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 web hosting software
specification and source code

CODE:	Ask it to the Quality Team
VERSION:	01.B
DATE:	July 31st, 2013

Authors:	Urko SERRANO (UPM)
Collaborators:	Luciano NICASTRO (INAF)
Revised by:	Raquel CEDAZO (UPM)
Approved by:	Francisco Manuel SANCHEZ (UPM)

Distribution List:

Name	Affiliation	Date
Urko SERRANO	UPM	July 17th, 2013
Luciano NICASTRO	INAF	July 17th, 2013
Raquel CEDAZO	UPM	July 17th, 2013
Francisco Manuel SANCHEZ	UPM	July 17th, 2013

Change Control

Issue	Date	Section	Page	Change Description
01.A	08/2/2013	All	All	Creation
01.B	31/07/2013	All	All	Major modifications in all sections

Reference Documents

Nº	Document Name	Code	Version

Overview

This document describes the solution to implement a P2P web hosting infrastructure capable of serving a website for unlimited number of users. 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.

Back-end

Web server

Objective

We aim at identifying the best available web server(s) best suited at implementing the two main objectives of WP9: 1. building a P2P distributed archive system for images and data produced by GLORIA and 2. implementing a web hosting software able to broadcast live astronomical events on the web using only the resources made available by the GLORIA subscribers worldwide.

The typical tree structure of a P2P server calls for an event-driven (asynchronous) architecture. In fact the content of a web page is "pushed" by nearby nodes to the underlying nodes in a hierarchically defined structure. Only servers implementing modern and effective event-driven architecture with a lightweight footprint will be considered. A large variety of devices should be considered as potential hosts, from multi-core server-rack machines to portable devices.

In terms of performance, web servers are benchmarked for figures like number of requests per second they can serve as a function of the number of simultaneous connections, or the RAM/CPU usage again as a function of the number of clients. However, some performance figures are not web server dependent but only reflect the limitation of the infrastructure they are connected to. A typical one is the internet bandwidth that a single device has access to.

Alternatives

Only cross-platform open source servers will be considered. The following candidate servers have been investigated:

- Nginx (nginx.org)
- Lighttpd (lighttpd.net)
- Cherokee (cherokee-project.com)
- Apache (apache.org)
- G-Wan (gwan.com)
- Mongoose

Mongoose has very small footprint and embeddable code could be useful in portable devices. On the other hand, G-Wan is very well suited to run on multi-core machines. It outperforms the others servers on these machines and then it is a good candidate for the high-end nodes of the GLORIA network.

Performance

Several benchmarks exist in order to evaluate servers performance and reliability. Custom version can also be easily implemented in order to test only particular capabilities or features of a given web server. In fact, in order to evaluate the performance of the GLORIA specific P2P architecture, we must put priorities on the various figures.

The reference architecture we must keep in mind is that with a large number of nodes involved in the network each of which is possibly serving a limited number of clients. For example, it is not that important to consider a large number of concurrent connections because it is likely that the device data transfer bandwidth would limit the connections anyway. Of course, some more robust nodes must be considered as part of the network too. In these cases the relevance of the various figures would be considered differently.

Among the possible test program available, we considered:

- ApacheBench (http://httpd.apache.org/docs/2.2/programs/ab.html)
- Weighttp (https://github.com/lighttpd/weighttp)
- Siege (http://www.joedog.org/siege-home/)

Conclusions

With in mind the mentioned parameters for performance evaluation, we investigated publicly available benchmarks for the listed web servers. Using smoke and manual tests, it is clear that the most popular server, Apache, is not suitable to our aim. It requires significant hardware resource and does not manage efficiently many concurrent connections.

Caching

This section covers the back-end techniques in order to speed up the requests from end-users accessing the GLORIA website during the live webcast of the scheduled astronomical events.

HTTP caching

As described in RFC 2616, the HTTP protocol of version 1.1 was carefully designed with caching requirements in mind. To fully leverage the benefits of web content caching techniques, it is important to thoroughly understand the key principles on which it is based on.

Motivation of HTTP caching features

An HTTP client requesting particular object over an HTTP protocol could be satisfied either directly from the HTTP server it has requested or from the intermediate caching facility. To provide the client with up-to-date content and to lower the contention of the HTTP server at the same time, there is a possibility for HTTP server to indicate how long the content is valid and in which manner the client or intermediate caching facility can use it for caching purposes.

Expires response header

The Expires response header provides the information how long the content is valid for cache. After the specified time, caching facility is required to check the server back for fresh content. From the server perspective, there is a possibility to specify the Expires header in a number of ways. Most common of them is setting of absolute time to expire, a time based on the last client access time and, finally, a time based on the last time the content was modified on the server.

The only value being valid for the Expires header is a HTTP date in Greenwich Mean Time (GMT). Values in different format indicate the content as non-cacheable. An example of the Expires header is shown below:

```
Expires: Thu, 31 Jan 2013 13:59:23 GMT
```

Cache-Control response headers

To further extend the possibilities and control of caching mechanisms placed between the HTTP client and server, the HTTP protocol version 1.1 added a new class of headers called Cache-Control, so the content publishers have an option to better specify their intent related to its caching, such as, for example:

- *max-age=[seconds]* – provides maximum amount of time for which the content is considered as fresh.
- *s-maxage=[seconds]* – similar to previous one, but only for proxy caches.
- *public* – indicates that authenticated content is fully cacheable.
- *private* – indicates that authenticated content is cacheable only on client (e.g. in a browser).
- *no-cache* – caching is disabled for particular response; valid content can be obtained only from a new request.

When both Expires and Cache-Control headers are returned from server, Cache-Control header has a priority.

Header usage recommendation

To maximize the effects provided by HTTP caching mechanisms, it is recommended to maximize the time for which the content is valid, up to several months, even whole year. In situations when there will be a need to change the particular content in a web page, it is advisable to embed a version number into the resource URL so that the resource could be easily changed without need to invalidate the previous version of it.

Distributed web hosting

Objective

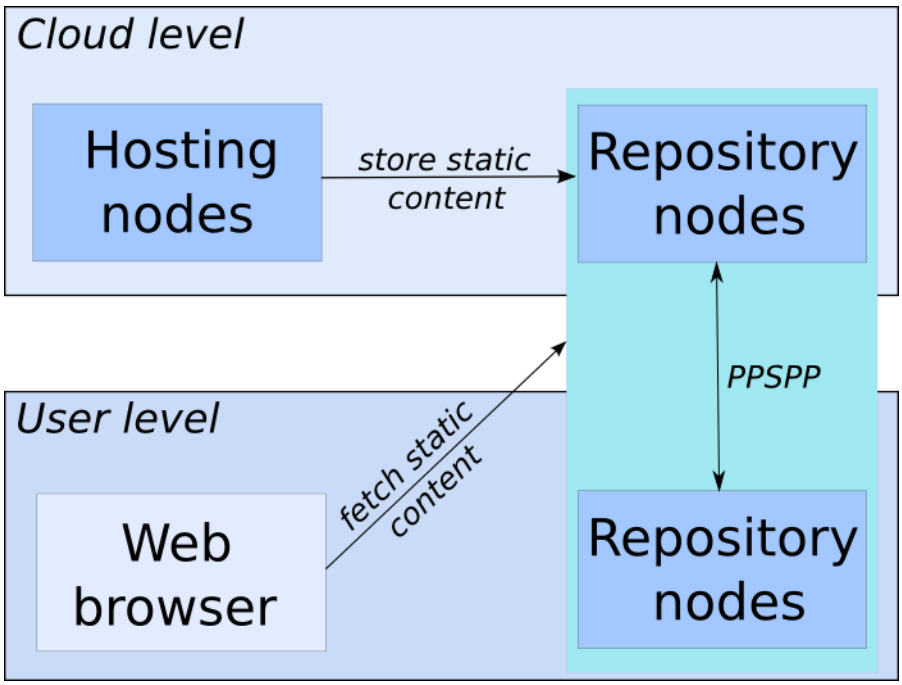
In order to extend compatibility among software solutions within the GLORIA network, a distributed network for web hosting is proposed. The idea behind this approach is to provide a resilient infrastructure for the GLORIA website that holds the live webcast of astronomical events. This infrastructure will combine both traditional cloud-based hosting solutions and a peer-to-peer overlay network. The peer-to-peer network will be used to persist the static files used by the GLORIA website, in the same way as media content is stored in the peer-to-peer data storage solution proposed in deliverable of this work package.

Design

The solution is divided into two different levels according to the capabilities of the network nodes in charge of providing the web hosting solution:

- *Cloud level*: represents the stable back-end solution of the distributed web hosting network. It is formed by storage and web application entities, creating a scalable solution for web servers and the distributed data repository for static web content.

- *User level*: defines the endpoint solution deployed at the user side. The end-user has the ability to contribute in the web hosting solution by donating storage resources. The system at the user level interacts with the cloud-based solution to coordinate the status of the distributed data repository.



Cloud level

At the cloud level, the system provides a cost effective and scale-out storage to host the GLORIA website and its static content. It is designed as a fully distributed system with no centralized coordinator, avoiding possible bottlenecks and overhead 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 a 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. For the aim of the distributed web hosting solution, two different network nodes will be used:

- *Hosting node:*

  This entity deploys a customized web server specifically designed to manage large amounts of requests against a static web page. The workload of the network traffic will be distributed among these hosting nodes.

- *Repository node*:

  This entity provides the content repository for the hosting nodes. The static content of the GLORIA website will be stored in these nodes, accepting concurrent requests from the web browsers of the end users. In addition, these nodes will be part of the peer-to-peer data storage network, providing a set of stable entities with high network capabilities. These repository nodes will act as peers in the same way as the repository nodes in the user level.

User level

In this level, the users can be part of the distributed web hosting solution by joining the peer-to-peer data storage network as repository nodes. These nodes interact with each other as well as with the nodes located in the cloud level. This solution provides a flexible storage repository built upon the resources donated by the GLORIA users. The repository nodes become peers in the network, to store and retrieve static content from the GLORIA website.

The network infrastructure for the repository nodes is based on the PPSP transport protocol. PPSP 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. PPSP is intended to be a built-in cross-platform protocol in the TCP/IP stack, supporting all major operating systems and web browsers.

Each web static content stored in this peer-to-peer network has a unique hash id. The web browsers of the GLORIA users only need this hash id to receive the data from any available source, while data integrity is verified. 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.

PPSP 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 use.

Deployment

The web hosting infrastructure will be deployed as follows:

1. A set of cloud-based repository nodes must be launched to store static content.
2. The repository nodes run a peer-to-peer software based on the PPSP transport protocol, accepting incoming connections from user-level peers.
3. A set of hosting nodes must be configured in the cloud level. The number of hosting nodes in the set is determined by an estimation of the number of users who will access the GLORIA website to watch the live webcast of the astronomical event.
4. A customized web server is installed in the hosting nodes. The configuration must be the same for all nodes to avoid inconsistency.
5. The GLORIA website is deployed in the hosting nodes. The static web content will be placed on the hosting nodes.

Front-end

Web design guidelines

The GLORIA website in charge of the astronomical live broadcasts must be designed in a way that allows the web hosting platform to scale seamlessly, while providing high availability content delivery.
One of the key features of a resilient web infrastructure is the number of requests from the end-user to the back-end. Reducing these requests per visitor, the back-end solution will support more visitors and therefore, better scalability in case of massive number of concurrent users access the website.

A website is defined by a set of web components, such as HTML documents, scripts, stylesheet, images, etc.. If all these web components are fetched from a single web server, the response time to load the website will be a critical. However, if the web components are loaded from a content delivery network, the performance could increase up to 20% in certain conditions.

Normally, users will access the GLORIA website more than once. This implies that there should be a caching mechanism to avoid frequent requests to the back-end if the web components are already downloaded. For this reason, a good practice is to add `expire` headers to the web components since in this case they will become cacheable.
The size of the web components is also an important factor when it comes to web performance. Using Gzip compression with the web components such as HTML documents or scripts, will decrease the loading time significantly. Minifying the scripts is also a good technique to speed up the response time.

Reducing the DNS lookups is also a good practice. The Internet service providers, operating systems and web browsers usually cache a DNS record for performance issues. Lowering the number of unique host names while keeping a certain amount of parallel downloads will also increase the response time.

Another technique in the GLORIA website is the use of static content. A simple web design will imply better response time and few requests to the back-end. In case data needs to be fetched from the server through ajax requests, these should be also cacheable.

Offline web application

The GLORIA website for astronomical live webcasts needs to be lightweight and make as less requests to the backend as possible. However, this is not the only requirement for a scenario where thousands of concurrent connections could cause a critical overhead in the web hosting infrastructure.

To avoid this situation, the offline feature provided by the HTML5 web standard can contribute on protecting against back-end outages, but at the cost of complexity. Offline technologies support caching and detailed control over caching process. Therefore, web apps can boot quickly and show data instantly.

The HTML5 specification for offline application defines two kinds of caching mechanisms: application and data. Since the GLORIA website for broadcasting will only present the live feed of the astronomical events and no user data is needed, the design of the website will be based only on application caching.

Application caching would retain the initial HTML document, the scripts, the CSS and frequently used images. Using this technique gives the GLORIA website three main advantages:

- Offline browsing: users can access the GLORIA website when they are offline or the back-end is unavailable.
- Speed: cached resources are local, and therefore the loading is faster.
- Reduced server load: the browser will only download resources from the back-end that have been changed.

All the information regarding which files the browser should cache are listed as resources in the cache manifest file, a simple text file. Once the application behaves in offline mode, it remains cached until the cache manifest file is modified or the user clears the local storage data of the web browser.

Distributed network client

A user can participate hosting static content of the GLORIA website using its own local resources. To do so, the GLORIA website will provide a client application ready to be installed. The client application will run in the context of the operating system, having access to the file system. When the application starts, it will contact with the distributed data repository, becoming a repository node in the network.

Testing

The tests will not only check the scalability and stability of the system, but will also measure the efficiency and compatibility of the code on the various platforms and devices. Parameter tuning for different hardware configurations will be encoded into device-specific initialization files to be distributed with the code.

Environment

Tests will be conducted in different environments and with various platforms. This will allow us to test all the levels of criticalities that will be met in the real world when the system will be deployed. Testing portable devices will be of particular interest and value. In fact we can easily assume that a foreseen list of environments include:

- Local, i.e. in a LAN: single machines with different OS and hardware resources, or cluster of machines running heterogeneous OS
- Open but limited within open platforms for planetary-scale projects like PlanetLab (planet-lab.org). PlanetLab is a global research network that supports the development of new network services.
- Real environments using the OpenStack cloud.

System testing

Integrated system test will involve a distributed set of nodes. In particular a cloud based system like the SOASTA CloudTest Lite (soasta.com) will be used. A certain set of different mobile devices will be included.

Acceptance testing

We plan to accomplish acceptance test by using both a simulated environment (e.g. Selenium - seleniumhq.org) and the real world with user accessing a live webcast of an astronomical event. For this we will invite the GLORIA community to participate with their resources to the test.

Performance testing

Load testing lets you measure your website's QOS performance based on actual customer behavior. When visitors reach a web site, a script recorder records the communication and then creates related interaction scripts. A load generator tries to replay the recorded scripts, which could possibly be modified with different test parameters before replay. In the replay procedure, both the hardware and software statistics will be monitored and collected by the conductor, these statistics include the CPU, memory, disk IO of the physical servers and the response time, throughput of the System Under Test (short as SUT), etc. And at last, all these statistics will be analyzed and a load testing report will be generated.

In spite of the fact that our package/service will not be subject to a service level agreement, we are committed to reach a good level of performance. This means we will try to remove any possible software intrinsic limitation, while having well in mind security of the user private data and a limited resources usage. In fact we will assume that users must be able to run other processes in parallel and eventually stop or start our P2P service anytime.

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 Malaga SPAIN
	UOXF University of Oxford UNITED KINGDOM
	UNIWARSAW Uniwersytet Warszawski POLAND