability of approach and demonstrating the efficiency of the proposed work allocation model. We get to extract some interesting conclusions that assert our proposals by means of this analysis.

Systems evaluations are often made in unusual conditions, mainly due to two different reasons: (i) systems are evaluated through simulations; (ii) the test environment is different from the deployment environment. In order to properly evaluate the system, we have tested it in a real environment. Our testbed consists of several heterogeneous resources geographical distributed through Spanish scientific wide area network, named RedIRIS, over the Comunidad de Madrid (Spain). Resources belong to different sites with different administrators and security policies:

- Arquitectura y Tecnología de Computadores (DAC) Department at the Universidad Rey Juan Carlos (URJC). It is located at the Escuela Técnica Superior de Ingeniería Informática of the URJC (Mostoles, Spain). Its resources are a shared memory multiprocessor (DAC1) composed of 8 processors Intel Itanium 2 1.5GHz with 16 GB of RAM memory and an Intel Pentium III 450 MHz node (DAC2) with 128 MB of RAM memory.
- Lenguajes y Sistemas Informáticos I (LS) Department at the URJC. It is located at the Escuela Técnica Superior de Ingeniería Informática of the URJC (Mostoles, Spain). It shares an Intel Pentium IV 2.66GHz node (LS1) with 256 MB and an Intel Pentium IV 2.4GHz node (LS2) with 1 GB of RAM memory of RAM memory.
- Arquitectura y Tecnología de Sistemas Informáticos Department at the Universidad Politécnica de Madrid (UPM). It is located at the Facultad de Informática of the UPM (Madrid, Spain). Its resources are two clusters UPM1 and UPM2, each one composed of 8 nodes. The nodes of the first one are Intel Xeon 2.40GHz with 1 GB of RAM memory. The nodes of the second one are Intel Xeon 3.0GHz nodes with 2 GB of RAM memory

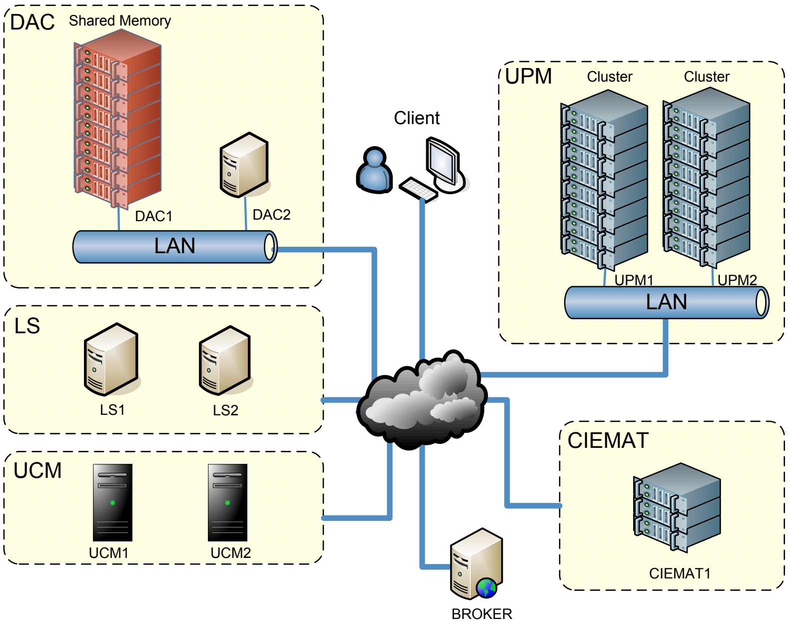

Figure 6. Testbed setup

- Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT). It is located at the Centro de Moncloa of the Ciudad Universitaria de Madrid (Madrid, Spain). It shares a cluster (CIEMAT1) composed of 3 Intel Xeon 2.40GHz nodes connected by a Gigabit network.
- Arquitectura de Computadores y Automática Department at the Universidad Complutense de Madrid (UCM). It is located at the Facultad de Informática of the UCM (Madrid, Spain). Its resources are 2 Intel Xeon 2 GHz nodes (UCM1 and UCM2) with 2 GB of RAM memory.

These sites are part of the research initiative of grid in the Comunidad de Madrid (GRIDIMadrid). For this evaluation, each site can be viewed as a TV channel. Figure 6 shows the grid testbed configuration. An Intel Pentium IV 2.4GHz computer with 2 GB of RAM memory located at the URJC has the role of the broker.

An *a priori* study of the testbed operation revealed the different computing and communication capacities shown in Table I. Each resource presents different values stating the testbed heterogeneity.

Table I. Computing and communication capacities of testbed elements

Resource	P	C
DAC1	228.1	0.31
DAC2	133.14	0.52
LS1	228.78	0.3
LS2	200.92	0.24
UPM1	202.11	0.44
UPM2	210.61	0.29
CIEMAT1	223.73	0.51
UCM1	199.01	0.33
UCM2	207.69	0.42

In order to validate our proposal several experiments have been designed for knowing and testing the system operation. 100 executions have been performed for all experiments. The values measured are its average response time. Experiments have been led with two different video shot distribution in order to compare them to show the benefits of our proposal. The first one, named work allocation (WA), distributes the workload taking into account the computing and communication capacities of each grid resource, as described in the distribution model proposed in Section 5.2. The second one does not take into account if resources are heterogeneous or not. This way, the shots are equally distributed (ED) among the elements available.

6.1. VU service results

This service has been tested deeply by uploading several videos to the system. Table II shows two examples of how the WA and ED distributions work for uploading 128,000 and 1,024,000 video shots to the system. The system previously stores 65,000 video shots. As it can be seen in an ED approach all resources store the same size of video data whereas in the proposed WA model each resource stores a data volume according to its computing and communication capacities.

For this experiment we have measured the response time of each specialized service involved in the *VU* service when using both a single grid resource and the 9 available ones applying or not WA. Table III shows data collected for this experiment. For the configuration with a single resource a couple of runs have been made, one with DAC2 and another one with LS1. These nodes have been chosen taking into account that DAC2 is the node with the lowest computing power and LS1 is the highest.

As it has been stated before, we have develop a parallel version of the SBD service. In the testbed this service is always run in DAC1, a shared memory multiprocessor with 8 processors.

Table II. Shot distribution over the testbed using WA vs. ED

Resource	128K shots with ED Distribution	128K shots with ED DB size (GB)	1024K with ED Distribution	1024K with ED DB size (GB)
All	14,222	97.22	113,778	777.78

Resource	128K shots with WA Distribution	128K shots with WA DB size (GB)	1024K with WA Distribution	1024K with WA DB size (GB)
DAC1	15,959	109.09	127,392	435.56
DAC2	9,203	62.91	74,246	253.47
LS1	16,013	109.46	127,778	436.91
LS2	14,075	96.22	112,230	383.78
UPM1	14,102	96.4	112,838	385.67
UPM2	14,712	100.57	117,600	402
CIEMAT1	15,613	106.73	124,911	426.94
UCM1	13,866	94.79	111,087	379.61
UCM2s	14,456	98.82	115,918	396.07

Table III. Breakdown of time consumed for the *VU* service

Num. nodes	SDB	SE	T_{S3}	RFT (sequential)	GRAM (sequential)	KFE (sequential)	FE (sequential)	T^{seq}_{S3} RFT+EKF+FE+GRAM
9 with WA	1,183.5	6.1	114.6	153.6	402.0	118.0	0.7	674.4
9 with ED	1,183.5	6.1	122.7	189.9	430.1	134.4	0.7	755.3
1 (DAC2)	1,183.5	6.1	1287.8	155.8	345.9	749.0	2.3	1253.1
1 (LS1)	1,183.5	6.1	415.8	61.3	236.6	105.3	0.5	403.8

That is the reason why the SBD time is the same for the 4 configurations. As shown in previous work [52], this represents a speedup value around 7 for 8 processors, which in fact means that if this process is done sequentially it would take around 8,400 seconds.

As shown in the SBD service, the SE service is done in the same multiprocessor so the time consumed is again the same for all the configurations. The following column (T_{S3}) represents the effective time taken to do the rest of the upload process, that is, the RFT, WS GRAM, KFE and FE services which conform the step 3 in Figure 3.

Sequential times are shown in the right side of the table. The RFT, WS GRAM, KFE and FE sequential times have been individually measured in each resource for each service. The aggregated time of all resources is shown in the table. Finally the last column (T^{seq}_{SEC}) shows the sequential time for the step 3 mentioned before. This value is obtained by adding the four RFT, WS GRAM, KFE and FE sequential times. These values allow us to analyze the benefits of the horizontal composition.

As it can be seen T_{S3} using 9 nodes with WA (114.6) represents a time reduction $(1-(T_{S3}/T_{S3}^{seq}))$ of 83% against its sequential time T_{S3}^{seq} (674.4). In the case of an ED approach using 9 nodes the time reduction is 83.75%. The time reduction for ED is higher since there is much more room for improvement because the response time is worse than with WA. Furthermore, it should be noted that in distributed heterogeneous environment like a grid, worse resources are penalizing the best ones. Otherwise, the improvement could be higher. As expected if we do the same comparison in the single resource configurations we can see that there is no reduction at all because this configuration does not take advantage of the horizontal service composition. The slight differences are due to the way the time is monitored. All these results state that the improvement obtained thanks to the horizontal composition is very meaningful.

On the other hand, it is possible to obtain how much faster horizontal composition is when using 9 nodes than a corresponding execution in a single grid resource. Comparing the consumed time of the WA configuration using 9 grid elements and a single resources a 11.23 speedup (1287.8/114.6) is achieved vs. DAC2 and a 3.63 (415.8/114.6) vs. LS1. The difference between speedups is due to the different characteristics of each grid resource, being DAC2 the one with the worse performance and LS1 the opposite case. In case of using an ED approach the speedup obtained vs. DAC2 is 10.49 and a 3.38 vs. LS1. Again, we can see horizontal composition of services benefits the video accesses when several grid elements are used in parallel. Nevertheless, it should be noted that the speedup obtained for the WA distribution is higher than the achieved by the ED approach. Although the proposed work allocation model only aims to improve the video retrieval stage, in the case that the system is loaded enough, as it occurs in this test, the proposal improves the operation of a equally distributed upload.

6.2. VR service results

This section presents two different experiments to evaluate both the performance and the scalability of the grid environment.

The first experiment has two aims. Firstly, it tries to establish how the number of shots in the system affects the response time. Secondly, it proves that the achieved times are in the range of what an interactive service should perform. For this purpose the number of resources used has been fixed and we have varied the number of shots stored in the system (from 500 to 1,024,000).

Figure 7 shows the results collected for this experiment. First, it should be noted that the achieved response times are under a minute in all cases. This can be considered good enough for an interactive retrieval system like the one proposed. The experiments collected values for a database containing from 500 and up to more than a million, representing an increase of

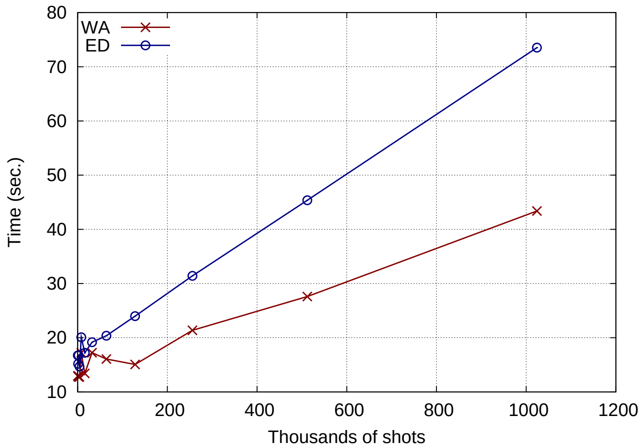

Figure 7. Retrieval response time using all the nodes available with different number of thousands of shots.

200 times the initial database size. Nevertheless, the response times increased from 12 to 43 seconds, which means a factor of 4. This is mostly due to the fact that the communication overhead is almost constant and thus it has a higher impact when the workload is small (especially the workload is very small when the FC is run in the developed parallel version [5] in a compound element, like a cluster). However, from these experiments it can be stated that performance achieved is satisfactory and meets the goals presented previously, offering good level of scalability when increasing the number of shots managed.

On the other hand it can be observed that for a limited number of shots the response times remain very similar. This is due to the fact that most of the response time is taken by the communication overhead of the middleware used (Globus in this case). Nevertheless, it can be seen a small gain of using the WA model which takes into account the heterogeneity of the system. Moreover, when increasing the number of shots stored in the system, the communication overhead remains almost constant and the response times increment are due to the actual retrieval process. In this case, the use of the WA approach has a higher impact achieving 41% time on the retrieval process less than the ED approach. These results show that high importance of having into account the degree of heterogeneity when distributing the shots.

The second experiment tries to evaluate the scalability of the system when increasing the number of nodes available, and thus, increasing the global workload. For this purpose response times of different configurations of the system have been measured. In this case the different configurations vary the number of nodes used for the retrieval process from 1 to 9. 128,000 shots are stored in the system if all elements are working. Otherwise, only the proportional

Table IV. Retrieval response using subsets of nodes.

Num. nodes	Random		Best-to-worse		Worse-to-best	
	with WA	with ED	with WA	with ED	with WA	with ED
1	8.86	9.72	4.96	3.57	15.13	19.96
2	11.03	12.00	3.77	3.63	14.96	20.67
3	12.45	13.10	4.31	3.90	15.46	19.28
4	12.93	14.49	4.14	4.13	15.82	19.76
5	13.67	14.32	6.54	6.23	18.38	24.77
6	15.06	20.15	6.53	6.67	19.12	24.49
7	15.73	17.05	6.46	6.11	14.94	20.21
8	16.21	18.57	7.55	7.80	15.90	19.89
9	16.36	21.39	15.79	20.21	15.50	19.16

part to this volume and the number of resources is stored among the selected grid elements. Thus, when a node is added the global workload is proportionally increased.

For this experiment, the following configurations have been used:

- Random: for each execution, the nodes involved are picked randomly from the whole set available.
- Best-to-worse sorting: The nodes are sorted by their computing power and they are added in the same order. In this way when using just one node the most powerful one will be used, when using two, the two most powerful will be used, and so on.
- Worse-to-best sorting: This configuration is similar to the previous one but this time doing a reverse sorting of the computing power.

Table IV collects the results obtained. In most of the experiments it can be seen that the results achieved using WA are better than the ones without it. This proves the big advantage for the scalability of having into account the heterogeneity of the system.

From the randomly selected configuration it can be deduced that in fact the response time increases when adding new nodes to the system. This is a logical behavior due to the overhead of the communications involved. However, it should be noted that when a node is added to the system it also increases the size of the database, that is, the workload of the system. A system is considered scalable when its response time increase is lower or equal than the linear increase expected due to the growth of the workload. For our system this relationship is depicted in Figure 8. It can be observed that in our case the increment is quite below linear, which in fact means it has very good scalability properties.

Analyzing the results of the best-to-worse sorting it can be observed that response time when using a small number of nodes are outstanding and the increment suffered adding a new node is almost negligible compared to the increment of the database size managed. However, this

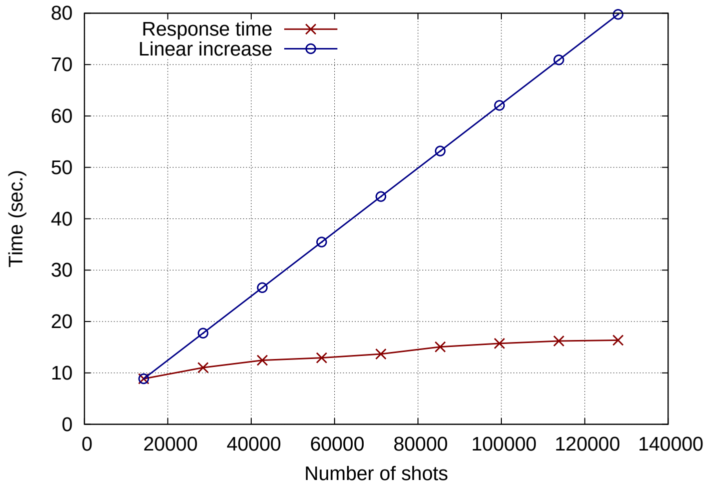

Figure 8. Scalability study of the retrieval service.

tendency is broken when adding the worst node. In this last case, the communication overhead increases rapidly making the response time worse. The proposed WA model mitigates this effect but even tough it is still quite important. From this data it can be concluded that the slowest nodes penalize the performance of the whole system. In the same way, it seems that it would be more efficient to use a smaller system with powerful nodes.This statement is also confirmed by the results of the worst-to-best sorting configuration.

7. Conclusions and future work

The work in this paper has pointed out the amount of problems that arise due to the way TV channels work. However, these problems can be solved using current high performance computing solutions. Specifically, grid technology can yield significant contributions taking into account the distributed and dynamical nature of the information.

To the authors knowledge the is not any other work tackling this problem except the one already presented by the authors [53]. A large number of improvements have been now introduced in a new prototype: the work allocation model which allows taking into account the heterogeneity of the system as well as taking advantage of it to improve response time; parallelization of the most time consuming services; the increase of the number of services and their functionality or a more exhaustive experiment set.

The proposed solution opens a new range of possibilities in the way work is done nowadays, proposing new functionalities and solving the problems explained in Section 1. In this way, a highly dynamic and optimized grid-based system has been proposed. In order to make it feasible it has been based in four key concepts: collaboration, efficiency, scalability and security.

From a qualitative point of view the system contributes with new functionalities which can not be carried out in today's work model. It helps the development of a collaborative work environment based on the agreements achieved by different organizations or a market-oriented model which can consider renting or selling resources, services and information. It allows to give remote access to information. For example, national channels can access information provided by local news centers, that otherwise they could not cover. Another advantage is the economical savings companies can benefit of. This savings can come from different sources: time saving, sharing computation resources which would be expensive to buy, or just the simple fact of sharing information which allows having less copies of the same data.

Efficiency has been proved by experimental results in Section 6. The proposed system is interactive and allows processing in seconds what nowadays takes hours. The work allocation model presented enables taking into account the inherent heterogeneity of this kind of systems and optimize its performance. On the other hand, a parallel implementation of the most time consuming services has been used [52, 5]. It allows these services to run in parallel, both in cluster or shared memory architectures, achieving almost lineal speedup figures.

Scalability is a fundamental property in this case because it can be said these systems needs will increase in the near future. Previous experiences in other fields show this tendency will not stop growing from a computational point of view, since the increase of digital processing possibilities will also cause an increase on computing needs, as well as from a storage point of view since day by day new video contents are produced and have to be stored.

Joining efficiency and scalability allows establishing a maximum response time not affected at all by the amount of information to be managed. In case of a great workload the broker can establish response times are over a threshold. At that point new resources can be added and then the broker can perform a new distribution of workload so the response time is again under the desired maximum.

Both GSI and CAS guarantee the system security. Each participant can define its own security policies and restrict user access following them. This way, real collaboration is possible without compromising each organization privacy. This makes it easier to develop in a future a market-oriented model in which access to resources is charged.

There are pending tasks to be considered for future work. Firstly, a grid is a dynamic environment prone to faults because of its heterogeneity and decentralization characteristics. In this way, data replication is one of the keys to ensure the availability of video data access services avoiding reliability problems. The main difficulty of data replication in grid environments is to choose the set of replication policies that manages the system. The inherent variability of the environment makes it difficult to choose a guideline that optimizes the process. Moreover, environment conditions can change suddenly worsening the performance. Therefore, it should be advisable to define a self-adaptive approach that allows the system to overcome grid changes readjusting their replication policies.

On the other hand, we are working on improving the broker scalability. To do that, we plan to study in depth whether it is better a hierarchical solution or a completely distributed one. Another aspect that could improve the system's performance under additional workload would be to add a dynamical workload balancing algorithm so unloaded nodes help most loaded ones to carry out their work along certain periods of time.

Acknowledgments

This work has been partially funded by the the Spanish Ministry of Education and Science (grants TIN2007-67188, TIN2007-68023-C02-01 and Consolider CSD2007-00050), as well as by the HiPEAC European Network of Excellence. Data used in this work was provided by the TRECVID project, sponsored by the National Institute of Standards and Technology (NIST) with additional support from other U.S. government agencies. Grid resources used are part of GRIDIMadrid *http://www.gridimadrid.org/*.

REFERENCES

1. Sameer Antani, Rangachar Kasturi, and Ramesh Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 35(4):945–965, April 2002.
2. Y. A. Aslandogan, C. Thier, and C. Yu. A system for effective content based image retrieval. In *ACM Multimedia'96*, pages 429–430, Boston, MA, EEUU, 1996. ACM.
3. Fran Berman, Geoffrey Fox, and Anthony J.G. Hey, editors. *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons, April 2003. ISBN 0-470-85319-0.
4. Y. Bin, P. Jia-Xiong. Invariance anallysis of improved Zernike moments. *Journal of Optics A: Pure Application Optics*, 4(2002): 606–614.
5. José L. Bosque, Oscar D. Robles, Luis Pastor, and Angel Rodríguez. Parallel cbir implementations with load balancing algorithms. *J. Parallel Distrib. Comput.*, 66(8):1062–1075, 2006.
6. José Luis Bosque, Óscar David Robles, Luis Pastor, and Angel Rodríguez. Performance analysis of a CBIR system on shared-memory systems and heterogeneous clusters. In *Proceedings on International Workshop on Computer Architectures for Machine Perception, IEEE CAMP 2005*, pages 309–314, Palermo, Italy, July 2005. IEEE, IEEE Computer Society.
7. José Luis Bosque, Óscar David Robles, Angel Rodríguez, and Luis Pastor. Study of a parallel CBIR implementation using MPI. In Virginio Cantoni and Concettina Guerra, editors, *Proceedings on International Workshop on Computer Architectures for Machine Perception, IEEE CAMP 2000*, pages 195–204, Padova, Italy, September 2000. ISBN 0-7695-0740-9.
8. Francisco Brasileiro, Eliane Araujo, William Voorsluys, Milena Oliveira, and Flavio Figueiredo. Bridging the high performance computing gap: the ourgrid experience. In Bruno Schulze, Rajkuma Buyya, Philippe Navaux, Walfredo Cirne, and Vinod Rebello, editors, *Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the GRID, CCGRID 2007*, pages 817–822, Rio de Janerio, Brazil, May 2007. IEEE, IEEE. doi: 10.1109/CCGRID.2007.28.
9. R. Buyya, D. Abramson, and J. Giddy. Nimrod/g: an architecture for a resource management and scheduling system in a global computational grid. volume 1, pages 283–289 vol.1, 2000.
10. Alberto del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, San Francisco, California, 1999. ISBN 1-55860-624-6.

11. Catalin Dumitrescu, Ioan Raicu, and Ian Foster. DI-GRUBER: A Distributed Approach to Grid Resource Brokering. In *SC '05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, page 38, Washington, DC, USA, 2005. IEEE Computer Society.
12. M. S. Pérez, A. Sánchez, V. Robles, P. Herrero, and J. M. Peña. Design and implementation of a data mining grid-aware architecture. *Future Generation Computer Systems*, 23(1):42–47, 2007.
13. Ralph Ewerth, Thomas Friese, Mathias Grube, and Bernd Freisleben. An efficient indexing method for content-based image retrieval. In *Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering, ISMSE'04*, pages 164–168, Kumamoto, Japan, December 2004. IEEE, IEEE. doi: 10.1109/MMSE.2004.47.
14. I. Foster. Globus Toolkit Version 4: Software for Service-Oriented Systems. In *IFIP International Conference on Network and Parallel Computing*, volume 3779 of *Lectures Notes in Computer Science*, pages 2–13. Springer Verlag, 2005.
15. I. Foster and A. Iamnitchi. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In M. F. Kaashoek and I. Stoica, editors, *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, volume 2735 of *Lecture Notes in Computer Science*, pages 118–128. Springer, 2003.
16. I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, Summer 1997.
17. I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *ACM Conference on Computer and Communications Security*, pages 83–92, 1998.
18. I. Foster, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, and J. Von Reich. The Open Grid Services Architecture. Technical Report Version 1.5, Global Grid Forum, June 2006.
19. Ian Foster and Carl Kesselman, editors. *The Grid 2: Blueprint for a New Computing Infrastructure*. Computer Architecture and Design. Morgan Kaufmann, 2 edition, November 2003.
20. Ian Foster, Carl Kesselman, Jeffrey M. Nick, and Steven Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Published online at *http://www.globus.org/research/papers/ogsa.pdf*, January 2002.
21. Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15(3):200–222, 2001.
22. James Frey, Todd Tannenbaum, Miron Livny, Ian Foster, and Steven Tuecke. Condor-g: A computation management agent for multi-institutional grids. *Cluster Computing*, 5(3):237–246, 2002.
23. S. Graham, A. Karmarkar, J. Mischkinsky, I. Robinson, and I. Sedukhin. Web Service Resource Framework. Technical Report Version 1.2, OASIS, 2006.
24. GridWay Metascheduler: Metascheduling Technologies for the Grid, accessed Feb 2009 [Online]. Available: *http://www.gridway.org/*.
25. Ahlem Ben Hassine, Shigeo Matsubara, and Toru Ishida. A constraint-based approach to horizontal web service composition. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 130–143. Springer, 2006.
26. Kai Hwang. *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. McGraw-Hill, 1993. ISBN 0-071-13342-9.
27. E. Izquierdo, J. Benois Pineau, and R. Andre Obrecht. Special issue on content-based multimedia indexing and retrieval. *Signal Processing: Image Communication*, 22(7-8):605–606, August 2007.
28. O. Kao, G. Steinert, and F. Drews. Scheduling aspects for image retrieval in cluster-based image databases. In R. Buyya, G. Mohay, and P. Roe, editors, *Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 329–336, Brisbane, Australia, May 2001. IEEE Comp. Soc. ISBN 0769510108.
29. A. Khotanzad and Y. H. Hong. Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.
30. E.V. Krishnamurthy. *Parallel Processing: Principles and Practice*. Addison-Wesley, 1989.
31. Oge Marques and Borko Furht. *Content-based Image and Video Retrieval*. Multimedia Systems and Application Series. Kluwer Academic, 2002.
32. P. McBrien, N. Rizopoulos, C. Lazanitis, and Z. Bellahsène. iXPeer: Implementing layers of abstraction in P2P Schema Mapping using AutoMed. In *2nd Workshop on Innovations in Web Infrastructure*, Edinburgh, UK, May 2006.
33. Globus: Monitoring and Discovery System, accessed Feb 2009 [Online]. Available: *http://www.globus.org/mds/*.

34. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. Technical Report HPL-2002-57, HP Lab, 2002.
35. MPI - the Message Passing Interface standard, accessed Feb 2009 [Online]. Available: *http://www-unix.mcs.anl.gov/mpi/*.
36. Globus: The Open Grid Service Architecture, accessed Feb 2009 [Online]. Available: http://www.globus.org/ogsa/.
37. L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A community authorization service for group collaboration. In *POLICY '02: Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02)*, page 50, Washington, DC, USA, 2002. IEEE Computer Society.
38. M. S. Pérez, A. Sánchez, P. Herrero, and V. Robles. *A New Approach for overcoming the I/O crisis in grid environments*, chapter 19, pages 311–321. American Scientific Publisher, January 2006.
39. I. Pitas, editor. *Parallel Algorithms for Digital Image Processing, Computer Vision and Neural Networks*. John Wiley & Sons, 1993.
40. W3C Recommendation. Web Services Addressing 1.0 - Core, accessed Feb 2009 [Online]. Available: *http://www.w3.org/TR/ws-addr-core/*, May 2006.
41. GT 4.0 Reliable File Transfer (RFT) Service, accessed Jun 2008 [Online]. Available: *http://www.globus.org/toolkit/docs/4.0/data/rft/*.
42. Óscar David Robles, José Luis Bosque, Luis Pastor, and Angel Rodríguez. CBIR on grids. In R. Meersman, Z. Tari, et al., editors, *Proc. on OTM 2006*, volume 4276 of *Lecture Notes on Computer Science*, pages 1412–1421, Montpellier, France, November 2006. Springer Verlag.
43. A. Sánchez. *Autonomic high performance storage for grid environments based on long term prediction*. PhD thesis, DATSI Department, Universidad Politécnica de Madrid, 2008.
44. Timothy K. Shih, editor. *Distributed multimedia databases: techniques & applications*. Idea Group Publishing, Hershey, PA, EEUU, 2002.
45. Alan F. Smeaton. Techniques used and open challenges to the analysis, indexing and retrieval of digital video. *Information Systems*, 32(4):545–559, June 2007. doi:10.1016/j.is.2006.09.001.
46. Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.
47. Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on PAMI*, 22(12):1349–1380, December 2000.
48. W. Smith, I. Foster, and V. Taylor. Scheduling with advanced reservations. In *Proceedings of the 14th International Symposium on Parallel and Distributed Processing (IPDPS '00)*, pages 127–132, Washington, DC, USA, 2000. IEEE Computer Society.
49. S. Srakaew, N. Alexandridis, P. Piamsa Nga, and G. Blankenship. Content-based multimedia data retrieval on cluster system environment. In P. Sloot, M. Bubak, A. Hoekstra, and B. Hertzberger, editors, *High-Performance Computing and Networking. 7^{th} International Conference, HPCN Europe 1999*, pages 1235–1241. Springer Verlag, 1999.
50. Pablo Toharia, Óscar D. Robles, Ángel Rodríguez, and Luis Pastor. Sbd, search and rushes: Trecvid 2006 experiments at urjc. In *Proc. of the TRECVID 2006 Workshop*, pages 411–423, Gaithersburg, Md, November 2006.
51. Pablo Toharia, Óscar D. Robles, Ángel Rodríguez, and Luis Pastor. A study of zernike invariants for content-based image retrieval. In D. Mery and L. Rueda, editors, *Proceedings of the 2007 IEEE Pacific Rim Symposium on Image Video and Technology, PSIVT2007*, volume 4872 of *Lecture Notes on Computer Science*, pages 944–957, Santiago, Chile, December 2007. IEEE, Springer Verlag.
52. Pablo Toharia, Óscar David Robles, José Luis Bosque, and Angel Rodríguez. Video shot extraction on parallel architectures. In M. Guo et al., editors, *Proc. on International Symposium on Parallel and Distributed Processing and Applications (ISPA 2006)*, volume 4330 of *Lecture Notes on Computer Science*, pages 869–883, Sorrento, Italy, December 2006. Springer Verlag.
53. Pablo Toharia, Alberto Sánchez, José Luis Bosque, and Oscar David Robles. Efficient grid-based video storage and retrieval. In Robert Meersman and Zahir Tari, editors, *OTM Conferences (1)*, volume 5331 of *Lecture Notes in Computer Science*, pages 833–851. Springer, 2008.
54. J. Tordsson. Resource brokering for grid environments. Master's thesis, Umea University, June 2004.
55. S. Vazhkudai, S. Tuecke, and I. Foster. Replica Selection in the Globus Data Grid. In *Proceedings of the 1st IEEE International Symposium on Cluster Computing and the Grid (CCGRID '01)*, pages 106–113, Washington, DC, USA, May 15-18 2001. IEEE Computer Society.

56. Srikumar Venugopal, Rajkumar Buyya, and Lyle Winton. A grid service broker for scheduling distributed data-oriented applications on global grids. In *MGC '04: Proceedings of the 2nd workshop on Middleware for grid computing*, pages 75–80, New York, NY, USA, 2004. ACM.
57. Srikumar Venugopal, Krishna Nadiminti, Hussein Gibbins, and Rajkumar Buyya. Designing a resource broker for heterogeneous grids. *Software: Practice & Experience*, 38(8):793–825, 2008.
58. Peter Wilkins, Paul Ferguson, and Alan F. Smeaton. Using score distributions for query-time fusion in multimedia retrieval. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 51–60, New York, NY, USA, 2006. ACM Press.
59. GT 4.0 WS Grid Resource Allocation Management, accessed Jun 2008 [Online]. Available: *http://www.globus.org/toolkit/docs/4.0/execution/wsgram/*.
60. B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Nineteenth International Conference on Data Engineering (ICDE'03)*, pages 49–62, 2003.
61. Minerva M. Yeung, Boon-Lock Yeo, and Charles A. Bouman, editors. *Proceedings of the SPIE/EI'2000 Symposium on Storage and Retrieval for Media Databases*, volume 3972. SPIE, January 2000. ISBN 0-8194-3590-2.
62. F. Zernike. Beugungstheorie des schneidenverfahrens und seiner verbesserten form, der phasenkontrastmethode (Diffraction theory of the cut procedure and its improved form, the phase contrast method). *Physica*, 1:689–704, 1934.