Using Grid Federations for Digital Preservation

Gonçalo Antunes
IST/INESC-ID Information Systems Group
Lisbon, Portugal
goncalo.antunes@ist.utl.pt

Helder Pina
IST/INESC-ID Information Systems Group
Lisbon, Portugal
helder.pina@ist.utl.pt

ABSTRACT

Digital preservation aims at guaranteeing that data or digital objects remain authentic and accessible to users over a long period of time, maintaining their value. Several communities, like biology, medicine, engineering or physics, manage large amounts of scientific information, including large datasets of structured data that matters to preserve, so that it can be used in future research. To achieve long-term digital preservation, it is required to store digital objects reliably, preventing data loss. The data redundancy strategy is required to be able to successfully preserve data. Many of the characteristics required to implement, manage and evolve a preservation environment are already present in existing data grid systems, such as replication and the possibility to federate with other grids in order to share resources. We propose the customization of a data grid platform in order to be able to take advantage of its replication and federation features. In that way, scenarios where federated grids not thought for preservation purposes can be extended to preservation and their spare resources used with that mission.

Categories and Subject Descriptors

H.3.7 [**Digital Libraries**]: Systems Issues; H.3.4 [**Systems and Software**]: Distributed Systems

General Terms

Algorithms, Design, Reliability, Verification.

Keywords

Data Grids, Digital Preservation, Redundancy, Federations, Replication.

1. INTRODUCTION

The Institute of Electrical and Electronics Engineers (IEEE) defines interoperability as ‘the ability of two or more systems or components to exchange information and to use the information that has been exchanged’ [1]. Digital preservation aims at ensuring interoperability in the time dimension (interoperate with the future), that is, guarantee that data or digital objects remain authentic and accessible to users over a long period of time, maintaining their value.

Several communities, like biology, medicine, engineering or physics, manage large amounts of scientific information. It usually includes large datasets of structured data (e.g., data captured by sensors), physical or mathematical simulations and several highly specialized documents reporting the work and conclusions of researchers.

The above mentioned information can be represented in a wide range of file formats and include a high level of relations that are not expressed in the data model of the file format. Moreover, the collaborative environment of the scientific community, and associated services and infrastructures, usually known as e-Science (or enhanced Science) [2], involves the requirement of interoperability and the respective data sharing. In a broad sense, e-Science concerns the set of techniques, services, personnel and organizations involved in collaborative and networked science. It includes technology but also human social structures and new large scale processes of making science. It also means, on the same time, a need and an opportunity for a better integration between science and engineering processes. Thus, long-term preservation can be thought as a required property for future science and engineering, to assure communication over time, so that information that is understood today is transmitted to an unknown system in the future.

In order to successfully transmit information to future generations, several strategies are possible such as format or storage media migration, emulation of hardware and software environments to be able to render the information, hardware refreshing, inertia, preservation metadata, and auditing [3].

To achieve long-term digital preservation, it is required to store digital objects reliably, preventing data loss. One potentially relevant strategy to achieve this goal is combining redundant storage and heterogeneous components. In using the redundancy strategy, digital preservation systems can take advantage of a basic attribute of digital information: it can be copied without any loss of information. This means that several copies of the data can be stored across many components. Through the use of the diversity strategy, which promotes the diversification of the properties of the components, the number of simultaneous failures in the system can be limited and the system is more likely to survive to a large correlated failure, such as in the case of a worm outbreak.

Achieving the goal of digital preservation may require a large investment in infrastructure for storing data, and on its management and maintenance. Such costs may be prohibitive for small organizations, or organizations that do not have steady revenue, like university libraries, research laboratories, or non-profit organizations.

An already common low-cost technology to handle e-Science collaboration and data management is the use of data grids [4].

iPRES2011, Nov. 1–4, 2011, Singapore.

Table 1 – Digital preservation threats and vulnerabilities taxonomy [3]

Vulnerabilities	Process	Software faults Software obsolescence
	Data	Media faults Media obsolescence
	Infrastructure	Hardware faults Hardware obsolescence Communication faults Network service failures
Threats	Disasters	Natural disasters Human operational errors
	Attacks	Internal attack External attacks
	Management	Economic failures Organization failures
	Legislation	Legislation changes Legal requirements

These are highly relevant solutions for digital preservation, as they already store massive amounts of the data that must be preserved, such as in e-Science domains, and they provide a set of functionalities required by digital preservation systems (e.g., redundancy, diversity). Furthermore, grids can be organized in different ways [5]. In particular, grids can be federated with each other. The federation model allows grids belonging to different institutions, and thus with independent administration and in different locations, to interoperate with each other so that data can be shared.

The iRODS data grid [6] is an adaptive middleware system that facilitates the management of data and policies according to the needs of the users. For that, it uses a rule engine to enforce and execute adaptive rules. Additionally, it supports the federation of different iRODS deployments. However, iRODS is not addressing specific digital preservation requirements, requiring customization to do so.

We propose the customization of the iRODS data grid platform in order to be able to take advantage of two types of scenarios: (*i*) grids *exclusive* for preservation, which comprises machines dedicated to running the data grid exclusively for digital preservation, which are likely to be under administration of the data owner; and (*ii*) grids *extended* for preservation, in which existing grid clusters, initially created for data processing, can be federated through the installation of an iRODS instance and extended for preservation. Their spare disc space, CPU, and bandwidth can be used to store data according to the preservation requirements. To be able to do so, we propose a set of micro-services and rules that use the replication features and federation configurations of iRODS to maintain data replicated geographically, so that it can be preserved from threats.

This paper is organized as follows. Section 2 describes related work, such as digital preservation threats and vulnerabilities, Data Grids, the iRODS data grid system, and the usage of data grids for preservation purposes in previous projects or publications. In section 3 we describe the problem of configuring the iRODS data grid in order to be able to take advantage of its replication and federation features. Then, in section 4 we discuss our proposal, and we finally conclude in section 5.

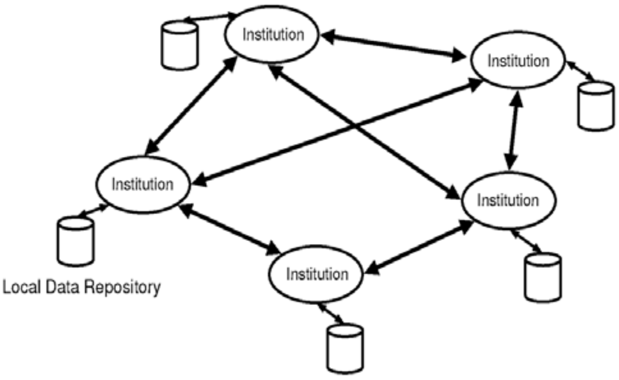

Figure 1. Grid federation model [5]

2. RELATED WORK

In this section, we describe related work, such as digital preservation threats and vulnerabilities, data grids, the iRODS data grid system, and the usage of data grids for preservation purposes.

2.1 Digital Preservation Threats and Vulnerabilities

In [3], the authors define a taxonomy of digital preservation threats and vulnerabilities in which a preservation environment is considered the aggregation of different components, namely the information entities, including preserved objects and metadata, processes controlling the information entities, and the technological infrastructure that supports the preservation environment.

Based on that assumption, each of these components may present several vulnerabilities: (*i*) process vulnerabilities, affecting the execution of processes (manual or supported by computational services) that control information entities; (*ii*) data vulnerabilities, affecting the information entities; and (*iii*) infrastructure vulnerabilities, enclosing the technical problems in the infrastructure's components.

Processes supported by software services can be affected by software faults and software obsolescence. Data vulnerabilities include media faults and media obsolescence. Infrastructure components can suffer hardware faults, hardware obsolescence, communication faults, and network services failures.

As for threats, those can be classified into disasters, attacks, management and legislation. Management failures are the consequences of wrong decisions that produce several threats to the preservation environment, such as economic failures and organization failures. Disasters correspond to non-deliberate actions that might affect the system, such as human operational errors, or uncontrollable events, such as natural disasters. Attacks correspond to deliberate actions affecting the system, such as internal or external attacks. Finally, legislation threats occur when digital preservation processes or preserved data violate existing legal requirements, or new or updated legislation (legislation changes).

2.2 Data Grids

Since it was defined in the 90’s, many applications of this technology were made, and grids are used in scientific research projects, in enterprises, and other environments that require high

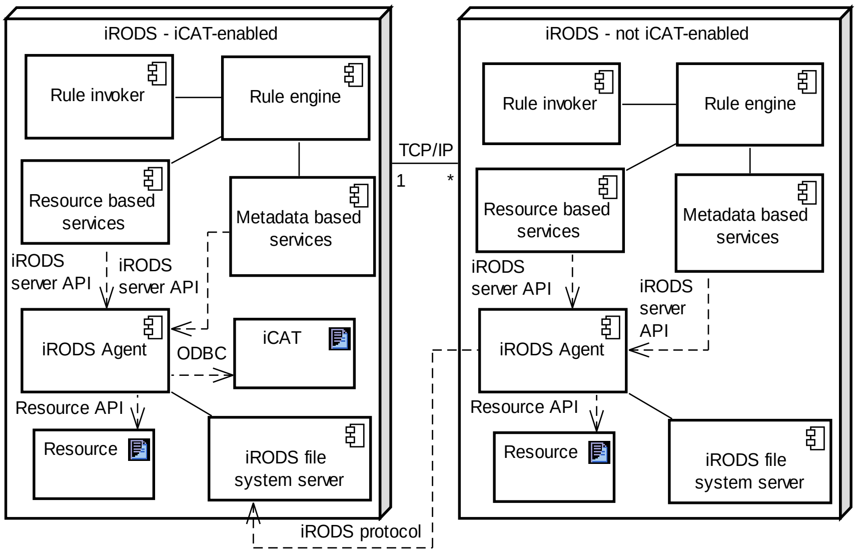

Figure 2 - iRODS deployment diagram [7]

processing power, while using low-cost hardware. Foster proposed a definition that consists in three properties that a system must comply with in order to be considered a grid [8]: (*i*) Resources are subjected to decentralized control; (*ii*) Standard, open, and general purpose protocols and interfaces are used; (*iii*) Nontrivial qualities of service are delivered.

In recent years, research has been done in defining a new type of grids, which deal with the management, sharing and processing of data. These were called data grids. Data grids offer distributed services and infrastructure that allow the support of applications that deal with massive data blocks stored in heterogeneous distributed resources [9]. In data grids, data is organized as collections or datasets, and is replicated using a replica management system that creates, manages and modifies replicas. Information about replicas is organized in a replica catalog. The main characteristics are the following:

- Massive Datasets: Data grids allow the management and access to enormous quantities of data, in the order of terabytes or even petabytes [10].
- Logical Namespace: Is provided through the use of virtual names for resources, files and users. In the case of resources and files, one logical name maps to one or more physical names.
- Replication: Increases scalability and reliability through greater availability and redundancy. Data grids, as big distributed systems, must implement data replication mechanisms, in order to guarantee system scalability.
- Authorization and Authentication: Due to the high importance and frailty of some of the shared data, authentication and authorization mechanisms must be taken into account in order to comply with the authenticity and integrity requirements.

Grids can be organized in federated zones. Each zone has full control of its administrative domain and can operate independently of other zones. A federation of zones allows the sharing of data and resources between zones in the federation. The main benefits of this configuration are Location Transparency, as users can access resources at any node in a transparent way; Availability, as the replication in different storage media, in different locations allows the data to be available throughout the grid; Administration, as systems of different administration share a single sign-on environment and access control lists; Fault tolerance, due to replication in local and remote storage systems; and Persistence, since data can be migrated to new local supports without affecting availability [11]. Figure 1 represents the federation model of organization of data grids.

2.3 iRODS

The iRODS[1] system is an open-source storage solution for data grids based on distributed client-server architecture. A database in a central repository, called iCAT, is used to maintain, among other things, the information about the nodes in the Grid, the state of data and its attributes, and information about users. A rule system is used to enforce and execute adaptive rules. This system belongs to the class of adaptive middleware systems, since it allows users to alter software functionalities without any recompilation. Figure 2 shows the UML deployment diagram of iRODS. Note that the iCAT database only resides in the central node and many other nodes can be connected to the central node.

iRODS uses the storage provided by the local file system, creating a virtual file system on top of it. That virtualization creates infrastructural independence, since logical names are given to files, users and resources. Management policies are mapped into rules that invoke and control operations (micro-services) on remote storage media. Rules can be used for access control, to access another grid system, etc. Middleware functions can be extended by composing new rules and policies.

The federation of multiple iRODS data grids is also a feature. Through the federation mechanism, an independent iRODS grid (i.e., an iCAT-enabled node and possibly zero or more non-iCAT servers connected in a grid) can interoperate with other independent iRODS grids. Each federated grid is called a *zone*. In

[1] https://www.irods.org

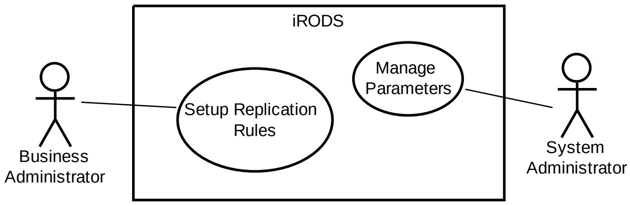

Figure 3 – Use case diagram of the proposal

a federated schema, a user with appropriate permissions can access objects stored in any iRODS node, belonging to any of the federated grids. iRODS also supports the existence of multiple federations.

2.4 Data Grids and Digital Preservation

Several publications and projects addressed the potential usage of data grids in digital preservation. The *InterPARES 2*[2] project studied the usage of data grid technologies for the building of preservation environments [12]. The conclusions were that many of the characteristics that were required to implement, manage and evolve a persistent archive were already present in existing data grids [13]. Data grids are also described as being useful for managing technological obsolescence, due to the virtualization of the underlying storage technologies [10].

In [14], the increasing needs of traditional archives and libraries for the preservation of large quantities of data is pointed as a driver for the collaboration with science and engineering partners for the use of data grid infrastructure. In the same reference, several US initiatives dealing with preservation using data grids are described. Data grids are also suggested for preservation purposes in [15].

As for concrete preservation solutions using data grids, in [16] an implementation of a prototype grid-based digital library using gLite[3] gLibrary[4] is described. iRODS extensibility features are explored in [17] to implement digital curation strategies. The use of the Storage Resource Broker (SRB) data grid for the preservation of digital media has been reported in [18]. iRODS usage for preservation purposes is explored in the SHAMAN project [19], while the DILIGENT[5] project explored the use of gLite [20]. The iRODS data grid technology was analyzed from the point of view of threats and vulnerabilities in [7].

3. THE PROBLEM

As already referred, redundancy is an important means to withstand failures that might endanger data. According to [3], redundancy is required to be able to recover data from storage media faults, natural disasters, human operational errors, internal attacks, and external attacks. In that sense, it can be said that redundancy is a crucial feature in any digital preservation solution.

Data grids in general offer the particularity of having, among other desired characteristics, a replication feature. Moreover, the fact that grids belonging to different institutions can engage in federations with other grids, while retaining full administrative control of their domains can be a useful feature in preservation scenarios, since additional storage space is obtained this way. This would allow a preservation system to take advantage of "borrowed" resources belonging to another administrative domain.

The iRODS data grid system supports both replication features and the creation of federations. However, it requires customization in order to take advantage of these features, namely through the micro-service mechanism. Micro-services are small and well-defined functions/procedures that execute a determined micro-level task. Users and administrators can chain micro-services in order to create macro-level functionalities (also called Actions). An example of a rule definition for an action is the following:

- *actionDef | condition | workflow-chain | recovery-chain*

The *actionDef* corresponds to the identifier of the rule. The *condition* field specifies a condition that must be met in order that the *workflow-chain* - a chain of micro-services - can be executed. Conditions can be one or more logical expressions. A workflow chain can be composed of several micro-services or actions (other rules), separated by "##" characters. The *recovery-chain* specifies a chain of recovery micro-services chain that will be executed in case something fails on the execution of the workflow-chain.

The composition of micro-services is not straightforward. iRODS already features a plethora of micro-services providing generic operations (for instance, a replication micro-service is already provided with iRODS). However, if one is looking for more specialized micro-services, programming skills are required. Furthermore, the composition of rules with the objective of implementing different kinds of data processing can also be cumbersome since it requires the learning of the syntax and direct editing of iRODS rule database. This requires that the person administrating the preservation system possesses strong technical skills which might be a barrier to the widespread adoption of this type of system as a digital preservation solution.

In addition to this, the federation feature of iRODS only allows a limited control of the resources and data of remote federated grids, due to each grid having its own administrative domain. For instance, access to data in a remote grid is possible for an authenticated user, but writing new data or updating existing data is a limited feature, requiring some tweaking.

4. THE PROPOSAL

In this section we describe our proposal, which is composed of an interface for the easy composition of replication rules, a compiler which transforms the composed rules into iRODS rules, a replication service which enforces the replication rules on the ingest of files, and an audit service which maintains the number of replicas. The replication and the audit services will have to run on each federated grid.

4.1 The Composition of Replication Rules

We can consider that we have two kinds of abstract actors: a Business Administrator, which is responsible for the creation and enforcement of replication rules and might not have strong technical skills, and a System Administrator, which is responsible for the administration and maintenance of the technical aspects of the system, and thus of replication.

[2] http://www.interpares.org/ip2/ip2_index.cfm

[3] http://glite.cern.ch/

[4] https://glibrary.ct.infn.it/glibrary_new/index.php

[5] http://diligent.ercim.eu/

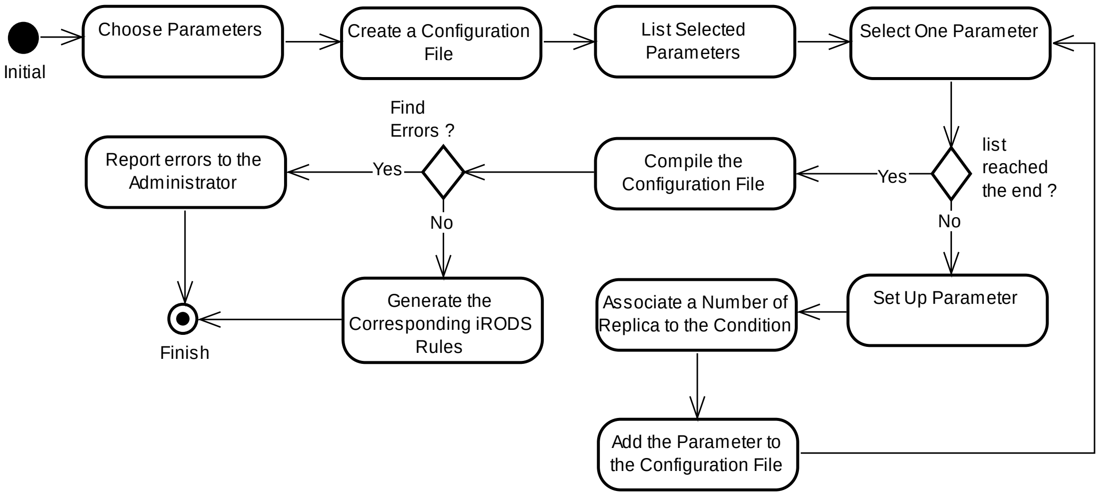
Initial
Choose Parameters
Create a Configuration File
List Selected Parameters
Select One Parameter
list reached the end ?
Yes
No
Set Up Parameter
Associate a Number of Replica to the Condition
Add the Parameter to the Configuration File
Compile the Configuration File
Find Errors ?
Yes
No
Report errors to the Administrator
Generate the Corresponding iRODS Rules
Finish

Figure 4 – Sequence diagram of the *Setup Replication Rules*

We propose an easy to use interface for managing replication rules so that business administrators without any specific technical skills can manage replication rules. That interface would support two use cases: *Setup Replication Rules* and *Manage Parameters*, which are represented in Figure 3.

In the *Setup Replication Rules* use case, the business administrator is able to create, in an easy way, specific rules based on determined parameters (e.g., file format, file size, user identity, etc.), in order to determine the minimum number of replicas to maintain of a file. The Parameters are listed as check boxes. The business administrator can select a list of parameters to be setup and then proceed to the configuration. When doing the configuration, for each parameter the business administrator can specify one condition (e.g., equal, different from, etc.) and enter a minimum number of replicas to be associated in case the condition is verified. The respectively configuration will be added to a configuration file. When finished, the configuration file is compiled and searched for any syntax errors. If no error is found, iRODS rules are generated and added to the rule base. Otherwise, the system will show the resultant errors. Figure 4 depicts the workflow sequence of this use case.

In the *Manage Parameters* use case, the system administrator can manage which parameters the business administrator can select and customize. The system can interpret certain parameters. The system administrator can choose from a previous list which parameters will be available to the business administrator. The system administrator can add or remove parameters to the actual list (the one that the business administrator can see), and apply the changes.

4.2 The Replication Service

Upon ingestion of a new file into the local iRODS deployment, the replication service checks if any of the rules configured by the business administrator apply to the file and, according to the rules, computes the number N of replicas to maintain of that file. That number is associated to the file through its metadata.

After that, the service checks the number of different federated grids. If the number of replicas is bigger or equal than the number of federated grids, a list of all federated grids is compiled. Otherwise, the first N federated grids listed are compiled into the list. Based on the list of federated grids available and on the number N of replicas, the number of replicas to be stored on the local iRODS deployment and on the remote federated grids is computed. The number of replicas to be stored in each zone, local or remote, is the integer division of N by the number of Zones. The remaining number of replicas until filling N is stored in the local Zone. The number of remote replicas R to be created is then registered and associated to the file metadata.

Then, one by one, each Zone contained in the federation list, will be used to create and store the replicas. If the local grid deployment is selected, the number of effectively created replicas L is registered and associated to the file metadata. If it is the case of a remote Zone, the file is copied and the number of remote replicas R that file should have is associated to that copied file. The file is not directly replicated, since the federation configuration does not allow the direct replication of files to a remote Zone. The file has to be copied and the replication has to be executed by the audit service running in the remote grid, which we will explain in the next section. Also, when copying a file to a remote zone, the associated metadata is not copied, hence the association of the desired number of replicas R after the copy. A reference to the file copied to the remote zone is also maintained. Figure 5 depicts the activity diagram of the replication service.

4.3 The Replica Audit Service

The replica audit service functions at two different levels. The first level audits the number of replicas stored in the local zone and the replicas of copies of local files stored in remote zones. The second level audits the number of replicas in the local zone which are owned by other zones (in other words, files which have been copied to the local zone from a remote zone).

Concerning the first level, a list of files contained in the local zone and owned by the local zone is compiled. Then, one by one, each file is selected and list of zones containing a replica of that file is compiled and, for each zone of the list, the number of replicas effectively stored in the selected zone is determined. When the list of zones is fully processed, if the total number of existing replicas is smaller than the replica number N contained in the file metadata, the number of necessary replicas in order to get N replicas is calculated. Then, a list containing all the zones where a copy of the file exists is compiled, and the number of replicas is increased to be as close as possible to L, in case of a

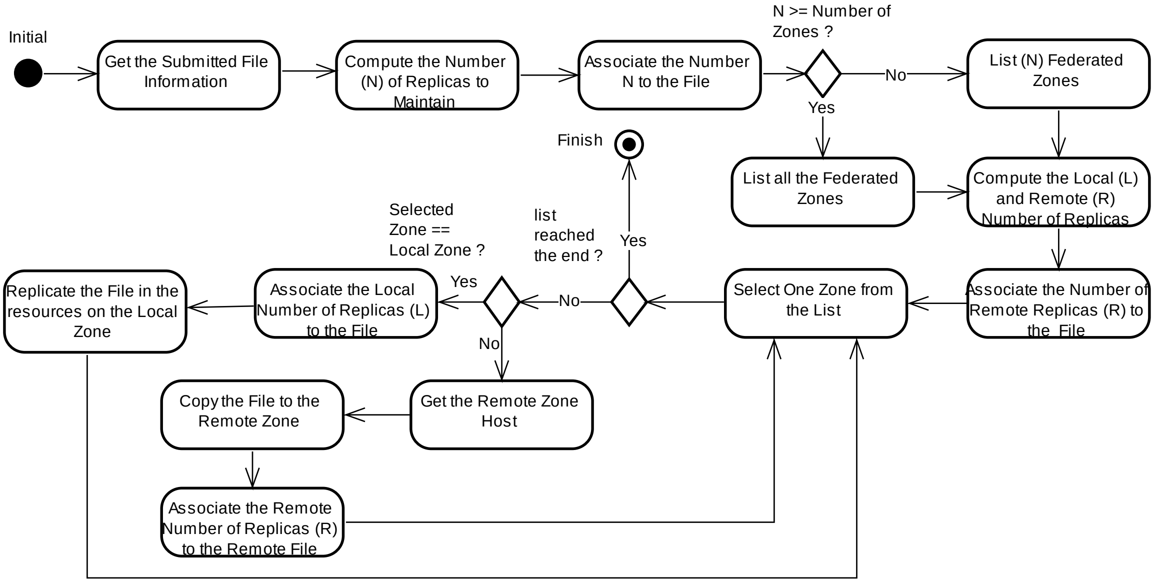
Initial
Get the Submitted File Information
Compute the Number (N) of Replicas to Maintain
Associate the Number N to the File
N >= Number of Zones ?
No
Yes
List (N) Federated Zones
List all the Federated Zones
Compute the Local (L) and Remote (R) Number of Replicas
Associate the Number of Remote Replicas (R) to the File
Select One Zone from the List
list reached the end ?
Yes
Finish
No
Selected Zone == Local Zone ?
Yes
No
Associate the Local Number of Replicas (L) to the File
Replicate the File in the resources on the Local Zone
Get the Remote Zone Host
Copy the File to the Remote Zone
Associate the Remote Number of Replicas (R) to the Remote File

Figure 5 – Activity diagram of the *Replication Service*

local zone, or to R in case of a remote zone. After all files have been verified, an audit report of the status of each file is compiled and sent to the system administrator, so that, in case of need, he can take informed decisions. Figure 6 depicts the activity diagram of the first level of the auditing service.

The second level of audit begins with the creation of a list of files contained in the local zone which are owned by remote zones. Then, one by one, each file is selected and the number of replicas that the file should have (R) is retrieved. The number of accessible replicas is retrieved and is compared with the desired number of replicas. If the number of accessible replicas is smaller than R, than the number of replicas needed in order to reach R is calculated. If the number of existing resources is smaller than the number of replicas needed to reach R, a list of all the resources that do not contain a replica is retrieved. If the list is empty, then the issue is registered for the audit report and another file is selected for verification. Otherwise, files are replicated throughout available resources. After all files have been verified, an audit report is compiled and sent to the system administrator, so that, in case of need (e.g., add more storage resources) he can take informed measures. Figure 7 depicts the activity diagram of the second level of auditing.

4.4 Implementation

The interface for the composition of replication rules was implemented as a website, using HTML and PHP. The user is guided through the configuration of the rules. Currently supported replication parameters are *file extension*, *file name*, *file size*, *user*, *submission date*, and *resource name*. When the composition of rules is finished, an xml configuration file is generated and is directly processed by a compiler.

The compiler is written in C. For each replication parameter it should verify the syntax and generate the corresponding iRODS rules. The compiler output is a set of iRODS rules as an iRODS rule base file so it can be included in the iRODS installation.

Both *Replication* and *Audit* Services are implemented using the rule mechanism and workflow capabilities provided by iRODS. The services are defined as a set of actions within a rule. The actions are composed by a set of micro-services. For the development of those micro-services we used the C language API provided by iRODS.

The *Replication* Service is triggered by a file submission. To access the file information we use available iRODS session variables. We send this information as input to the rules previously generated by the compiler. The execution of the rule results in a minimum number of replicas. This number is then associated to the file as a metadata attribute, using a micro-service already packed with iRODS.

When using resources located in remote zones for replication, the file has to be copied to the remote zone and the number of copies to maintain is associated with the file metadata. The remote zone then takes care of the replication. While we had to develop a set of micro-services to perform some of the operations involved, we also used micro-services already included in the installation.

The Audit Service is composed by two iRODS rules, one for auditing files owned by the local zone, and the other for auditing the files owned by remote zones. Again, some micro-services were developed to specifically for this purpose. Other micro-services were already packed with iRODS, such as the case of *msiSendMail*, which is used to send the audit report to the system administrator.

5. CONCLUSIONS AND FUTURE WORK

Data grids are systems which possess characteristics which are highly desirable for digital preservation, such as replication. Replication makes possible the adoption of a data redundancy strategy which is crucial to withstand failures. In addition, the federation configuration model present is data grids such as iRODS allows the interoperability between independent data grid installations, thus making possible the sharing of resources.

This paper presented a proposal based on the customization of the iRODS platform to be able to take advantage of its replication and federation features. That proposal was implemented with basis on the rule engine mechanism which allows the creation of rules that

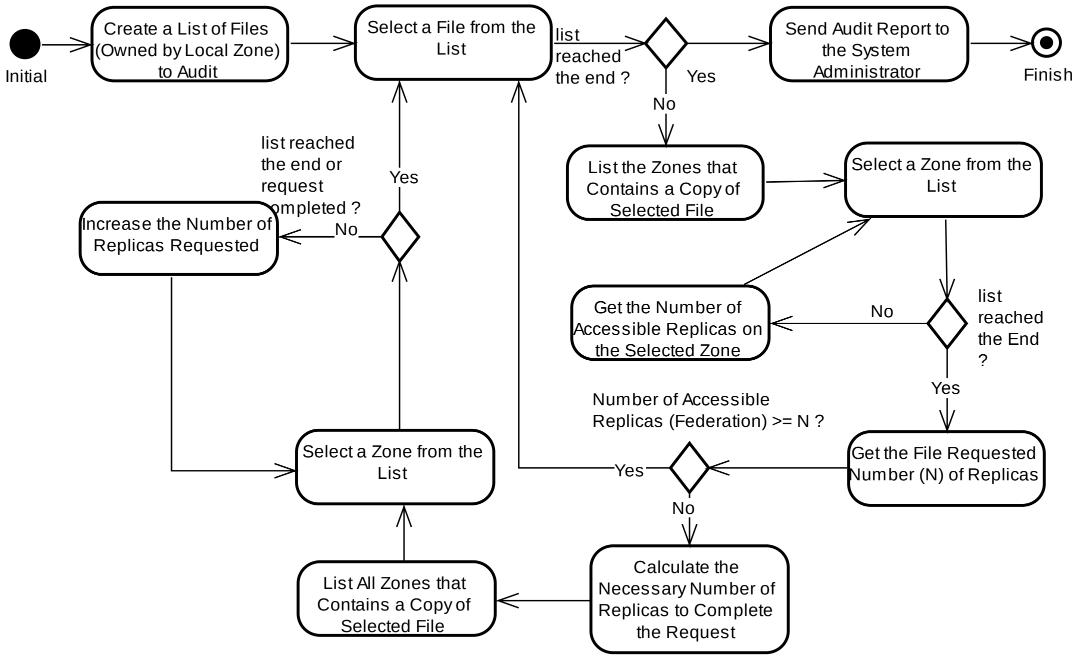
Initial
Create a List of Files (Owned by Local Zone) to Audit
Select a File from the List
list reached the end ?
Yes
No
Send Audit Report to the System Administrator
Finish
List the Zones that Contains a Copy of Selected File
Select a Zone from the List
list reached the end or request completed ?
Yes
No
Increase the Number of Replicas Requested
Get the Number of Accessible Replicas on the Selected Zone
No
list reached the End ?
Yes
Number of Accessible Replicas (Federation) >= N ?
Select a Zone from the List
Yes
No
Get the File Requested Number (N) of Replicas
List All Zones that Contains a Copy of Selected File
Calculate the Necessary Number of Replicas to Complete the Request

Figure 6 – Activity diagram of the *Audit of Locally Owned Files*

orchestrate micro-services. Through the creation of specialized micro-services, it was possible to create a complex replication service which takes advantage of the federation configuration, using the spare resources of federated grids to create replicas, thus using geographic distance and independent administration to lower the risk of losing data.

In addition to the replication service, an audit service was created, using the same mechanisms, which audits replicas at two levels: at the level of the locally-owned data, in which the data owned by a local grid, stored locally or remotely, can be audited; an at the level of remotely-owned data, in which the local grid audits data owned by remote federated grids, but stored locally.

Besides the implemented services, an interface for the composition of replication rules was also described. The use of a user-friendly interface would allow business administrators, with little technical knowledge, to define the rules applicable to the data. The kinds of rules that a business administrator can define have to be determined by the system administrator, more knowledgeable of technical aspects. The rules defined in the interface are then compiled into iRODS rules and included in the rules database.

Future work will focus on the validation of the proposed solution in the context of project SHAMAN[6] and TIMBUS[7]. Both projects address scenarios where the usage of data grids for preserving data assumes major relevance.

6. ACKNOWLEDGMENTS

This work was supported by FCT (INESC-ID multi-annual funding) through the PIDDAC Program funds and by the projects SHAMAN and TIMBUS, funded under FP7 of the EU under contract 216736 and 269940, respectively.

7. REFERENCES

[1] IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries," New York, 1990.

[2] Miles, S., Wong, S. C., Fang, W., Groth, P., Zauner, K.-P. and Moreau, L. 2007. Provenance-based validation of e-science experiments," *Web Semant.*5 (Mar. 2007), 28–38.

[3] Barateiro, J., Antunes, G., Freitas, F. and Borbinha, J. 2010. Designing Digital Preservation Solutions: A Risk Management-Based Approach. *The International Journal of Digital Curation*. 1, 5 (Jun. 2010), 4-17.

[4] Johnston W. E. 2002. Computational and Data Grids in Large-scale Science and Engineering. *Future Gener. of Comput. Syst*. 18, 8, 1085-1100

[5] Venugopal, S., Buyya, R., and Ramamohanarao, K. A. 2006. Taxonomy of data grids for distributed data sharing, management, and processing. *ACM Computing Surveys*. 38,1, 1–53.

[6] Rajasekar, A., Wan, M., Moore, R. and Schroeder, W. 2006. A prototype rule-based distributed data management system. In *HPDC workshop on Next Generation Distributed Data Management* (Paris, France).

[7] Barateiro, J., Antunes, G., Cabral, M., Borbinha, J. and Rodrigues, R. 2008. Using a GRID for digital preservation. In *Proceeding of the International Conference on Asian Pacific Digital Libraries* (Bali, Indonesia).

[8] Foster, I. 2002. What is a grid? A three point checklist. *Grid Today*. 1(6).

[9] Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., and Tuecke, S. 2000. The data grid: Towards an architecture for

[6] http://www.shaman-ip.eu

[7] http://timbusproject.net/

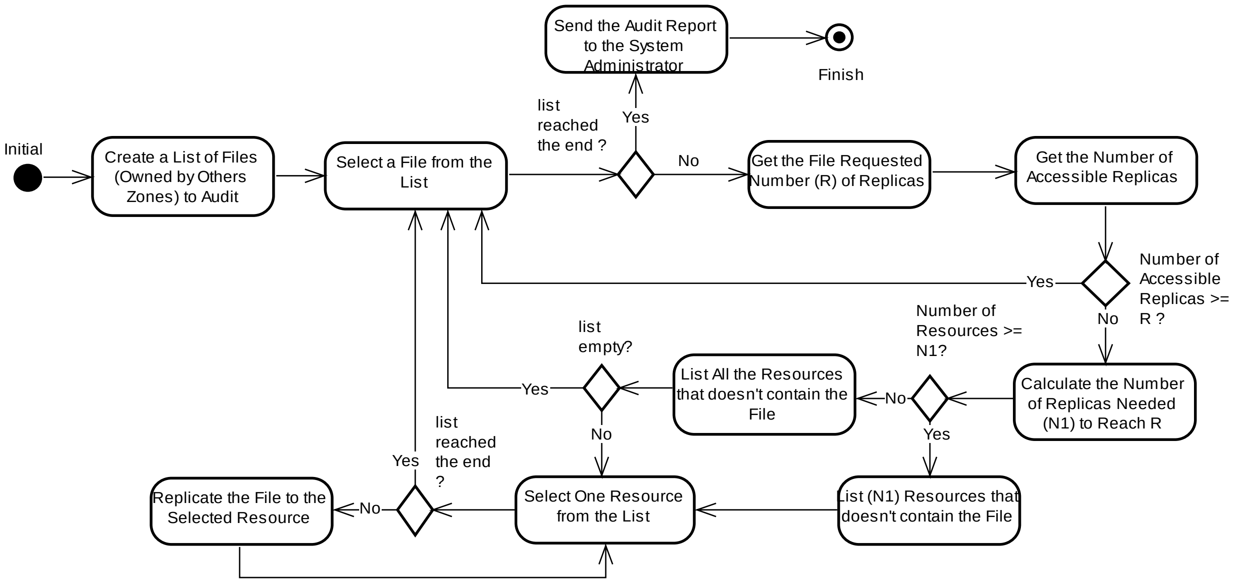
Send the Audit Report to the System Administrator
Finish
Initial
Create a List of Files (Owned by Others Zones) to Audit
Select a File from the List
list reached the end ?
Yes
No
Get the File Requested Number (R) of Replicas
Get the Number of Accessible Replicas
Number of Accessible Replicas >= R ?
Yes
No
Number of Resources >= N1?
Calculate the Number of Replicas Needed (N1) to Reach R
No
Yes
List All the Resources that doesn't contain the File
list empty?
Yes
No
List (N1) Resources that doesn't contain the File
Select One Resource from the List
list reached the end ?
Yes
No
Replicate the File to the Selected Resource

Figure 7 - Activity diagram of the *Audit of Remotely Owned Files*

the distributed management and analysis of large scientific datasets. *Journal of Network and Computer Applications.* 23, 187–200.

[10] Moore, R. 2005. Building preservation environments with data grid technology. *American Archivist,* 69, 1, 139–158.

[11] Rajasekar, A., Wan, M., and Moore, R. 2002. MySRB & SRB - components of a data grid. In *The 11th International Symposium on High Performance Distributed Computing* (Edinburgh, Scotland). (HPDC-11).

[12] Duranti, L. 2005. The Long-Term Preservation of Accurate and Authentic Digital Data: The InterPARES Project. *Data Science Journal*. 4, 25 (Oct. 2005), 106-118.

[13] Moore, R. 2007. General Study 01 Final Report: Building Preservation Environments with Data Grid Technology. InterPARES 2 Project.

[14] Jordan, C., Kozbia, A., Minor, D. and McDonald, R. H. 2008. Encouraging Cyberinfrastructure Collaboration for Digital Preservation. In *Proc. iPRES2008* (London, UK).

[15] Gao, J., Li, Z., Wang, X. and Zhu, C. 2009. Research on Grid Storage Technology and its Application in Digital Library. In *the 2nd International Symposium in Knowledge Acquisition and Modeling* (Wuhan, China).

[16] Calanducci, A. S., Barbera, R., Cedillo, J. S., De Fillipo, A., Saso, M., Iannizzotto, S., De Mattia, F. and Vicinanza D. 2009. Data Grids for Conservation of Cultural Inheritance. In *Proc. DaGreS ’09* (Ischia, Italy).

[17] Hedges, M., Hasan, A., Blanke, T. 2007. Management and Preservation of Research Data with iRODS. In *Proc. of CIMS ’07* (Lisbon, Portugal).

[18] Chien-Yi Hou, Altintas, I., Jaeger-Frank, E., Gilbert, L., Moore, R., Rajasekar, A. and Marciano, R. 2006. A scientific workflow solution to the archiving of digital media. In *Workshop on Workflows in Support of Large-Scale Science* (WORKS '06)

[19] Innocenti, P., Ross, S., Maceiviciute, E., Wilson, T., Ludwig, J. and Pempe, W. 2009. Assessing Digital Preservation Frameworks: the Approach of the SHAMAN Project. In *Proc. of MEDES ’09* (Lyon, France).

[20] Candela, L., Akal, F., Avancini, H., Castelli, D., Fusco, L., Guidetti, V., Langguth, C., Manzi, A., Pagano, P., Schuldt, H., Simi, M., Springmann, M. and Voicu, L. 2007. DILIGENT: integrating digital library and Grid technologies for a new Earth observation research infrastructure. In *Int. J. Digit. Libr.* 7, 59-80.