

FILE MANAGEMENT IN THE DATA GRID

Isaac Iglesias Faustino
Summer Student, 2003

The Grid

The main subject of my work here in Desy has been the Grid. The Grid is a new way of interconnecting computers so they can share not only information (as the Web does) but also every computational resource. There is a Grid project supported by the European Union which is called the European DataGrid and that is the topic I centered my work in. Also the main target of the IT Group here in Desy is to build our own Grid at the same time that we learn how it works. There are different nodes in a Grid that develop very different functions:

UI (User Interface): It is the computer where the User logs in. It receives the petitions of the users and dispatches them to the Resource Broker.

RB (Resource Broker): It receives jobs from a User Interface with some specific requirements. It chooses which of the available connected Computer Elements matches the requirements of the job and sends it to it.

CE (Computing Element): It is the node where the computation task takes place. It is actually a group of nodes internally: a Gatekeeper and some Worker Nodes.

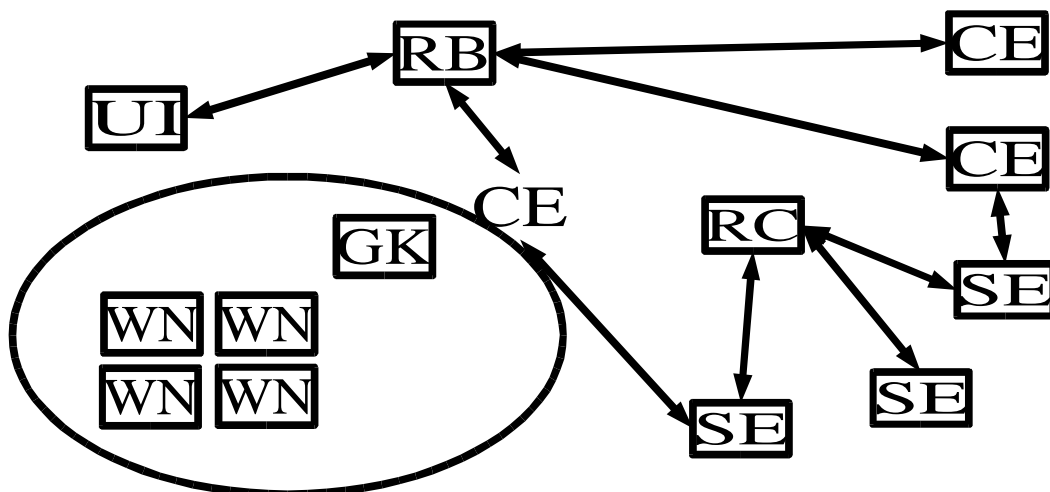
GK (GateKeeper): Its the frontend of a Computing Element, that handles the interaction with the rest of the Grid.

WN (Worker Node): Here is the computation actually performed.

SE (Storage Element): In a Grid huge amounts of data are constantly being handled. All that information is stored in those nodes.

RC (Replica Catalog): This node's function is try to have updated all the replicas or copies of the files in the Grid. Some files can be in different Storage Elements, so what the RC does, is keeping a database of them and also manages them.

There is a diagram below that may give some help to understand the way all the nodes are interconnected.



The nodes I have mainly worked with, are the CE and the UI as we will soon see. Nonetheless, before starting to work with them I had to read a lot of information and also do some practical exercises...

Getting used to the Grid environment:

The first thing I did after reading some preliminary information on the Grid was playing around a Grid in the UK, provided by David Kant. There I got used to the way of submitting jobs in a Grid, as well as to some JDL syntax and also how internally everything works. However, at that time the Desy-Grid was about to start running. There was already a User Interface running and also a Storage Element was up, so it was time to dwell into the way the Grid handles the files. The next duty they assigned me, was getting some information on the Grid file management and make a presentation about it to be given to the Grid group. That presentation should contain some practical examples about the three main file management operations: putting a file, copying it and erasing it. That way, the job will also be useful to check whether the SE is working or not. The presentation I am talking about is attached to this report although a brief summary can be clarifying...

File Management in a DataGrid:

To deal with files in a Grid, two different tools can be used: the low-level GDMP and the high-level RM; both part of the Globus Project, some project whose aim is build technologies to use Grids. GDMP (Globus Data Mirroring Package) is an efficient tools for safely replicating files in a Grid, while RM (Replica Manager) is a client tool that develops the same task using GDMP software. That is the reason why it is in an upper level.

The main differences between both are that GDMP replicates only within Storage Elements, replicate sets of files and is a client-server tools; while RM can work with SE's, CE's or even UI's (if they run gsiftp), replicates single files and only implements client side.

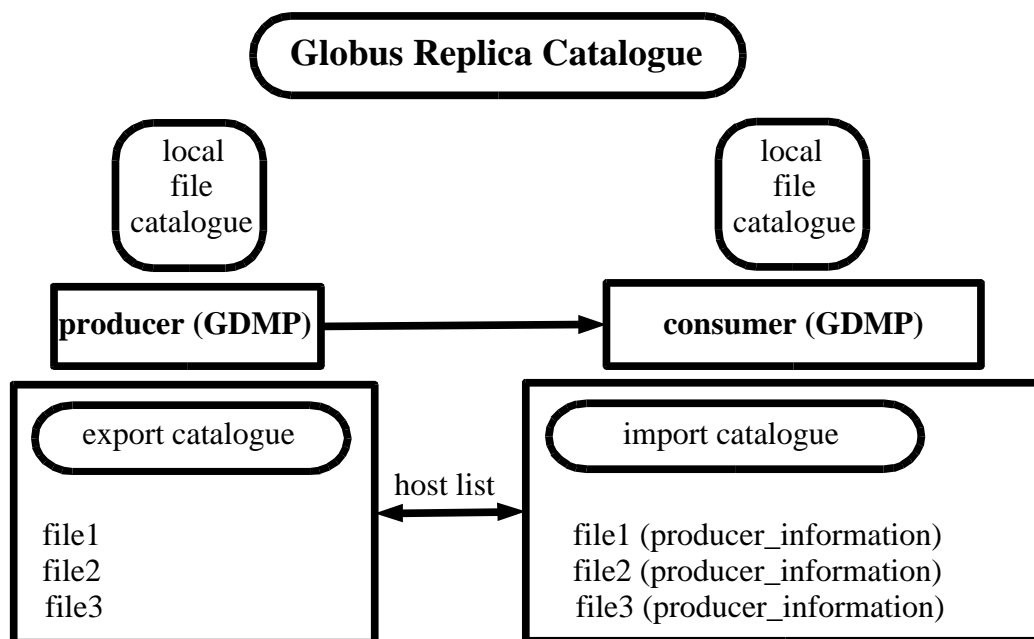
Another important fact to be know is the use of catalogues by the SE to manage the files. In every SE there are three diferent catalogues, which are nothing else but three files listing some other files. They are as follows:

Local File Catalogue: Contains all the files that a Storage Element locally manages. Its contents ought to be published so that other nodes are able to access the files and copy them.

Export Catalogue: Contents all the files that a nodes offers to another one to copy. All the files of the above catalogue are registered in this one at the moment of publishing.

Import Catalogue: Contains all the files that are wanted to be copied by the machine as well as some extra informaiton on the producer. There is a host list for different hosts to suscribe so their export and import catalogues are synchronized.

There is a diagram here as well, to make its function a bit clearer:



After all this introduction, the formal description of how to put, copy and erase will be next but we don't want to get too formal or too deep here in the report. Hence, we will only say a finally summary about the topic of my presentation.

To summarize, I would say that EDG-RM is a much safer, more powerful (API core implemented) and simpler (high level) than GDMP. However, some operation cannot be done with EDG-RM because it needs a gsiftp server running. In that case, we have to go lower and use GDMP.

While making the presentation we found out that the putting and copying operations could be easily tried in the Storage Element we already had. By the way, they worked, so we assumed that *grid002*, the first Storage Element set up by J. Nowak was working all right. Nonetheless, we noticed that to try the copying operation, two different Storage Elements were needed. Therefore, they assigned me a new duty. I should install a new Storage Element following J. Nowak's documentation. That way, I could test whether his documentation was complete and consistent. Afterwards, I should try the new Storage Element exactly the same way I tried *grid002* and then, having two SE's up, finally try the copying functions I had written in my presentation!

First of all, I tried to install a new SE in *grid005* and turn out to be a great mess. I guess I made some mistakes when installing it and, besides, the computer had serious hardware! Problems so it ended up a total chaos. We decided to physically shut down *grid005*, forget about it and try again in *grid010*. The time spent in *grid005* was not useless: I had some practise on installing SE's, so I installed SE in *grid010* quite straightforward...

Installation of a Storage Element in grid010

First of all, the installer is supposed to follow the five common points to each node in the Grid. Globus Toolkit, Certificate Authorities and EDG must be installed first. In order to do that, you must download the needed libraries. Those are easily found in useful lists in the official web site of the European DataGrid. You must note that it is not necessary to install the three items totally. Hence, each node has its own rpms to be downloaded.

The Certificate Authorities and Globus Toolkit never bring any problem at all, but the EDG is usually much more complicated. If you first try to install the EDG packages you will immediately come across to a handful of module dependencies. Most of this dependencies can be solved used the External list of rpms, part of the ones mentioned above. Some others, can be solved by picking up the rpms from the own Suse Linux of the node, in */net/linux/suse*. Nonetheless, some dependencies have to be downloaded directly from the Internet. Besides, every new rpm can depend on some other ones, so this task can get really tedious. A very typical problem I found was that I needed a specific version of a specific package that I could not install because a newer version was already installed. There are two different ways of solving this problem: choosing the *-nodep* option when installing the requesting rpm and hope it works ok with the newer version (that is the way it should be) or you can as well remove the newer version and install the old one using the *-force* option when installing the requested rpm.

Once you have finally installed the EDG package, boring but not too complicated duty, you only need to add a few lines to the linux configuration file *ld.so.conf* telling which the library folders are.

Afterwards, the so-called 1-3 step for a Computing Element node should be performed. First step must be installing the host certificate and key. To carry this out, the Virtual Organization has to provide you those two files *hostcert.pem* and *hostkey.pem* for you to copy. In Desy, those certificates can be found in */afs/desy.de/user/g/gellrich/it/grid/cert*. Then the two-line file */etc/sysconfig/globus* must be created and the globus initialization script must be run. None of those task gave me any problem.

Now, we get into the specific SE installation. Next step is setting up three important configuration files: */etc/globus.conf*, */etc/globus2.conf* and */etc/edg/info-mds.conf*. I did not get really thorough with those configuration files, I just followed Nowak's indications and filled out the required fields.

Next, you have to configurate GDMP for every VO your site supports. You must run the */opt/edg/sbin/configure_gdmp* command to create a directory in */opt/edg/etc* for each VO, in our case, just one: Desy. Two configuration files have to be modified in that directory as well: *gdmp.private.conf* and *gdmp.conf*.

Then, we must run the four daemons that are needed in a Storage Element. The first one: GDMP Server (*gdmp_server start*) does not bring any problem at all, but the other three did. They are the Globus MDS (*globus-mds start*) the Globus Gatekeeper (*globus-gatekeeper start*) and the GSI FTP (*globus_gsi_wuftp start*). All of them use command that does not exist in SUSE Linux: *daemon*. Therefore, I had to change that command for the equivalent in SUSE: */sbin/start_proc*. Moreover, the *globus_mds*

did not work even after that. The problem was really hard to find and when trying to find it, once, I executed the starting program to see the output message because I was sure at that point that it was not working. Then, with no apparent reason, it worked and I do not have a clue of what it made it work. I read again all the instructions I had typed before and I have the intuition that is something related to tokens and permissions but, unfortunately, I am not sure.

Last step is creating the */flatfiles/grid010/desy* directory where the copied files will be stored. The appropriate permissions should be set and the ones of J. Nowak turned out to be too restrictive to me, so I had to change them and also some other users probably would have to do it too. Now with all the daemon running and everything installed it is time to test it...

Testing of the Storage Element in grid010

I tried first of the globus tools, *globus-url-copy* and perfectly worked. The next step was registering the file using one of the *gmdp* commands. At this point it failed because of some permission problem. I use to try the other node as a different user (*jnowak*) but, at that moment, as myself, it was not working. Apparently, the software was running well but the node was badly configured. Dwelling a bit into it, it turned out to be a deeper problem, because the other *grid002* node actually only worked with *jnowak* and the was something else needing to be configured.

At this point, unhappily, the end of the Summer Program period come and I had to leave the work at that point, with the feeling that *grid010* was in fact running all right, but with something to change in the configuration of the nodes, not explicitly said in the documentation we have at the moment. So, obviously the next thing to do, if I could stay, or for whoever who keeps on working on the topic, would be trying to solve this problem.

Final Conclusion

Now that the Summer Program has come to an end, I finish with a bit of a disappointment feeling for not being able to finish the checking of *grid010*. Although my work here had no specific aim, but to get as far as possible in the knowledge of the Grid; I would like have like to end with practical application of all that I read and studied first on file management.

However, I did learn a lot of things about the topic that are indeed going to be really useful during my last years of degree. Moreover, the most important thing that I learnt was something totally new for me: the way research is done is Research Centers. The weekly meetings with the Grid group as well as the communication between all the researcher or the collaboration with other institutions taught me how to work in a group. The main fact I get from that is that, when working in a group, the most important thing is to save your job in a tangible, clear, physical way, so it can really be useful. That way, when another person works on the same topic, can start at the point you left it and not over from the beginning. That kind of abstract knowledge on how to do research on something is what I enjoyed best, rather than concrete knowledge about some concrete topic, which is ok, but much less important.