An Open Source Grid Storage Controller for Genealogy Storage

Mikael Ronström, Ph.D
Open Source Consultant
iClaustron AB
mikael.ronstrom@gmail.com

1. Introduction

Storing genealogical information requires vast amount of storage space. The 2 million microfilms when they have been scanned into digital format will require many petabytes of storage space. New historical documents are constantly added. Also many pages requires storing multiple copies of the document with different resolution and so forth to enable better readability of the document.

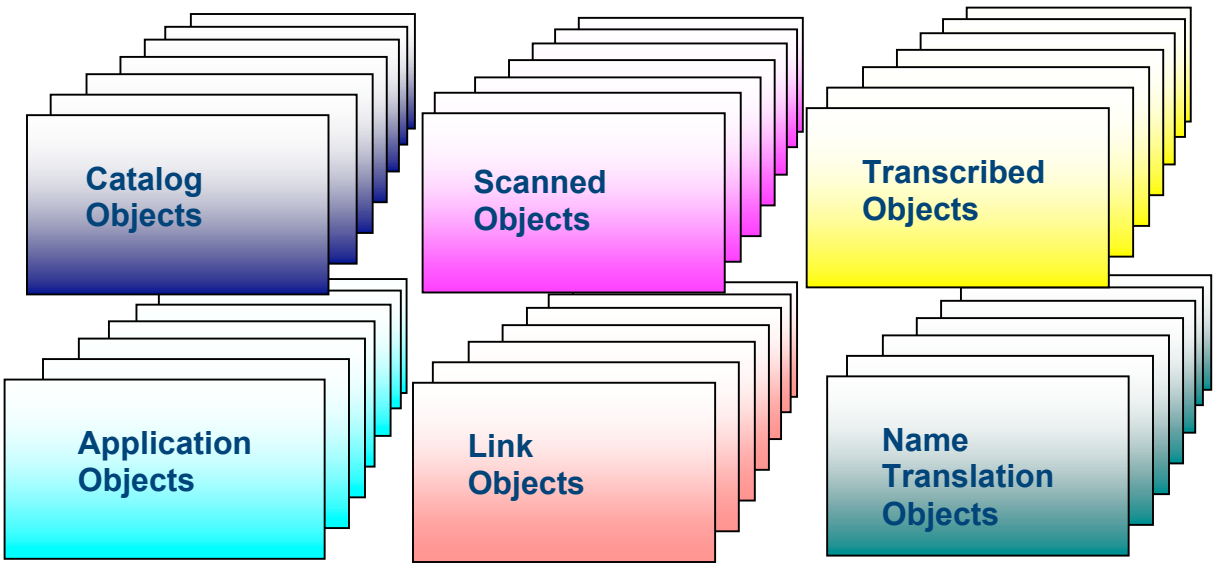
Storing the data in digital format is already possible at a fair price using tapes and DVD's. However this doesn't provide on-line accessibility of the documents which is essential to support the many millions of genealogists around the world. In Sweden a commercial company has already scanned all church books and made them available for viewing on the Internet for subscribers.

So the technology to make all historical documents available on the internet is already technically feasible. The question is now to also make it economically feasible. Current solutions for a system in the petabyte range, has a price tag which is not economically feasible. Using servers with cheap disks is economically feasible but 2 problems remain. The first problem is that software to bind a large set of servers together in a reliable manner is not available. The second problem is to ensure that running those servers have an economically feasible electricity bill.

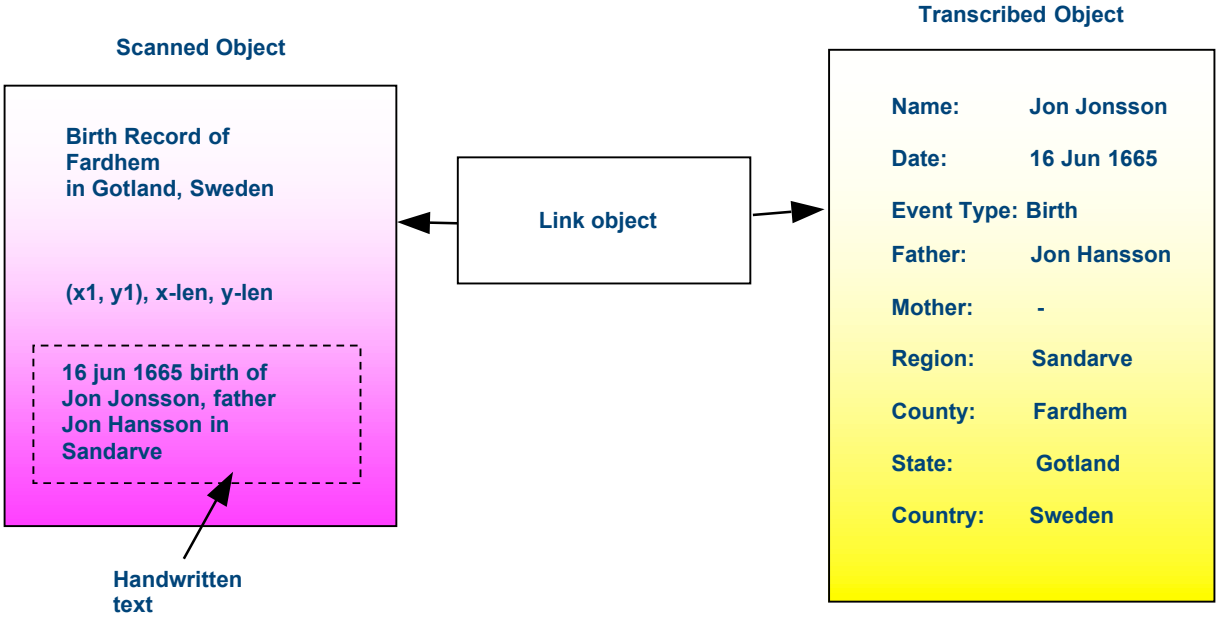
This paper describes some ideas on how to build an Open Source Grid Storage Controller that can be used for Genealogy Storage and other applications requiring vast amounts of storage. It also describes some ideas on how to apply this architecture for the genealogy storage application. Finally it provides some ideas on how to create mappings between scanned objects, transcribed objects and application objects that describe the relations between our ancestors.

2. Mapping between Scanned objects and other objects

There are a number of different object types in a genealogy application. There are catalog objects to describe the information available. There are scanned objects derived from historical documents. There are transcriptions of the scanned objects. There are also application objects describing the actual persons and their relations. In this paper it is also suggested to add Link Objects and Name Translation Objects. Name Translation Objects are used to handle diverse spellings of the same name but also localised to a region. This is essentially information provided by subject matter experts.

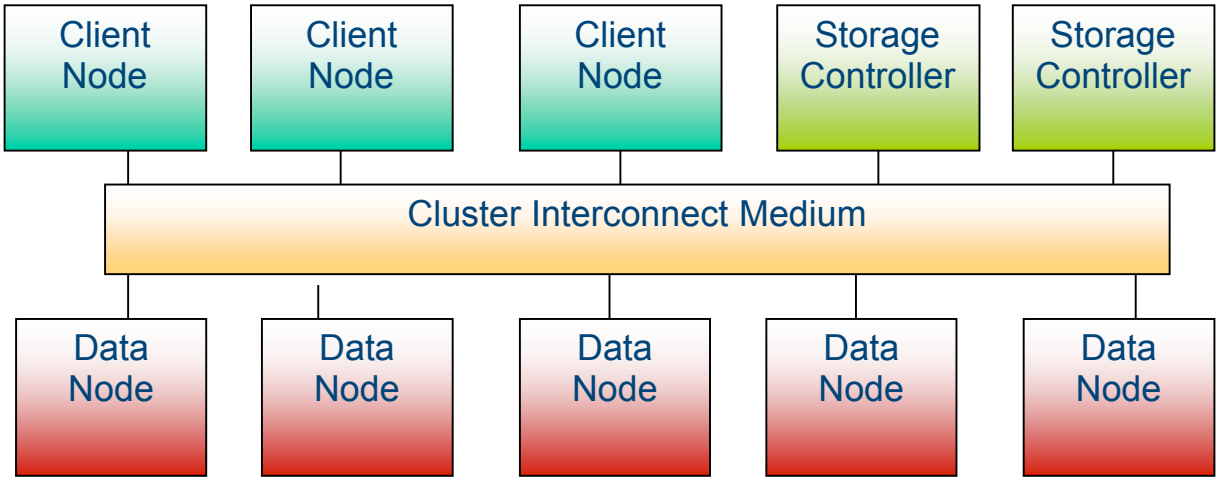


Link Objects are used to make mappings between the different objects. In the example below we have a scanned object describing a page of historical data from a birth record. It describes the birth of Jon Jonsson. Someone has added a transcription of this part of the historical document and this is described by a rectangle in the historical document. From this rectangle there can be many Link objects mapping the birth record to application objects, transcription objects and other genealogy objects. In this manner one can say that the referencing from a genealogy becomes global data.
In order to ensure proper quality of the information the idea is that one will use the same type of prioritisation as used by e.g. Amazon. So other genealogists will give this Link Object a certain weight. Also genealogists can give weight to each other such that genealogy experts can have a higher influence than beginners.



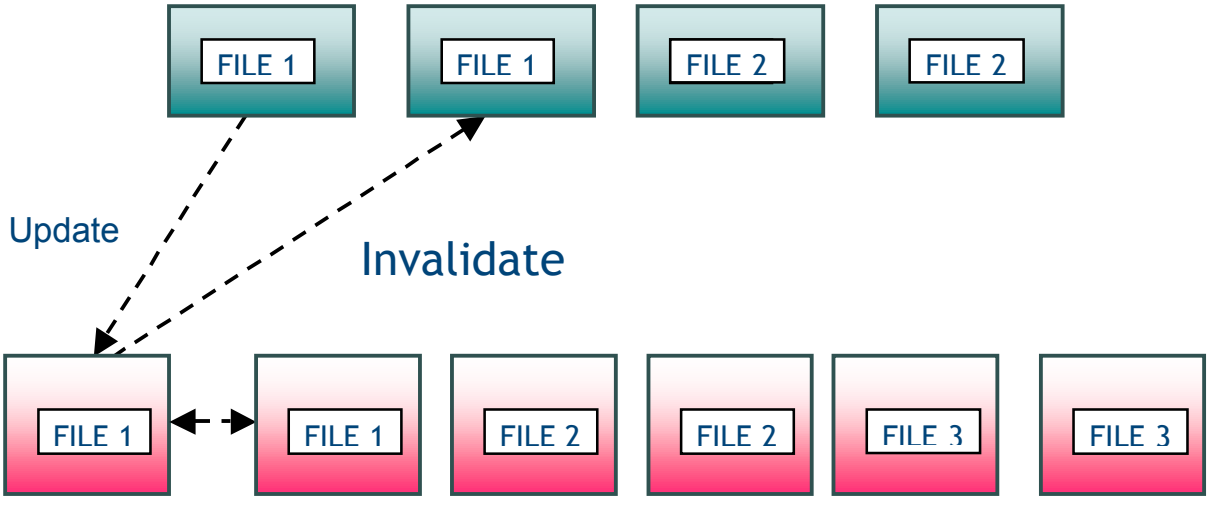
3. Cluster Architecture

The proposal for this genealogy storage is to base it on the MySQL Cluster Software. MySQL Cluster is a parallel database. The idea is to reuse the data nodes of MySQL Cluster and replace the MySQL Servers by Client Nodes that implement either Clustered Filesystems or iSCSI protocols. There are also Storage Controllers that are used for exporting data to clients outside of the cluster. Basing it on an existing open source product means that it will have high availability and many years of development work can be reused to implement the genealogy storage system.



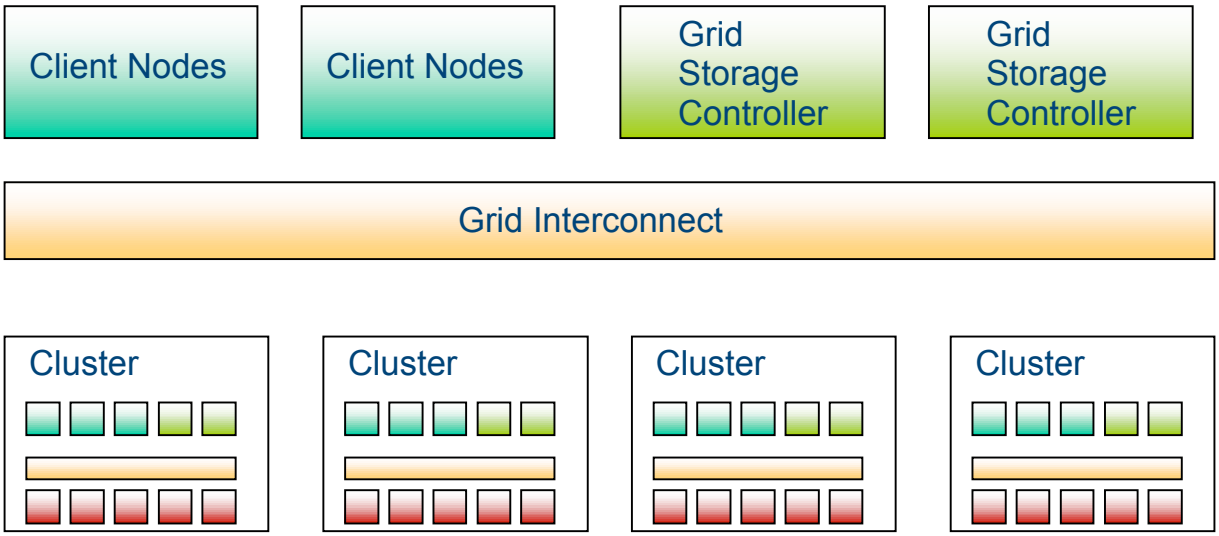
3.1 Cache Synchronisation

One of the new additional problems that one needs to solve is synchronisation of caching of file contents in the OS caches. The idea is that the Data Nodes will keep track of each open file in the system and thus each time someone updates a part of a file, the Data Node will know where to send all the invalidate messages such that all clients can invalidate those cache entries that have been changed. For smaller changes on popular data it could also be an idea to update the caches instead of invalidating them.

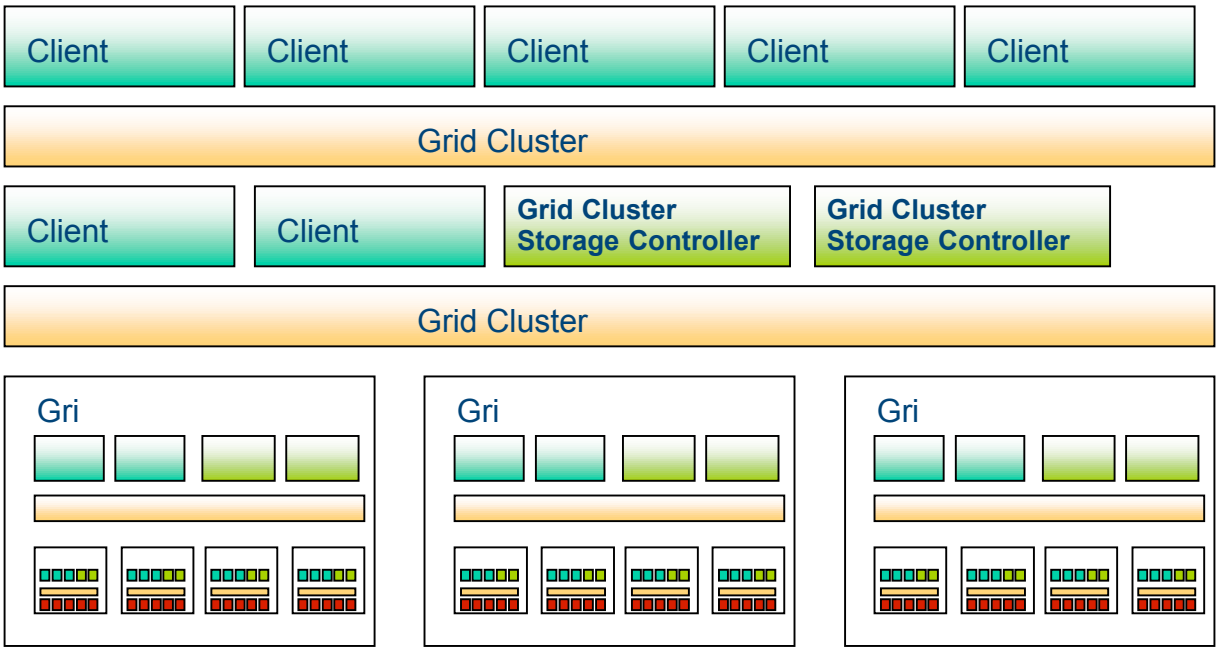


3.2 Grid Cluster

For genealogy storage requiring many petabytes of storage, it won't be enough with only one cluster. It will be required to have several clusters joined together into a grid of clusters. So here the client nodes will access the Storage Controllers of the clusters. It will be possible to access a grid or a cluster at a time. As an example we could imagine that we have a grid for all scanned documents and that there is a cluster for each country. In this case if we know that we are only accessing files within one country it's enough to access a client node in that country's cluster. If we need to access files in many different clusters then instead we connect to a client node with global accessibility.
Grid Storage Storage Controller is used to provide access to client nodes that needs access to several grids.



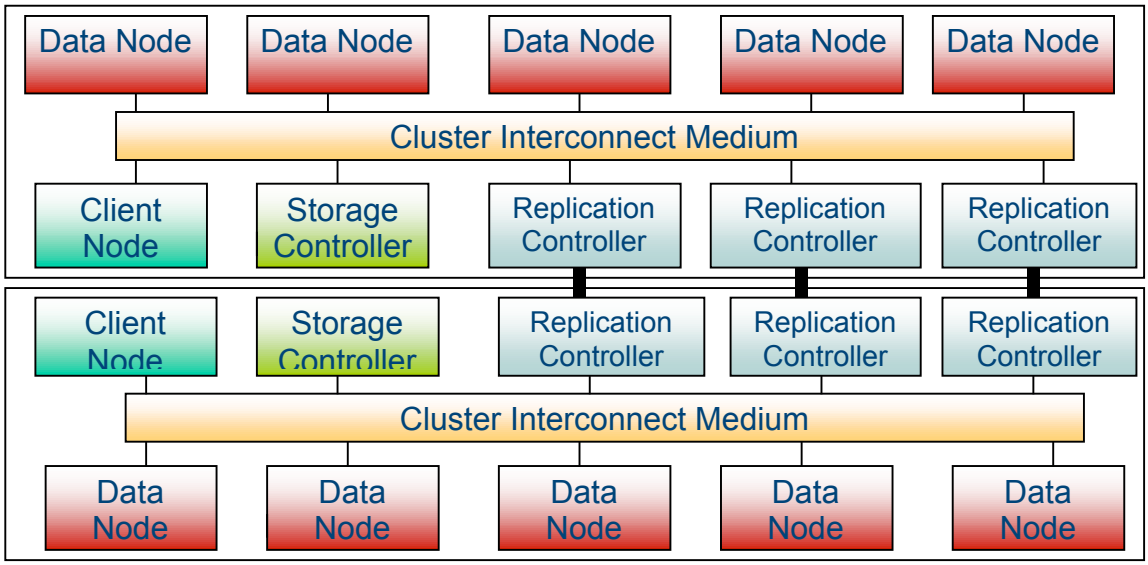
The final step is to also have a cluster of grids. It might not be necessary for genealogy storage but is a natural extension of the grid concept.



3.3 Replication between Clusters

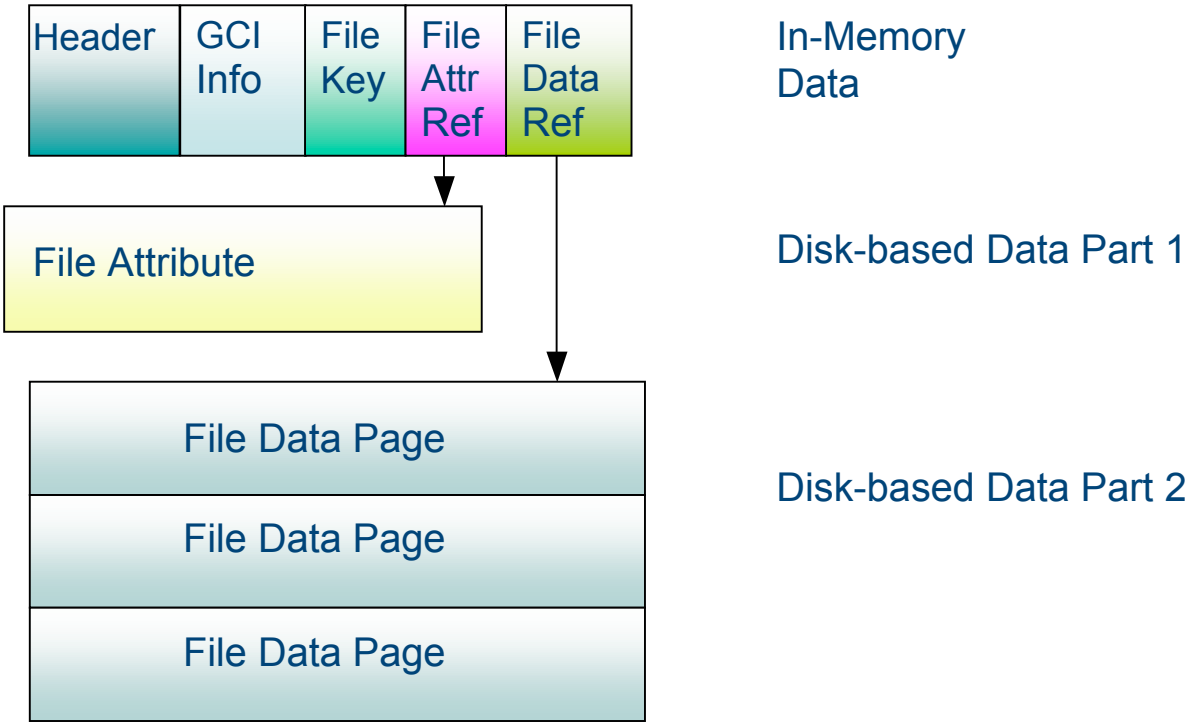
It's easy to think of the need to have clusters placed where they are mostly used from. As an example the cluster for swedish scanned documents seems natural to place in Sweden. It is likely that one wants a central copy of the data as well. The Solution to this problem is replication between clusters.

The solution to this problem is always solved from cluster to cluster. Thus a set of replication controllers ensure that data is replicated between the clusters. Thus if the primary replica of the Swedish cluster is placed in Salt Lake City, it is easy to also have a cluster located in Sweden for easy access for users in Europe. At the same time this replicated cluster enables a higher availability of the genealogy storage.



3.4 File Data Structure

Finally some words on how to implement this file storage solution on lower level. The idea is to use the fantastic development of memories and use this to store some part of the file in memory. This solution has the benefit that we can quickly access any part of the file with no more than one disk access and the file attributes could potentially also be stored in memory. Thus using memory for all search structures but using disks to handle the vasts amount of storage for genealogy documents.



4. Conclusion

Ideas on implementing an extremely large storage system using open source technology have been presented. This system could be used to build a genealogy storage system for petabytes of storage in an affordable manner.

Also some ideas on how one can use references to and from these genealogical documents in implementing a global genealogical database.