GISS 2.0 : map module
why & how to

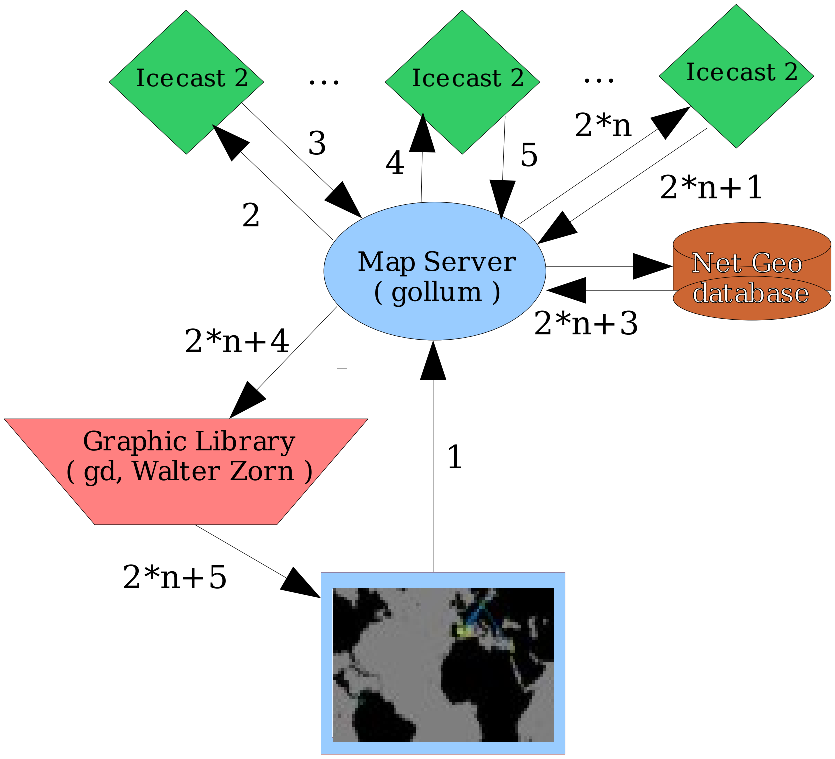
1. Introduction

This document explains some elements of software architecture of GISS map module and how to configure servers in order to be part of the good streaming material out of commercial communication strategies.

2. GISS map version 1.0

The former GISS/Gollum map was based on the following principles :

- a central machine was hosting the map and the GeoIP database that is needed to locate servers, sources and clients on a global map. This central machine was in charge of querying on request all icecast servers in the ring to get dynamic informations about mountpoints, clients, sources ,...
- visualization of the map using dynamic imaging libraries (1st version : gd, php server side library), then using some dynamic HTML library (2nd version : Walter Zorn, javascript).



Architecture of Gollum Map version 1.0

So, according to this schema, an update of a map on a client request (but we don't like this term of client) was taking like (2*n +5) requests over the internet, which is less than optimal.

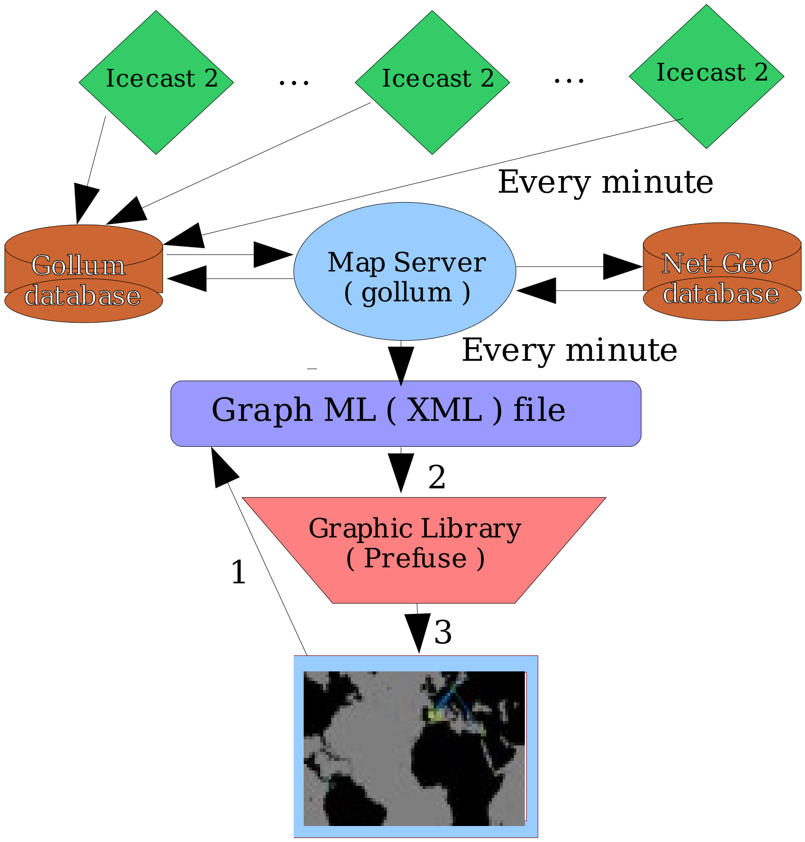
Some other problems were :

- people owning a server needed to communicate their icecast password for the map server to be able to query informations from them, these passwords were also traveling uncrypted over the internet every minute.
- The graphics libraries involved were no too flexible, one (gd) needed a request to the server for every zoom action from the user, the other one (walter zorn) was rapidly getting to a client machine freeze when too many nodes were active.

3. GISS map version 2.0

The major evolutions in the new architecture are :

- the creation of a database on the map server that can keep servers informations and the self-update procedure form the servers themselves.
- the visualization made with a java prefus'ed applet that can allow easy zooming, filtering and search.



Architecture of Gollum Map version 2.0

In the new architecture, all the server informations are updated on a regularly basis and stored in a map database, the client asking for an update only needs to reload informations from this database.

The servers are uploading their status to the central database by calling a script locally through a scheduling mechanism (crontab). No need for the map server to know of any passwords on the icecast servers.

The use of a commonly supported format (graphML) can open the visualization to many visualization toolkits that can understand this format, for now only Prefuse Mercator Visualization is available :
http://gollum.artefacte.org/mapuse/map.html

4. How to register a server

You can get all GISS map source code at sourceforge :

```
cvs -d:pserver:anonymous@etc-groups.cvs.sourceforge.net:/cvsroot/etc-groups login
( when it asks for a password, type <return> )
cvs -z3 -d:pserver:anonymous@etc-groups.cvs.sourceforge.net:/cvsroot/etc-groups co -P mapuse
```

In this package, which is all the map component sources, only the script xml.php is needed to configure a server. At the beginning, you need to change the following lines :

```
# icecast configuration section
$hostname = "_ihost_";
 --> change to your icecast server
$url = "http://_iuser_:_ipassword_@$hostname::8000/admin/stats.xml";
 --> change to your icecast admin account and password
# end icecast configuration section

# gollum database configuration section
$gdbHost="_host_";
$gdbUser="_user_";
$gdbPass="_password_";
# end gollum database configuration section
come to irc.goto10.org, channel #giss  to get these informations
```

Once this script has been modified, you need to put it in your crontab configuration file like this :

```
#m    h     dom   mon   dow   user   command
*     *     *     *     *     root   <path>/php <path>/xml.php
```

And join the streaming party, if any information is missing, connect to irc.goto10.org, channel #giss