A Distributed Architecture for Multi-dimensional Indexing and Data Retrieval in Grid Environments *

Athanasia Asiki, Katerina Doka, Ioannis Konstantinou, Antonis Zissimos
and Nectarios Koziris

National Technical University of Athens
School of Electrical and Computer Engineering
Computing Systems Laboratory
Zografou Campus, Zografou 15773, Greece
email: [nasia, katerina, ikons, azisi, nkoziris]@cslab.ece.ntua.gr

Abstract

In this paper, we describe a service-oriented architecture of a generic middleware platform, which provides the required services for data management in a distributed environment. Our design introduces concepts from Peer-to-Peer computing in order to provide a scalable and reliable infrastructure for storage, search and retrieval of annotated content. To ensure fast searching in the distributed repositories of a Virtual Organization, our system incorporates a multidimensional indexing scheme, which serves the need for supporting both point and range queries over a set of metadata attributes. Finally, multimedia file transfers are conducted using GridTorrent, a grid-enabled, Peer-to-Peer mechanism that allows the aggregate data transfer throughput to scale and effectively copes with flash crowds.

1 Introduction

Due to the explosion of network technologies and the astounding growth in the performance of computing systems, there is an increasing tendency to apply techniques learned in high-performance and cluster-based computing in a completely distributed environment consisting of computing resources communicating over the Internet. Apart from achieving maximum utilization of computational idle cycles, it is very challenging to set a global-scale data management scheme, namely to provide the required protocols and algorithms, so as files to be shared, searched, located, transferred and replicated among geographically distributed resources.

A form of distributed computing focusing mainly in sharing and manipulating large datasets is *Peer-to-Peer (hence P2P) computing*, which is gaining an increasing interest by the academic and the large Internet community. P2P systems are mainly used for content sharing. Many P2P networks exist for file

*The described work is partly supported by the European Commission through the FP6 IST Framework Programme (FP6-34363 [1]).

publishing and discovery among thousand of users and their number is growing every day. However, the distributed nature of P2P systems and the lack of centralized structures poses difficulties in locating the required content. Therefore, much effort has been given by the research community for the development of indexing methods to enable efficient search mechanisms.

A different approach to the evolution of distributed computing is the *Service Oriented Architecture* (SOA) applied pragmatically in Grid computing. A Grid system is a wide-area, large-scale system, in which remotely located and heterogeneous resources are integrated under a common software architecture. The basic difference from P2P computing is the existence of explicitly defined rules and policies to enable flexible, secure and coordinated resource sharing among dynamic virtual collections of users, named *Virtual Organizations* (hence VOs). Although this philosophy seems contradictory to file sharing in P2P overlays, there is a growing trend of introducing P2P techniques and algorithms into Data Grid architectures, in order to enhance existent basic services, like the Data Transfer service, responsible for moving files among nodes (e.g. GridFTP), or the Replica Location service, charged with keeping track of the physical locations of files.

The content to be shared in our system is mainly annotated multimedia content. Multimedia content bombards our daily life and is produced by the majority of scientific and business applications. Large amounts of audiovisual data are becoming available continuously on the World Wide Web, in broadcast data streams, in personal and business databases. It is evident, that multimedia data can contain a lot of concentrated information and display it in a more descriptive way than plain text, but its value depends on the existence of efficient mechanisms for discovery, access and management. Storing and retrieving multimedia files becomes difficult due to their sheer volume. Searching multimedia content is also challenging, because it is not feasible to search the content itself. Therefore, each multimedia file should be described with metadata and the effectiveness of the search mechanism relies on its ability to find and process these annotations.

In this paper, we present a service-oriented middleware architecture for data management in Grid environments. Our design is completely distributed, requiring no form of centralized structures to coordinate interactions among different services and supervise their execution. The above-mentioned middleware provides services for efficient data search, discovery and transfer of annotated content. Our interest in the proposed architecture is based on the belief that it would provide users manipulating annotated content with the necessary services to build a promising Grid infrastructure and integrate with other systems and existing middlewares.

The described architecture accommodates large-scale, distributed facilities for handling massively produced content. Searches for the stored content are supported in two different levels. At the first level, a user can perform advanced searches in the annotations based on a predefined metadata schema and stored in a P2P overlay. The search performance in this overlay is improved by multidimensional indexing. At the second level, a data item can be searched according to its unique identifier, which is stored in a distributed catalogue, containing mappings of filenames to physical locations, where data is actually stored.

2 Related Work

The work described in this paper exploits *Peer-to-Peer* techniques in order to extend the capabilities of the *Data Grid* architecture. The Peer-to-Peer approach provides a scalable alternative solution to the conventional server - client approach, where the server usually constitutes a vulnerable point of delays and failures. Depending on their structure, Peer-to-Peer systems can be divided into three categories, namely structured, unstructured and hybrid. In a structured overlay, a *Distributed Hash Table* (DHT) is constructed, where (key, value) pairs are stored. A DHT-based system, such as Kademlia [2] guarantees that, if a key exists in the overlay, the lookup algorithm will find it with a lookup cost bounded to the logarithm of the search space, namely the number of nodes participating in the overlay.

The *Data Grid* refers to a wide-area distributed infrastructure of heterogeneous resources capable of managing immense amount of data. The core services for accessing heterogeneous storage resources, storing, transferring and searching large datasets are described in [3]. In a Data Grid, it is a common practice the distribution of multiple copies of a file, called *replicas*, among resources. The physical instances of files are located by the *Replica Location Service*, which interacts with a Replica Catalogue containing mappings between *logical filenames* (LFNs) and *physical filenames* (PFNs). Efforts on distributing the catalogue, in order to improve the scalability and resilience of the system, resulted to the most widespread solution currently deployed on the Grid for replica management named *Giggle* (GIGA-scale Global Location Engine) *Framework* [4], [5].

Another fundamental building block of the Data Grid architecture is the data transfer mechanism among storage nodes. The established protocol is *GridFTP*, a protocol defined by the Global Grid Forum and adopted by the majority of the existing middlewares. GridFTP extends the standard FTP protocol including features like *Grid Security infrastructure* (GSI) [6] and third-party control and data channel. A more distributed approach of the GridFTP service attracted the attention of the Grid community leading to the Globus Stripped GridFTP protocol [7], included in the current release of the Globus Toolkit 4 [8].

The existence of Metadata services in the context of the Grid philosophy able to interact with other data services plays important role in handling data sets. In [9], Deelman et al. emphasize in the need of services responsible for handling metadata descriptions of data objects. Their claim is being justified by real use cases from the scientific community. They introduce a Metadata Catalogue Service (MCS) implemented on top of a database.

3 Architecture

The proposed architecture deals with data management in a distributed environment consisting of resources belonging to different Virtual Organizations. A main characteristic of this environment is the heterogeneity of nodes in terms of computational power, storage capacity and bandwidth. It has been taken into account that resources with different features, for example laptops, desktop computers, dedicated servers or even mobile phones, participate in this platform. Moreover, we assume that it is highly possible some resources not to remain connected to the system for a long period of time. Therefore, the design confronts with nodes arrivals and departures so as to provide a robust overlay.

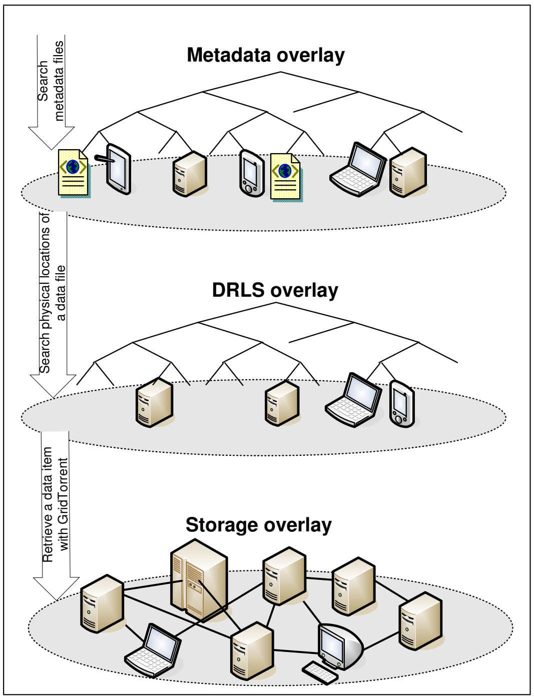
Metadata overlay
Search metadata files
DRLS overlay
Search physical locations of a data file
Storage overlay
Retrieve a data item with GridTorrent

Fig. 1: Overview of the proposed architecture

The specific architecture comprises of three different overlays, namely the *Metadata overlay*, the *DRLS overlay* and the *Storage overlay*. The Metadata and the DRLS overlays are implemented as extensions to the *Kademlia DHT* [2], a distributed hash table for decentralized peer-to-peer computer networks. Each node of the system can participate in various overlays according to the services it can host. The links among nodes of each overlay are between nodes with close IDs in the ID space and do not require that these nodes are physically close as well. The idea behind the multiple overlays is the design of an extensible platform, which can integrate and interact with other Grid services and favors the development of Grid applications. The three overlays of our architecture are shown in Figure 1. As it can be noticed in this Figure, the nodes in the Metadata and the DRLS overlay are treated as leaves of the Kademlia binary tree. Each node in the Storage overlay is connected to its known neighbors.

In our design, a *data file* contains the actual content to be stored in the Storage overlay. The sharing of resources in this overlay occurs among users of the same Virtual Organization. The consequence is that a *data file* can be stored to or retrieved from a storage node, only if the user is granted the appropriate permissions. The nodes in the Storage overlay act as file servers and should provide the corresponding services persistently.

Each *data file* is described by attributes of a predefined metadata schema, which are included in a *metadata file*. The search mechanism in the Metadata overlay aims to manipulate these metadata files efficiently, in order to return data items fulfilling the user's search criteria. The metadata file includes also the LFN of the corresponding data file. This LFN is used as a key in our distributed catalogue implemented in the *Distributed Replica Location Service* (hence DRLS) overlay. In this overlay, mappings of LFNs to the physical locations of data files represented by PFNs are stored. In order to adjust the functionality of the DRLS to the requirements of a Grid environment, it has been assumed that one DRLS overlay is deployed per VO. Each key of the DHT corresponds to the hash of the LFN of the *data file* and it is the unique identifier complementing all data operations. A value for a key is actually a list containing the physical locations of replicas (PFNs) for a given identifier.

A user may perform *store* and *search* operations in the proposed platform as follows:

- When a user initializes the upload procedure, the *data file* is assigned with a unique identifier, namely a LFN. The LFN is also included in the *metadata file* among the rest of the description provided by the user. The LFN is the link between the actual data and their descriptions. There is no requirement for the *data files* and the *metadata files* to be stored in the same node. The *data file* is uploaded in the Storage overlay using the GridTorrent mechanism described in Subsection 3.3. If the user's node participates in the Storage overlay, the file is stored locally. Otherwise, the data file is uploaded to another known storage node. The physical location(s) of the file is (are) inserted in the DRLS overlay.
- A user is able to query for files satisfying its criteria with the search mechanism provided by the Metadata overlay. The search mechanism returns the relevant *metadata files*. Before the download of a specific *data file* takes place, the data transfer mechanism queries the DRLS overlay for its PFNs. Finally, the GridTorrent protocol downloads the file by exploiting the collaborative sharing properties of the inheriting BitTorrent protocol, in order to boost aggregate performance. The flow of the search procedure among the different overlays is also shown by the arrows in Figure 1.

3.1 The Metadata overlay

Our approach to support multi-dimensional queries exploits the fundamental property of *Space Filling Curves* (SFCs) to continuously map a compact interval to a d-dimensional space and vice versa. SFCs preserve locality, so that points in the 1-dimensional space are mapped to close points in the d-dimensional space. The SFCs are utilized in the Metadata overlay in the partition strategy of metadata files among peers in the Kademlia-based Metadata overlay. The search mechanism supports queries on the decided metadata schema. The most important attributes of this schema are indexed.

The set of d attributes to be indexed forms a d-dimensional space. Each point in this space represents a combination of values of indexed attributes. The points of the d-dimensional space are mapped down to a single dimension by a SFC, such as the Hilbert curve or the Z-curve. The result is the partitioning of the d-dimensional space into 2^{kd} cells, which in turn are mapped through the SFC to 2^{kd} points of a single dimension. The values of the single dimension represent the keyset of the DHT overlay. Each key is kd bits long and each node in the overlay manages data in one contiguous range of the SFC.

Since the attributes of the metadata file are considered to form a multidimensional space, the answer to a query corresponds to points of either one or more intervals of the SFC. In case of a point query, the matching intervals are downgraded into a point. The processing of a query consists of two consecutive phases. In the first phase, the clusters of the SFC answering the query are determined. In the next phase, lookup operations for all the cluster(s) start.

3.2 The Distributed Replica Location Service

The DRLS overlay implements a *Distributed Replica Location Service* [12] with a DHT by correlating its inherent (key, value) pairs to (LFN, PFNs) mappings. Every data file is assigned with a LFN, which is considered to identify the file uniquely in the system. The LFN is included in the metadata file as well and links the metadata description to the actual data. In the DRLS overlay, the LFNs can be used as identifiers by the overlay network to route lookup queries to corresponding PFN lists.

In the DRLS overlay, every lookup for a LFN always queries all nodes responsible for a specific (key value) pair, compare the results based on some predefined version vector (indicating the latest update of the value) and propagate the changes to the nodes it has found responsible for storage but not yet up-to-date with the latest value. The process for locating data items does not stop when the first value is returned, but continues until all available versions of the pair are present at the initiator. The querying node then decides which version to keep and sends corresponding store messages back to the peers that seem to hold older or invalid values. Updates to (LFN, PFNs) could therefore be implemented through the predefined set operation, as version checking would also be done by nodes receiving store commands. The latter should check their local storage repositories for an already present identifier, and if there is a conflict, keep the latest version of the two values in hand. The version of a (key, value) pair is determined by a timestamp indicator.

3.3 The GridTorrent data transfer mechanism

The purpose of this layer is to provide a data transfer mechanism that effectively deals with large file uploads and downloads, even when numerous requests rely on a single data source, maximizing bandwidth utilization. The proposed solution, *GridTorrent* [11] constitutes a decentralized approach, that, unlike GridFTP, takes advantage of multiple replicas to boost aggregate transfer throughput.

GridTorrent is an implementation of the popular BitTorrent protocol [13], [14] designed to interface and integrate with well-defined and deployed Data Grid components and protocols (e.g. GridFTP, RLS). In short, GridTorrent works as follows: A request to GridTorrent for a file triggers a query to the Distributed Replica Location Service, which is repeated periodically, in order to detect any changes in the locations of file replicas or of any joins or departures of nodes, either GridFTP servers or GridTorrent peers. Upon receiving the list of peers, GridTorrent acts according to the protocol prefix of the PFN. If it concerns a GridTorrent client, the two involved peers initiate communication by exchanging the BitTorrent *bit field* message, informing each other of the pieces they possess. Furthermore, each time a peer downloads a piece, it sends a *have* message notifying all peers connected to it of its new acquisition. In order to download data from another GridTorrent client, the peer issues a *request* message for blocks. Blocks are parts of a piece, referenced by the piece index, a zero-based byte offset within the piece and their length. Having information about the available pieces, GridTorrent starts downloading pieces following a rarest-first policy. In case of a GridFTP server, the peer does not need to exchange bit field messages. As for the downloading technique, the client issues a GridFTP partial get message for the data within the specific block it intends to download.

4 Concluding remarks

This paper addresses the subject of efficient and reliable annotated content sharing and management in a Grid environment, where heterogeneous resources cooperate. In this context, we proposed a service-oriented middleware architecture, that provides store, search and retrieve primitives for manipulation of data files. We introduced the idea of separate DHT overlays for metadata and data management, thus avoiding the use of centralized entities. We strongly believe that the proposed architecture, incorporating various P2P techniques, will provide a robust and scalable infrastructure for storing annotated data and searching over a large set of metadata attributes.

The implementation of the described platform is currently at a prototype status. The functionality of basic components has been predetermined and the corresponding interfaces have been deployed. The proposed services for data management are developed as *Grid services* according to the Open Grid Service Architecture (OGSA) [10], making use of the libraries of Globus Toolkit 4 [8].

References

1. GRid enabled access to rich mEDIA content (GREDIA), http://www.gredia.eu/
2. Maymounkov, P., Mazieres, D.: Kademlia: A peer-to-peer information system based on the XOR metric *in Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS02),* Cambridge, USA,2002.
3. Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., Tuecke, S.: The data grid: Towards an architecture for the distributed management and analysis of large scale scientific datasets, *Journal of Network and Computer Applications,* vol. 23, no. 3, pp. 187-200, 2000.
4. Chervenak, A., Deelman, E., Foster, I., Hoschek, W., Iamnitchi, A., Kesselman, C., Ripeanu, M., Schwartzkopf, B., Stockinger, H., Tierney, B.: Giggle: a framework for constructing scalable replica location services", *in Proc. of the 2002 ACM/IEEE conference on Supercomputing,* Nagoya, Japan, pp. 1-17, jan, 2002.
5. Chervenak, A., Palavalli, N., Bharathi, S., Kesselman, Schwartzkopf, B., Stockinger, H., Tierney, B.: Performance and Scalability of a replica location service", *in Proc. of the 13th IEEE International Symposioum on High Performance Distributed Computing Conference (HPDC-13),* Honolulu, jun, 2004.
6. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A security architecture for computational grids, *in Proc. of the 5th ACM conference on Computer and communications security,* pp. 83-92, 1998.
7. Allcock, W., Bresnahan, J., Kettimuthu, R., Link, M.: The Globus Striped GridFTP Framework and Server, *in Proc. of the 2005 ACM/IEEE conference on Supercomputing,* 2005.
8. Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented, *Journal of Computer Science and Technology,* vol. 21, no. 4, pp. 513-520, 2006.
9. Deelman, E., Singh, G., Atkinson, M.P., Chervenak, A., Chue Hong, N.P., Kesselman, C., Patil, S., Pearlman, L., Mei-Hui Su: Grid-based metadata services, *in Proc. of 16th Conference on Scientific and Statistical Database Management,* pp. 393-402, Los Angeles, CA, USA, jun, 2004.
10. Foster, I., Kishimoto, H., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Maciel, F., Siebenlist, F., Subramaniam, R.,Treadwell, J., Von Reich, J.: The Open Grid Services Architecture, Version 1.0, Informational Document,Global Grid Forum (GGF), jan, 2005.
11. Zissimos, A., Doka, K., Chazapis, A., Koziris, N.: GridTorrent: Optimizing data transfers in the Grid with collaborative sharing, *in Proc. of the 11th Panhellenic Conference on Informatics (PCI2007),* Patras, Greece, may, 2007.
12. Chazapis, A.,Zissimos, A., Koziris, N.: A peer-to-peer replica management service for high-throughput Grids, *in Proc. of the 2005 International Conference on Parallel Processing (ICPP05),* Oslo, Norway, jun, 2005.
13. BitTorrent.org, http://www.bittorrent.org/index.html.
14. Cohen, B.: Incentives Build Robustness in BitTorrent, *in Workshop on Economics of Peer-to-Peer Systems,* Berkeley, USA, jun, 2003.