GLORIA is funded by the European Union 7th Framework Programme (FP7/2007-2013) under grant agreement n° 283783

GLObal Robotic telescopes Intelligent Array for e-Science

Peer-to-peer data storage software
Specification and source code

CODE: DEL-050

VERSION: 01

DATE: November 28th, 2012

Authors:	Urko SERRANO (UPM)
Collaborators:	Fernando SERENA (UPM)
Revised by:	Fernando IBÁÑEZ (UPM)
Approved by:	Francisco Manuel SÁNCHEZ (UPM)

Distribution List:

Name	Affiliation	Date
Urko SERRANO	UPM	November 28th, 2012
Fernando SERENA	UPM	November 28th, 2012
Fernando IBÁÑEZ (UPM)	UPM	November 28th, 2012
Francisco Manuel SÁNCHEZ	UPM	November 28th, 2012

Change Control

Issue	Date	Section	Page	Change Description
01.A	28/11/2012	All	All	Creation

Reference Documents

Nº	Document Name	Code	Version
R.1	Peer-to-peer live broadcasting software: spec and source code update	DEL-9.6	01.C

Index

Illustration Index

Index of Tables

1. Overview

This document covers the design and implementation of a distributed data storage infrastructure that provides a virtual image repository for media content generated within the GLORIA system. All the content has been elaborated by the partners of the GLORIA project, which means "GLObal Robotic-telescopes Intelligent Array for e-Science". This project is funded by the European Union 7th Framework Programme FP7/2007-2013) under grant agreement no. 283783.

1.1. Purpose

The main topic of this document is the specification of a cost effective storage platform that enables the GLORIA community to store and access media content such as astronomical images in a large scale system. The design is made as a robust and resilient infrastructure, enabling the GLORIA community to access the media resources at any time. The platform is organized into two main software solutions: cloud-based and peer-to-peer infrastructure.

1.2. Scope

This document is primarily aimed for developers of the GLORIA system. It includes the design and the deployment of the distributed storage platform for the GLORIA community.

1.3. Outline

The outline of this document proceeds as follows. The *Introduction* section summarizes the concept of the storage platform, along with a brief explanation of both cloud-based and peer-to-peer solutions. The *Cloud-based solution* section presents the concept of a scale-out storage built-in in the cloud and accessed by an intuitive API. The *Peer-to-peer solution* section defines the collaborative transport protocol available for the GLORIA community. The *Platform* section encompasses both solutions into an integrated infrastructure.

2. Introduction

The GLORIA network is mainly form by a set of robotic telescopes. Within this network, each robotic telescope dumps large amounts of media content periodically. This media content such as images needs to be stored for future access with high availability from the GLORIA system.

To do so, a distributed storage platform is designed. The main aim is to provide a resilient platform that provides a media repository where media content is stored persistently and accessible for the GLORIA users, regardless of the overhead and current performance of the platform. This platform is scalable and handles a considerable number of GLORIA user concurrently.

The platform is divided into two separate solutions:

- The cloud-based solution is a cloud computing approach where a pool of servers manage the media repository. These servers are seen as a unique entry point from the GLORIA user's perspective. The solution provides a common framework to deliver the media content as a whole.
- The peer-to-peer solution consists of a network where GLORIA users can contribute by donating their resources. The result is a network of GLORIA users who store the media content and relief the overhead of the cloud-based solution. Adding more GLORIA users in this network implies a more flexible solution to fulfil the requirements of the distributed storage platform.

The two solutions are combined into a single distributed storage platform. The design will benefit the features of both approaches, having a robust and stable platform while supporting a flexible system in critical scenarios with high demand of requests.

The platform provides an intuitive interface for third-party applications that want to access or store the media content. As part of the features of this platform, applications running in the GLORIA system will get access to this interface and they will retrieve and manage their media resources. This document details the design of the platform and the requirements for the implementation.

3. Cloud-based solution

This solution provides a cost effective and scale-out storage to manage media content. It is designed to be a fully distributed repository with no centralized coordinator to avoid a single point of failure and a possible bottleneck in the system. The design is based on web standards and it is open for collaboration from the developer community.

The standard used for this cloud-based solution is managed by the OpenStack project. OpenStack is a global collaboration of developers implementing an ubiquitous open source cloud computing platform for public and corporate clouds. The project aims to deliver solutions for heterogeneous clouds by simplifying the interface and implementation, while providing a massively scalable infrastructure.

The technology consists of different interrelated projects where the storage alternative has been chosen for this cloud-based solution. In particular, the solution is based on object storage and it is suitable for storing data such as images from the GLORIA system.

OpenStack object storage provides the following capabilities:

- Distributed storage system for static content (ex. images, videos, backups and archives).
- High availability and good scalability of the media content.
- Flexible solution using clusters or standardized servers.
- Data replication and integrity across the pool of servers.
- No centralized coordination point.

Technically, the OpenStack features below correspond to the characteristics needed for the cloud-based solution:

Feature	Description
Unlimited storage	Highly scalable read and write access from the storage system.
Real-time monitoring	Intuitive tool to know what users are requesting the media content.
Direct object access	Control panel-like feature to enable direct browser access to content.
Expiring objects	GLORIA users could set an expiration time for specific media content.
Built-in user management	Interface for user administration and container management.
Failure alerts	Automatic drive failure detection to prevent data corruption.
Decentralized database	Increases performance and avoids single point of failures.

Table 3.1: OpenStack features

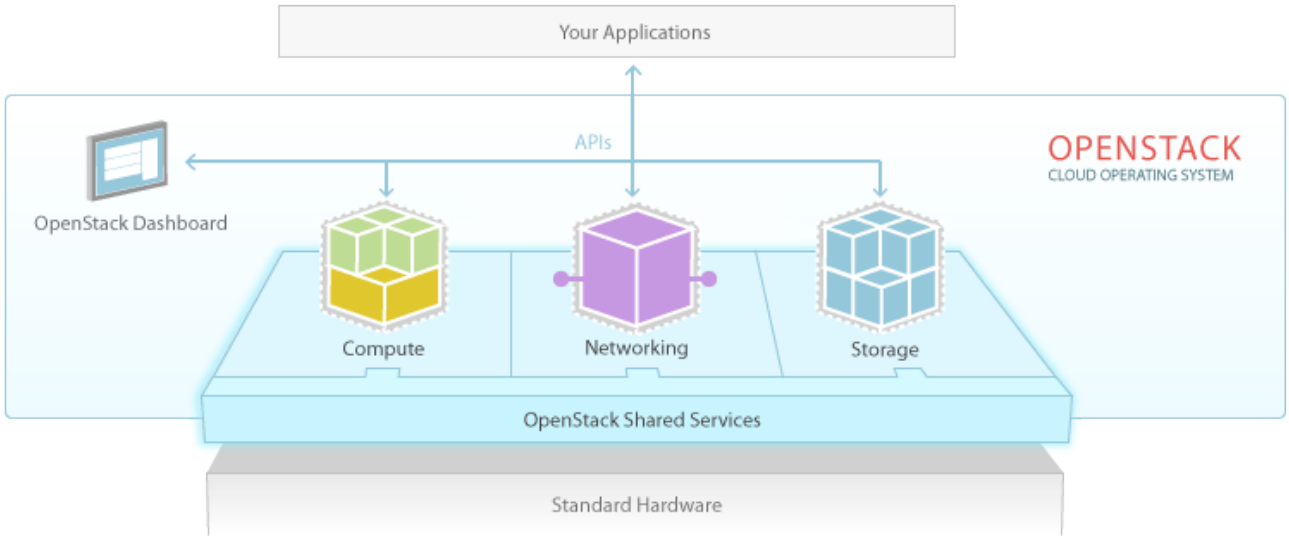

Figure 1: OpenStack cloud operating system

4. Peer-to-peer solution

This solution aims for providing a flexible storage repository built upon the resources donated by the GLORIA users. The implementation will register each GLORIA user as a peer in the network in order to store and retrieve media content. The objective is to reuse the current solutions being implemented in GLORIA to enhance the compatibility with other software components.

The design of the peer-to-peer solution shares the same features as the solution described in DEL-9.6. Combining the same infrastructure, GLORIA users can also contribute not only in the media storage but also during the live webcast of the astronomical events scheduled in the GLORIA project.

The infrastructure of the peer-to-peer network is based on the PPSPP transport protocol. Traditional Internet protocols relies on a communication between a client and a remote server. The general scenario is a client that requests a server to fetch specific data. This communication provides the abstraction of conversation between the client and the server.

This approach does not scale well in a large-scale scenario with thousands of concurrent requests to the system. Each user makes at least one request to the server that has to be handled internally. If multiple concurrent requests are being processed, the memory and computing resources of the server can be reduced drastically. A possible result could be a denial of service failure in the system that blocks GLORIA users to access the media content.

PPSPP is a peer-to-peer based transport protocol for content dissemination. The protocol is defined by a network of users who participate by forwarding the content to each other via a mesh-like topology. PPSPP is intended to be a built-in cross-platform protocol in the TCP/IP stack, supporting all major operating systems and web browsers.

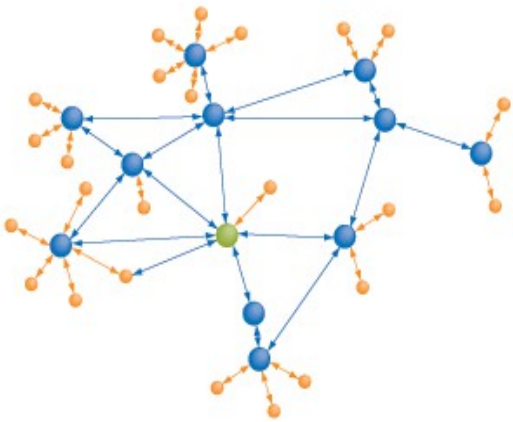

Figure 2: Network mesh topology

Each median file stored in the system has a unique hash id in the overlay network. Users only need this hash id to receive the data from any available source, while data integrity is checked with Merkle hash trees. The protocol abstracts away the complex underlying connections among network entities to provide an intuitive API.

The transport protocol relies on the idea that bandwidth and memory resources have to be used in a proactive way. There is no benefit in keeping these resources unoccupied. Therefore, the implementation is focused on a lightweight footprint, automatic disk space management and non-intrusive congestion protocol.

PPSPP features short initial playback delay and scalability. It can use different mechanisms to prevent users not sharing and only downloading the content (free riders). The transport protocol also works with different centralized or distributed peer discovery schemes, and it offers NAT traversal techniques for limited users.

4.1. Message passing

The basic unit of communication in the network is the message. Multiple messages are combined into a single datagram for transmission. The following list represents all the message types available:

Message	Description
HANDSHAKE	Contains the initial information of a peer such as protocol options and network id.
HAVE	Includes which chunks of content a peer has available for download.
ACK	Contains the specification of a successfully checked chunk of content.
DATA	Contains the chunk data and its id.
INTEGRITY	Includes the information to verify the integrity of a chunk of content.
REQUEST	Contains the specification of chunks that a peer wants to download.
CANCEL	Includes a list of chunks that the peer no longer wants to request.

Table 4.1: Handshake protocol messages

4.2. Peer registration

Following the PPSPP handshake protocol, user A wants to download content from the network. First, user A needs to have the hash id of the content and optionally, a list of trackers if there is no decentralized tracking mechanism. Then, user A communicates with a tracker (1) and receives a list of peers already in the network such as user B, C, and D. When user A registers itself in the tracker, it becomes a network peer ready to share content.

User A parses the user list and sends a HANDSHAKE message to all the new discovered users (2). This message includes information regarding protocol version and network id to verify that user A has joined the correct network. Users B and C reply back with a HANDSHAKE message and at least one HAVE message (3). The HAVE messages conveys the chunk availability of both users respectively. User D sends only a HANDSHAKE message in order to not choking A.

4.3. Peer data exchange

Following the handshake protocol, user A sends a REQUEST message to user B and C in response to their replies (4). Both messages are disjunct sets of chunks that user A wants to download. Users B and C reply (5) with three messages: INTEGRITY, DATA and HAVE. The INTEGRITY message conveys the hash ids of all the chunks sent in the DATA message in order to check if they are correct. The HAVE message is used by user A to update the chunk availability of users B and C.

When user A processes the messages, it sends back HAVE messages to users B and C with the information of received chunks from all the users (6). In addition, user A sends an ACK message for the received chunks and a REQUEST message for the ones to download. ACK messages are only sent in unreliable transport protocols.

User C discovers that user A received a chunk that user C does not have yet, so its chunk id will be included in a REQUEST message. User D does not send HAVE messages to user A until it will unchoke user A. In this case, user D sends a datagram with HAVE messages to user A to inform it about its chunk availability (7). If user B and C decide to choke A, they will stop sending HAVE and DATA messages.

4.4. Peer deregistration

Depending on the underlying transport protocol, users can leave the network by sending leave messages or just stop replying all messages. If the user leaves normally, it should deregister itself from the original tracker.

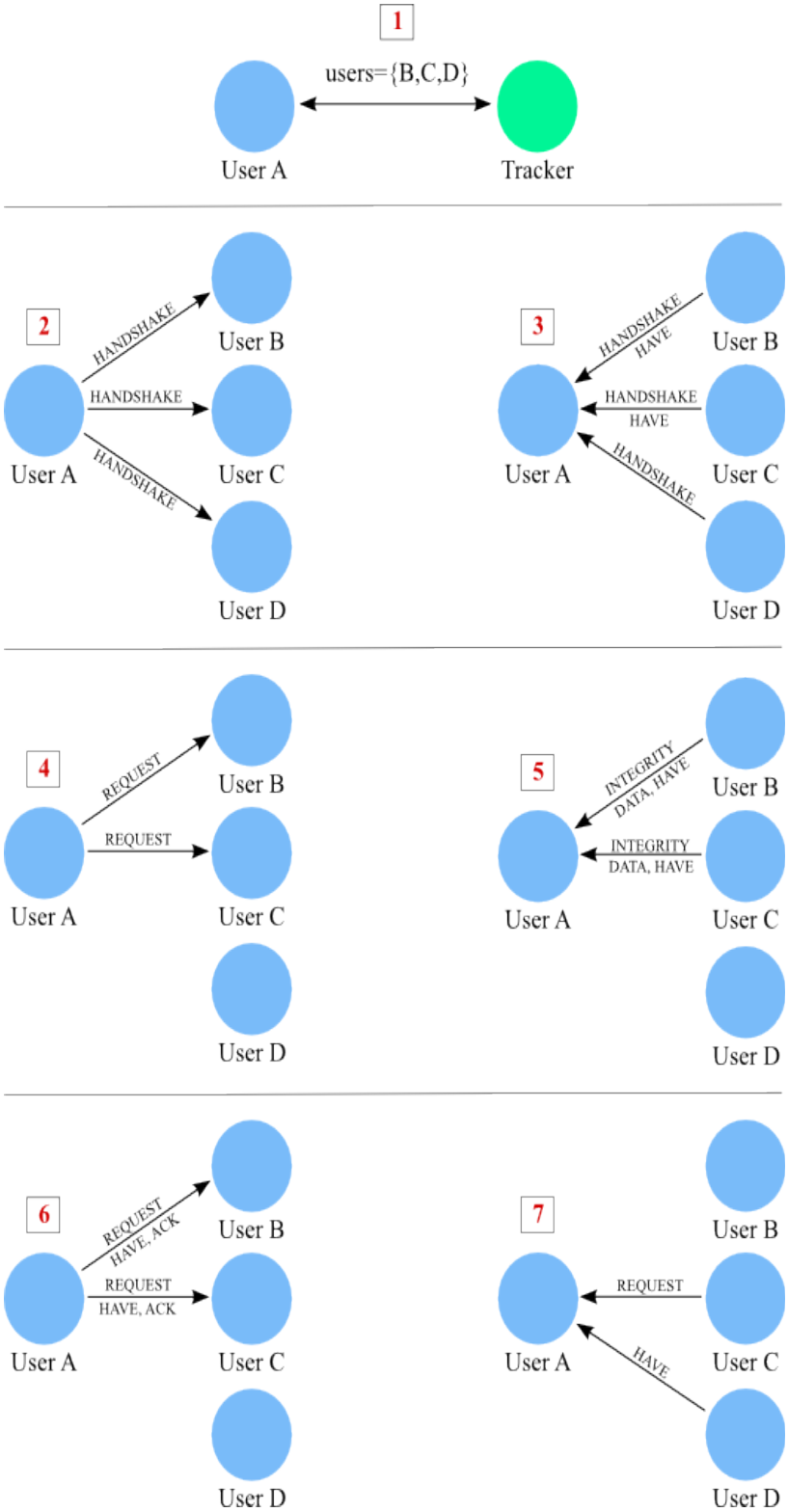

Figure 3: PPSPP handshake protocol

4.5. Merkle hash tree

All content managed by PPSPP have a unique hash that is the root in a Merkle hash tree calculated from the content. The Merkle hash tree of a content is divided into N chunks of data with their corresponding hash ids.

A binary tree is created with enough height that the lowest level in the tree has enough nodes to hold all chunk hashes in the set. The figure shows a tree of a content with 7 chunks of data. The leaves of the tree represents sorted chunks of data starting from the left-most leaf. Remaining leaves not covered by the tree will be set as zero.

The hash values of upper levels in the tree are calculated by concatenating the hash value of the two children in a left-right order and generating the hash of that aggregate. This algorithm ends in a hash value for the root node called root hash.

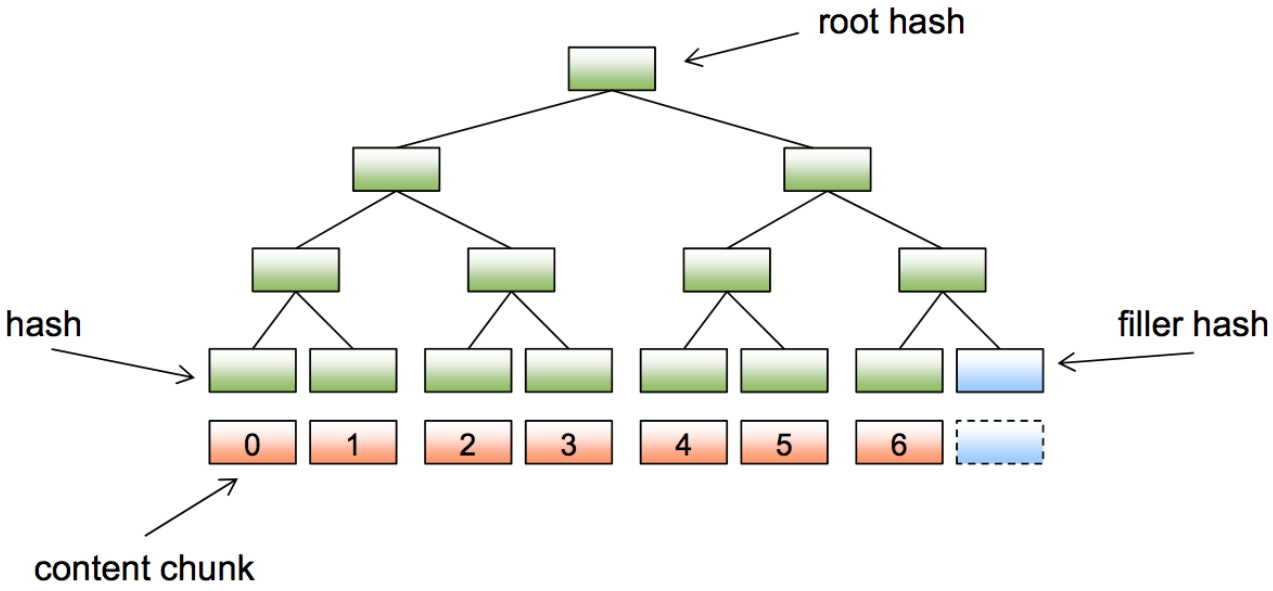

Figure 4: Merkle hash tree

5. Platform

The design of the distributed storage platform is a combination of both cloud-based and peer-to-peer solutions. For the applications of the GLORIA system and the GLORIA users, the platform behaves as a single entry point to access the media content. The integration of the solutions is done internally and transparently to the final user.

The main component of the platform is the cloud-based features provided by the OpenStack project. The cloud-based solution correponds to the core of the infrastructure, since the pool of servers using OpenStack provides a reliable and stable service for the media repository.

On top of the OpenStack network, a built-in framework will act as the bridge between the cloud-based and the peer-to-peer solution. The framework will run in specific nodes named entry servers that will expose an interface to allow access to the media content from the peer-to-peer network. In this scenario, the GLORIA users acting as peers in the peer-to-peer network will collect the media content through this interface.

On the other hand, external users will access the content via different services included in the cloud-based solution. For GLORIA applications, a web service will be available to perform operations in the distributed storage platform. Other services such as a web browser applications could be used to get access to the media content for online users.

GLORIA Partners

UPM
Universidad Politécnica de Madrid
SPAIN

AUAV
Astronomical Institute, Academy of Sciences of the Czech Republic
CZECH REPUBLIC

CSIC
Consejo Superior de Investigaciones Científicas
SPAIN

INAT & CVUT
Czech Technical University in Prague
CZECH REPUBLIC

IP-ASCR
Institute of Physics of the Academy of Sciences of the Czech Republic
CZECH REPUBLIC

IAC
Instituto de Astrofísica de Canarias
SPAIN

INAF
Istituto Nazionale di Astrofisica
ITALY

SAO
Special Astrophysical Observatory of Russian Academy of Sciences
RUSSIA

UCDNUID
University College Dublin
IRELAND

UC
University of Chile
CHILE

UMA
University of Málaga
SPAIN

UOXF
University of Oxford
UNITED KINGDOM

UNIWARSAW
Uniwersytet Warszawski
POLAND