

Data grids in theory and practice

IBM Internship report

Wouter Alexander de Landgraaf

IBM Healthcare and Life Sciences
Vrije Universiteit Amsterdam

Dr. Nicky Hekster, supervisor at IBM
Prof. Dr. Henri Bal, supervisor at VU

April 4, 2008

Contents

1	Introduction	6
2	Biobanks and the AMC case	7
2.1	Data integration problems	7
2.2	Biobank requirements	8
2.3	Standards within Healthcare	9
2.4	AMC case description	10
3	Background: Grid Computing	12
3.1	History of the grid	12
3.2	Types of Grids	13
3.3	Usage of Grids	14
3.4	The future of Grids	16
4	Background: VL-e	17
4.1	VL-e Proof-of-Concept Grid Architecture	18
4.2	VL-e Proof-of-Concept Client Architecture	21
4.3	Privacy	22
4.4	Security in VL-e	23
4.5	Infrastructure Strengths and Weaknesses	24
4.6	Related Grid technology	25
5	Background: IBM's grid technology	27
5.1	Current IBM Grid involvement	27
5.2	IBM Grid Medical Archive Solution (GMAS)	28
6	Problem Analysis and the implemented solution	37
6.1	Alternative solution: local storage	37
6.2	Alternative solution: proprietary off-site storage	37
6.3	Implemented solution: off-site EGEE/VL-e grid storage	38
6.4	Deployment of the GAP/VGFS solution at the AMC	41
6.5	Issues encountered	43
7	Assessment and Testing	45
7.1	Grid certificates and Security	45
7.2	Privacy, Access Control and Encryption	45
7.3	Availability	46
7.4	Performance	48
7.5	Integrity	49
7.6	Integration with IBM products	50
7.7	Integration into AMC sequencing process	50

8	Conclusions	52
8.1	Recommendations to VL-e	53
8.2	Recommendations to AMC	54
8.3	Recommendations to IBM	55
8.4	Recommendations to developers using grids	55
A	Genome sequencer transfer script	58
B	VGFS technical details	60
B.1	Status of VGFS development	60
C	IBM Healthcare and Life Sciences products	63
C.1	IBM Content Management Offering (CMO)	63
C.2	IBM Clinical Genomics Solution (CGS)	69
D	Other internship activities	75
D.1	VL-e Medical Presentation	75
D.2	Project "Zorgkonijn"	75
D.3	Literature Study	75
D.4	AMC and UvA	75
D.5	IBM, workshops and other activities	76

Acknowledgements

Within IBM Healthcare and Life Sciences I have been working under the supervision of Nicky Hekster, technical lead of the division within IBM Netherlands. At the Vrije Universiteit my supervisor was Henri Bal, head of the Parallel and Grid Computing group. Within the VL-e project Silvia Olabariaga, project lead of the VL-e Medical group, has been my supervisor concerning usage of the VL-e infrastructure and sparring partner for the AMC case. At the AMC we have worked with Frank Baas, Jan-Willem van der Wal, Antoine van Kampen and Barberra van Schaik of the neurogenetics and bioinformatics departments in order to investigate the problems concerning the case, discuss possible solutions and implement the solution we concluded was the most appropriate. I would also like to mention René Verheij, IBM IT specialist and IT coordinator of the DIO project at the AMC. I would like this opportunity to thank everyone above for their cooperation and insight, which I wouldn't have been able to complete my internship or this report without.

Abstract

This document describes the project completed during my internship at IBM Netherlands. At the AMC we solved a data storage and management problem using the Dutch VL-e grid infrastructure.

In order to do so we developed a Linux-based grid filesystem interface layer, allowing the grid to be easily integrated into an existing IT infrastructure using standard network protocols. This grid filesystem layer (VGFS) was developed using the FUSE kernel module and it interfaces with the gLite data management APIs used within VL-e. VGFS also offers transparent encryption when sending files to the grid, allowing sensitive data to be stored securely. A number of IBM products have been selected to complement VGFS. The resulting system was deployed at the AMC in order to solve the data storage and management problem.

Our conclusion, after the assessment of the deployed system, is that although we have improved usability and security when using the VL-e grid infrastructure the unreliability of grids hamper their usage within a production environment. Nevertheless grids can provide resources in a useful way and with continued effort can become a useful part of future IT solutions.

1 Introduction

Over the course of the previous 6 months, I have been able to follow an internship at IBM Healthcare and Life Sciences for my Computer Science Master Course of the Vrije Universiteit, Amsterdam (VU). During this time I have been able to investigate the topic of Grid Computing, communicate with a number of other parties in order to come up with a limited pilot project in the healthcare sector, analyze the possible solutions for this pilot project, develop a solution using facilities of the VL-e grid infrastructure and deploy and test this solution in a production environment. The conclusion of this project will lead to an evaluation of the current situation of grid computing, what tasks it is suitable for and if it is usable in a production environment with non-technical users. This is the report of my findings.

The project consisted of both a theoretical study and a case study within the healthcare sector. After discussions with the VL-e Medical project group, researchers at the Amsterdam Medical Center (AMC) and my supervisors at IBM and the Vrije Universiteit we agreed on the following pilot project: At the AMC a new genome sequencer has been installed which vastly improves the speed and accuracy of DNA sequencing. This sequencer in the neurogenetics department, of which there only are two similar devices in the Netherlands, offers many new possibilities to researchers and clinical specialists. There are however issues due to the large amount of data that is required to be stored after each run of the sequencer. During my internship it has been my task to solve this data storage and management task using the VL-e grid infrastructure and IBM products and offerings.

The scientific contribution of my internship is two-fold: From a theoretical point of view it is useful to have an overview of the current state of grids and the VL-e data infrastructure in particular. From a practical point of view my contribution to VL-e is the development of a secure grid-filesystem interface layer and the development, deployment and assessment of a solution for grid data storage and management in a production environment.

The rest of this report is divided into the following sections: The following four chapters (2 to 5) describe the topics of Biobanks in combination with the chosen case at the AMC, Grid Computing, VL-e and IBM's grid technology. In Chapter 6 we will analyze the problem described in Chapter 2, discuss different solutions in solving the problem using the background of the four topics discussed and describe the solution we developed and implemented at the AMC. In Chapter 7 we will assess the implemented solution and discuss our test results and experiences. In Chapter 8 we will show which conclusions can be drawn from this project and the lessons that can be learned from it. After this appendices and references are included for further study.

2 Biobanks and the AMC case

In this section we look at the area of Healthcare and Life Sciences and biobanks in particular. We then relate this to the chosen use-case for the project. In Chapters 6 and 7 we analyze this case using the background from the following chapters, look at the possible solutions and describe the solution we implemented.

From treating symptoms to being able to sequence the DNA of the human genome, biological and medical research has made tremendous leaps in the 20th century. This trend is expected to continue with increasingly more effective scanners and a better understanding of genetics and proteomics, leading to personalized medicine and an improved quality of life.

Problems due to the increasingly more sophisticated tools and the resulting explosion of data are however starting to pop up in the clinical and research departments of hospitals and universities. The most immediate problem is that storage requirements are ballooning due to larger data sets and having to store these sets for many years. A more fundamental problem is that researchers need to be able to use the generated data in an efficient way, and correlate this data with many other sources. Existing LIMS (Laboratory Information Management System) and PACS (Picture Archiving and Communication System) systems are however often incompatible with each other and the data that exists often isn't accessible to researchers working at other locations.

Hospital information systems are struggling to cope, legacy systems contain large amounts of data in proprietary formats and researchers and doctors have to deal with inefficient processes that haven't been modernized to make effective use of modern information technology.

An information biobank provides a single efficient system to combine data from multiple existing sources, giving researchers the ability to easily query large datasets and administrators to centralize data into one storage pool.

2.1 Data integration problems

From a technical point of view, a biobank is strictly a data integration problem[21]. This means that the sources use different data models and formats, different ontologies, have requirements on anonymization and privacy and that the resulting biobank requires access control and means for auditing.

- Differing data models and formats

Currently data standardization in the clinical environment is only catching on slowly. Although DICOM and HL7 provide accepted formats, many legacy systems instead use relational databases or spreadsheet-based models, if the data is at all accessible. Also, versions of the accepted standards can be incompatible, leading to problems even in comparably recent systems. Medical data can come in all shapes and sizes, ranging from data entries and text files to huge amounts of images.

- Differing ontologies

If in an ideal case all data were in compatible formats, another issue still exists: ontologies and taxonomies frequently differ, which in the highly-specialized medical domains leads to challenging problems when a researcher tries to merge data from multiple domains.

- Anonymization and privacy

For ethical, moral and legal reasons, data integration means anonymization of this data. Personal information is not to be known automatically, even if access to the medical data is required. The data collected must also be legally available to a researcher, as consent is required by the participant for usage of the data beyond a clinical setting.

- Ownership and auditing

Even if the data is anonymous and usable, it is important for medical institutions to be able to control access. It is also required to have an audit log of which users had access to what data.

2.2 Biobank requirements

There are two main problems that biobanks have to solve: the ability for researchers to combine multiple data sources into one coherent system for querying and to allow data storage and archiving of the different systems.

Further functionalities required are: the ability for administrators to monitor and control the biobank, the functionality to allow different users to have fine-grained access to different data sources and the ability to have access to different incompatible legacy systems via a single interface.

Security and Authentication

The type of data stored within biobanks is highly sensitive and must be securely transferred and stored. A high level of encryption is required, and the data integrity of data objects must be safeguarded.

Authentication has to be possible both on an individual level as well as on the level of a (virtual) organization.

Privacy

Because of both legal and ethical reasons, patient data that is sent outside the original organization has to be anonymous to a very high degree. Patients themselves also have to give consent for their results to be used for research purposes. It is currently unknown if clinical data without consent is allowed to leave the organization at all, even with strong layers of anonymity and de-identification.

Scalability

Biobanks have the possibility to increase exponentially due to both ever increasing amounts of data and the inclusion of new data sources in the future. Any biobank solution must be scalable and allow seamless upgrading and expansion.

High Availability

Depending on the purposes of the biobank, high availability must be guaranteed. For research purposes alone, a very high degree of availability is not critical, however in a clinical setting the biobank must be available at all times. Redundant deployment of all layers in a biobank solution is required.

Performance

How much a biobank will actually be used is a difficult question up-front. If performance is to be improved, then a high degree of scalability is essential. It is important to know at which level of usage the performance of the biobank can be guaranteed, thus the performance of all layers must be tested up-front.

Maintainability

In an ideal situation a biobank would require little maintenance; however this too depends on all the layers in the biobank solution. Existing test cases can provide a method of determining how much maintenance is required.

Auditing

As mentioned before, the ability to audit the biobank must be possible. It is essential to be able to determine which users had access to what data.

Ease-of-use, Ease-of-integration

In the medical domain users are known for their ability to save lives, but not for their flexibility when it comes to new technology or processes. Researchers should be able to use the biobank via a better interface than previously used, and existing users should be able to use the legacy systems without problems.

Integration of a biobank must be with as little downtime and as few problems as possible. This is especially the case with systems that are in use in a clinical setting.

2.3 Standards within Healthcare

DICOM

The Digital Imaging and Communications in Medicine (DICOM) standard is published by the National Electrical Manufacturers Association and aims to unify all protocols used by imaging equipment in the medical domain. It also addresses interoperability and exchange of digital information between medical imaging equipment and other systems via network communication and by defining the DICOM File Format.

The DICOM File Format is unique in that it combines imaging data with meta-data into a single file, thus ensuring that the correct patient information is always sent together with the medical images.

Usage of the DICOM Standard facilitates implementations of PACS solutions, however it doesn't guarantee interoperability. Although DICOM is mostly backwards-compatible, different manufacturers of PACS solutions have created incompatible systems with each claiming to conform to the DICOM standard.

HL7

The HL7 organization develops standards, guidelines and methodologies to allow exchange and interoperability of electronic health records. By implementing the HL7 standards, it is possible to have lab information systems, radiology information systems and hospital information systems communicate with each other.

HL7 is a broad set of rules to simplify data exchange, not a standard that was designed to be followed to every detail. In the words of [4], HL7 is an 80% standard, leaving the remaining 20% of the implementation to be decided by the healthcare facility's unique demands and requirements.

HL7 has a high percentage of adoption in healthcare facilities; however the drawback is that various HL7 implementations are often unable to communicate directly with each other. To make matters worse, different versions of HL7 standards contain new features and options, breaking backwards-compatibility. The use of HL7 interface engines facilitates the linking of different systems and the deployment of these interface engines is a common situation.

2.4 AMC case description

A full-scale biobank is out of the question for an internship project, however we are able to look at one aspect within the biobank requirements: storing large datasets that have to be accessible from multiple locations. Via Prof. Frank Baas from the neurogenetics department at the Amsterdam Medical Center we are able to test the datagrid properties of the VL-e grid infrastructure.

The neurogenetics department has a new DNA-sequencer (Genome Sequencer FLX System[12], Figure 1), with the capacity to sequence 400000 reads (with each read being a chunk of up to 300 DNA bases) in a single 7-hour run. Using a combination of adding nucleotides to each of the samples, so creating a chemical reaction, and high-resolution imaging techniques, the Roche Genome Sequencer FLX System is able to obtain a large amount of information from a DNA sample. After data analysis the obtained amount of information (the DNA sequence) is fairly limited, however as this is a relatively new technique and as geneticists are still actively improving the data analysis results it is required to store the raw image data as a backup for future processing.

The amount of raw data generated by each run is about 14GB. With 2 runs a day, the amount of data storage required is very large (10TB/year in the case of continuous usage). If the Genome Sequencer FLX System were to be able to

Figure 1: Genome Sequencer FLX [12]



store the data on the Big Grid infrastructure then this would prevent a lot of extra costs and headaches for the AMC IT department.

A lack of data storage in general is becoming a problem for the AMC. This would be a use-case of limited scope, but with the potential to become a useful component in the AMC IT strategy for online Long-Term Storage.

The data that is to be stored is research-data and has already been made anonymous, so privacy isn't a direct issue. Data integrity, access control, access auditing and security naturally are important factors. The main factors however are the speed at which this data can be transferred to and from the Grid infrastructure and the usability of the system, as researchers have to be able to access the raw data themselves. In the future this data has to be accessible from multiple sites, however for the initial case this isn't a requirement. Furthermore, the production environment at the AMC will provide a useful testbed for determining the usability of the grid in a less academic setting, with users that aren't IT specialists or programmers.

3 Background: Grid Computing

In the last 10 years grid computing has grown from a purely academic challenge to commercially useful technology. Where previously data was stored isolated on servers, the rise of high-speed Internet connections has made many options for interconnection possible. In recent years there has even been a considerable amount of hype surrounding grids, which has lead to both very large amounts of private and government funding and (inevitably) unrealistic expectations and disillusion.

In this section we take a look at what grids actually are, how they are used and in which direction grids might evolve. This section serves as a background against which the project has taken place.

3.1 History of the grid

Using multiple computers for a single task isn't new. Already in the 50's and 60's, multiple hundred-thousand-dollar computers were connected together to speed up computations. Considering the speed at which those computations took place, it was almost a requirement in order to get more than the most basic of calculations done within a reasonable amount of time. Over the last 60 years computing resources have grown at an exponential rate [25], however there has always been a need for more. With ever-larger amounts of scientific data, there is a strong need for distributed systems and distributed computing [16, 24]. Termed "grid computing", over the last 15 years researchers, developers and industry have come together in order to allow computing resources to be used, managed and allocated across organizations.

In the early 90's Ian Foster and Carl Kesselman [10, 25] laid out their ideas on what the future of computing should be. Instead of everyone having their own processing power and data, people would be able to 'plug in' to a grid infrastructure that would give users access to these resources, analogous to the electricity grid infrastructure. Foster and Kesselman would go on to be key members in the Open Grid Forum [28], a group of academic, community and industry partners that work together in order to standardize the grid. Foster and Kesselman have driven the development of the Globus Toolkit, the current de-facto grid middleware.

One definition by Foster [26]: "A Grid is a system that coordinates resources that are not subject to centralized control using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service"

From a completely different point of view, community or scavenger grids became popular in the late 90's. These community-driven projects let users donate unused CPU cycles for various tasks, like distributed encryption cracking [7], protein folding [31] and the search for extra-terrestrial life [23]. These voluntary projects gathered large groups of users, each group trying to out-do the rest in the amount of work units they could complete.

3.2 Types of Grids

Even though there are many definitions on what a grid is, each leads to different interpretations on what does and what doesn't constitute a grid. In this section a number of infrastructures are detailed that have been coined a "grid":

Cluster Grids

A cluster grid is a number of interconnected computers located physically together. They are used and administered by a single organization and have a dedicated high-speed local area network that link them together in order to work on a single task.

Certain people argue that supercomputers and mainframes also fall within this "grid" definition. Such a grid provides dedicated computer resources to the organization, and although it is separated from other networks by a single firewall it still is a combination of multiple computers. This type of grid doesn't comply with Foster's definition, as this grid is both under centralized control and physically at a single site. Primarily because of the latter we consider this infrastructure to be a single-site distributed system, but not a grid.

Distributed Enterprise Grids

A distributed enterprise grid constitutes a number of computers sharing resources within a single organization, however these computers are physically separated from each other. A group of computers at one site work together with computers at a second site to provide a single payroll-system, for example.

This type of grid is often connected with a dedicated network or it uses the Internet via a virtual private network to ensure security.

Managed Hosted Grids

A managed grid is a dedicated grid, hosted and located at a third party. This grid is provided as a service to the organization, without the need to physically install and manage the grid within the organization.

The architecture used at the hosting organization can be either a centralized (cluster) grid or a distributed grid. Access to this grid is then obtained via a virtual private network, a web service or another means of remote access.

Utility Grids

This type of grid comes closest to the ideas of Foster and Kesselmanns, where the grid used is hosted and located at a third party, but isn't dedicated to your organization. As with a managed hosted grid, the computers are installed and managed by the grid service provider, and access to this grid can happen in a variety of ways.

The main difference is that the grid service provider has many organizations using its grid simultaneously. The grid of the grid service provider can also be

located at other organizations, with the grid service provider being the broker between the organizations that require computer resources and the organizations that have underutilized computer resources.

Scavenger and Community Grids

This type of grid is made up out of many normal computers which work on a single, easily distributable task. When a computer becomes idle it requests an amount of work from a central server, and when the work is complete the results are sent back to the central server. The computer resources are only used when its user doesn't need them, and although the resources of a single computer can vary widely the total amount of resources of a large grid remains relatively constant.

Although there is a central organization required to give out and receive the work units, the actual resources are widely distributed across the Internet. No amount of service can be guaranteed, however on average the amount of work that is completed is either stable or growing.

The main problem with this type of grid is that the work tasks have to be easily distributable, and each application has to be written with this in mind. Another problem is that there can't be any communication between work units, as it is difficult to get the distributed computers to locate each other without the risk of either computer stopping its work or with the communication getting out of hand. A third problem is that participation is voluntarily, although this does foster a community around the grid getting initial momentum to drive such an application is difficult, even with standardized toolkits.

Even with these problems, community grids do provide a world-wide collaboration effort of many individuals who give their computer resources for a common goal.

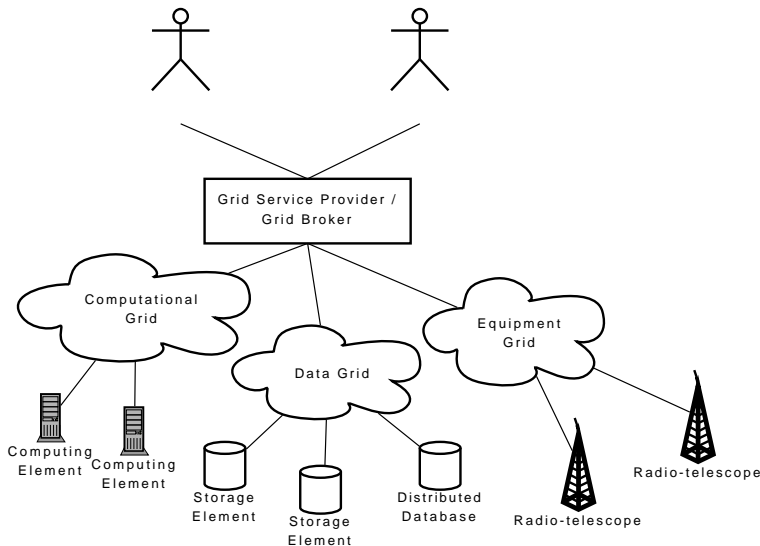
3.3 Usage of Grids

The type of grid specifies what usage is possible, but even so there are many applications possible where multiple computer resources are required. We will limit ourselves to the resources that such a grid application could access. An overview of grid usage can be found in Figure 2, where users obtain access to various resources via a grid service provider.

Computational Grid

The most-often used grid resource is pure computational (CPU) power. In many domains, researchers require raw computational power to solve complex problems, for instance in weather simulations, solving engineering problems, molecular modeling and diagnosing medical conditions [10]. Although processor speeds have gone through the roof (a PC today has more computing power than a supercomputer 20 years ago), there is always a need for more faster computational resources [13].

Figure 2: A conceptual overview of usage of a grid



A computational grid solves this need. Instead of requiring dedicated large clusters or supercomputers, organizations could request a larger amount of computational resources for a limited amount of time. Such an organization would be much more flexible in outsourcing large computational tasks, especially if these resources were only occasionally required. For current clusters, numbers of less than 30% utilization are not uncommon, leading to a waste in electricity and manpower in order to serve an occasional need.

Computation is probably the most frequent type of usage of a grid, with tasks that require only a small amount of data and a large amount of computation. All the mentioned grid types can provide computational resources, however depending on the tasks to be performed certain grids are more suited for certain types of tasks.

Data Grids

Another usage of a grid is as a data storage resource. Projects like the Large Hadron Collider will generate petabytes a year [9, 22], all of which have to be stored and available for researchers all over the world to access. Also in other research areas, for example in health care and bioinformatics, many terabytes of data are generated every year that possibly have to be stored for generations [18]. Each research center or hospital could build their own clusters and data repositories, however with the ballooning amount of data and the cost involved they are running into problems.

A data grid provides a means for high-throughput, high-capacity, long-term data storage. Additional possibilities are the accessing of this data by a group

of users and organizations, combined into a single Virtual Organization (VO). A key functionality of the data grid is that users are able to store and retrieve data from the grid without having to know where it is located physically. Data storage is virtualized; the files stored can be located (and replicated) anywhere across the data grid.

On a lower level, data grids provide various types of data: Flat-files, SQL and XML query-results. Depending on the type and size of data, different grid services are required. For instance, accessing the data grid as a database requires fast querying and retrieval of results, while file-retrieval requires a global namespace, fast file storage and retrieval and possibly direct input/output operations.

Scientific instrument Grids

A grid resource can also be specialized, like access to a particular high-cost piece of equipment. Access to such instruments via a grid infrastructure is currently rare, but allows additional benefits combined with data and computational resources. An array of telescopes could be accessed, with the data stored and processed using a single grid infrastructure, with the end-user obtaining the processed results. An example of a scientific instrument grid is the LOFAR project, a distributed radio telescope where the results are processed on an IBM BlueGene cluster grid.

3.4 The future of Grids

Obviously there are very large differences between the analogy of a grid infrastructure for electricity and for computer resources, and it would take a true optimist to think that such an infrastructure for computer resources would ever be as easy to use.

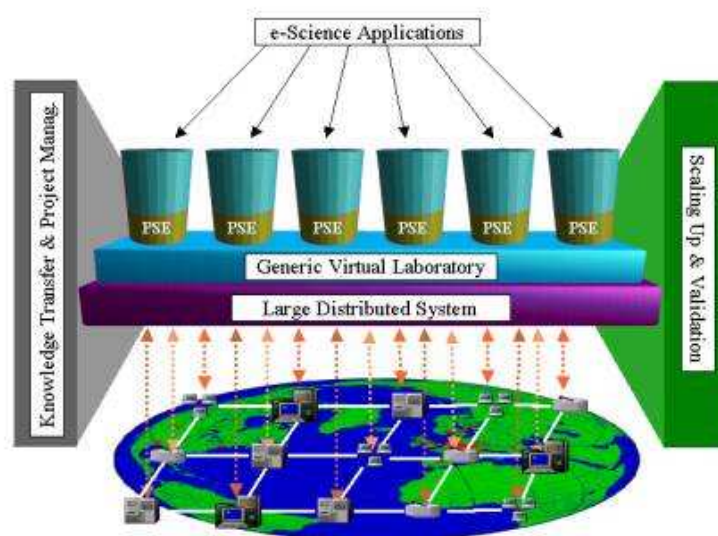
Currently there are many different interfaces and protocols. Despite efforts of the Open Grid Forum and the Globus Alliance, different standards and toolkits are used in different organizations, often more due to political and historical reasons than true technical reasons. Although there already are a number of large grid infrastructures, there isn't a single global grid that users and suppliers can freely and easily connect to.

Still, strong efforts of global-scale scavenger grids, the Globus Alliance, the Open Grid Forum and the development of grid protocols based on web standards are all indications of grid technology leading to a higher degree of convergence. Non-academic usage of grids does take place, for example the use of Globus by IBM in the financial services sector. There are still major obstacles to be tackled, most of which extra funding is unlikely to help, however large data centers are already openly discussing the possibilities of becoming a grid service provider. The main question will remain if there are enough potential users out there to grow such an infrastructure, or if usage of grids will remain in a niche.

4 Background: VL-e

The Virtual Laboratory for e-Science (VL-e) [29] project was started in order to boost e-Science by building generic software for research. By using this infrastructure, researchers and scientists from various disciplines are able to do their research faster and more efficiently. VL-e is backed by both academic and commercial partners (including IBM), and strives to provide a complete research infrastructure for both public and private organizations [17]. An overview of the VL-e project can be found in Figure 3.

Figure 3: Overview of the VL-e project

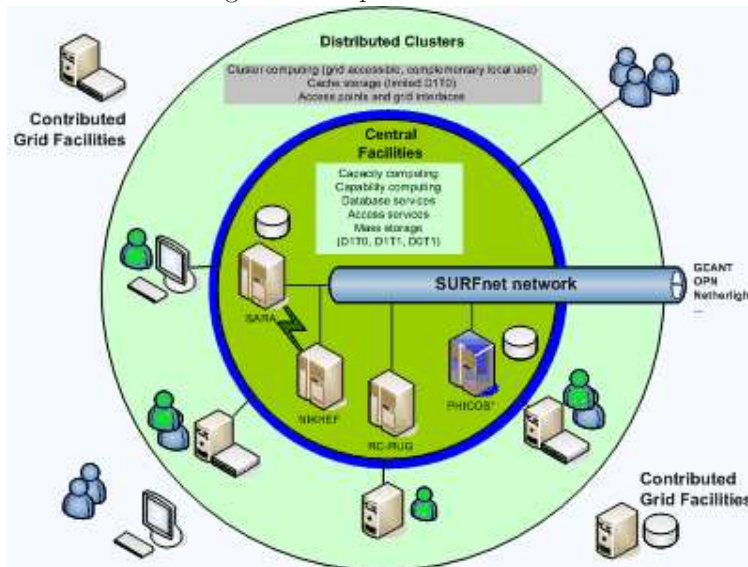


The recently-launched BIG GRID project [5] has been split out from the VL-e project and is focused on developing a sustainable nation-wide grid infrastructure. BIG GRID provides additional resources to build this infrastructure, in particular for large-scale data management and data storage purposes. In the rest of this document we will talk about VL-e, but in the future BIG GRID will fund the actual infrastructure. A schematic of the infrastructure can be found in Figure 4.

Within the VL-e project there are two separate grid infrastructures. The Distributed ASCI Supercomputer (DAS) grid infrastructure [2] is used for computer science research between a number of Dutch universities, while the EGEE/gLite-based grid is hosted by other parties in order to provide a proof-of-concept infrastructure upon which other types of research and projects can take place. During this project the primary focus has been in using the latter grid infrastructure in order to assess if it is mature enough for production use.

The EU-initiated Enabling Grids for E-science (EGEE) project [11] comprises of more than 240 institutions from 45 countries, and is focused on develop-

Figure 4: Proposed BIG GRID infrastructure



ing a seamless grid infrastructure between all members. The key sciences driving EGEE are High-Energy Physics and Life Sciences. Via EGEE, researchers currently have potential access to over 41000 CPUs and 5 petabytes of disk space. EGEE provides the grid middleware (gLite) used by the VL-e / BIG GRID projects and links these projects to many others across the world.

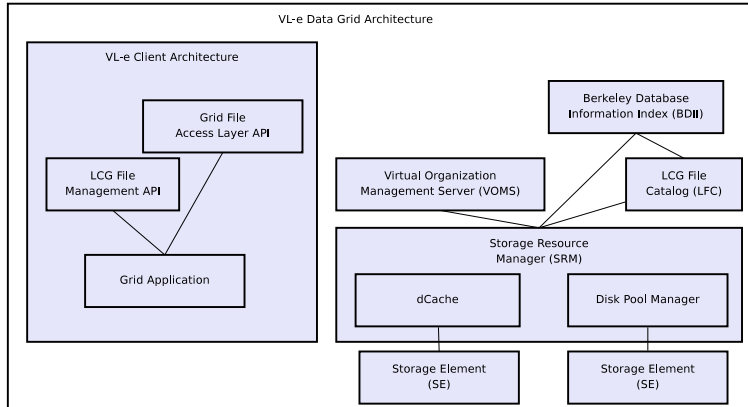
For this section we will limit ourselves to the **data-grid infrastructure**. We will show the various components of the current VL-e grid infrastructure, the possibilities and functionalities that this grid infrastructure has. We will then look at the strengths and weaknesses of this infrastructure from a technical point of view for data storage and management.

The software currently used in the VL-e Proof-of-Concept project is composed of a Proof-of-Concept client image, with all the software necessary to access the grid, and a grid infrastructure composed of various servers and services. Together, they allow users to run programs and store data on the grid. The architecture of both the client and the grid infrastructure can be found in Figure 5.

4.1 VL-e Proof-of-Concept Grid Architecture

The VL-e grid infrastructure is composed of a number of different libraries and services, that together provide a complete architecture for data management and data storage. We will describe these libraries and services top-down, starting with the libraries the client interacts directly with and ending with the storage elements that store the data on disk or tape.

Figure 5: The architecture of VL-e Proof-of-Concept grid architecture



Berkeley Database Information Index

BDII is the grid information service. It is this service that the grid client first accesses when communicating with an EGEE grid. It contains the locations of file catalogs, storage resource managers and other accessible grid resources. In the most common setup, a hierarchy of BDII-servers are available that each maintain control over a subset of these grid resources, with requests propagating between BDII-servers. In functionality they are comparable to an LDAP server.

LCG File Catalog

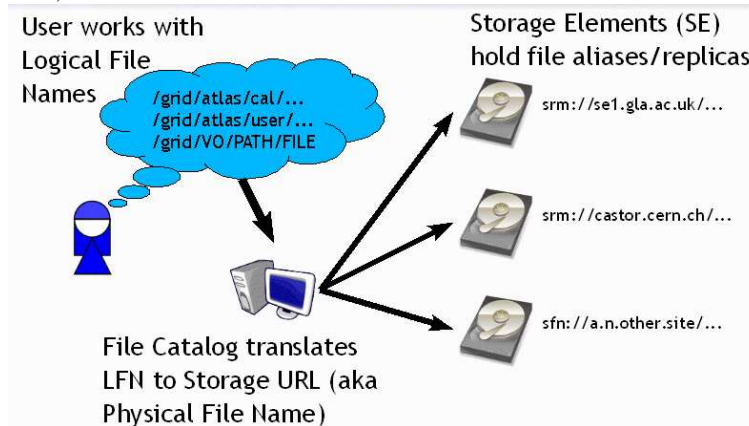
The LCG File Catalog (LFC) grid service provides the means to look up files and directories via a single Logical File Name (in URL syntax: *lfn:/grid/vlmed/alex/my_file.txt*). A file on the VL-e grid can be at any of the Storage Elements used, or even at multiple Storage Elements in the case of replication. Via the LCG File Catalog, the client can use a single naming structure without having to know where the file is actually stored. The LCG File Catalog hands a site-specific URL back to the client application, with which the application can access the remote file using the Storage Resource Manager interface. LCG is an abbreviation of the Large Hadron Collider Computing Grid project at CERN, where this grid service was developed.

An example of this translation can be found in Figure 6.

Storage Resource Manager

Storage Resource Managers are the key components in a datagrid infrastructure. Their task, as their name implies, is the management of large distributed datasets across the grid, handling the storage of files sent from the grid clients, the access a particular grid user has, the retrieval of files from their locations on either tape or disk and the allocation of space and quotas for users. The Storage Resource Manager itself doesn't transfer the files, instead it interacts

Figure 6: LCG File Catalog and Storage Resource Manager (image from GridPP wiki)



with other components (like GridFTP or RFIO servers) in order to perform the actual file transfer.

A Storage Resource Manager (SRM) has to be deployed at every site that wishes to be part of the data grid. Each SRM manages a Storage Element with one or more physical disks. The SRMs are contacted by the grid client after a Logical File Name has been translated to a filename at a particular Storage Element.

It is important to note that there isn't a single Storage Resource Manager; An Open Grid Forum Working Group (Grid Resource Management WG) [15] has drawn up a specification of all the functionality required in a SRM and there are a number of implementations, of which DPM and dCache are the two used within the VL-e context. These will be looked at below.

Disk Pool Manager

The Disk Pool Manager (DPM) is a Storage Resource Manager written by CERN and is aimed at smaller grid sites (Tier-2 sites of the LCG/EGEE grid) [14]. It supports multiple SRM versions, GridFTP and RFIO and authentication and authorization.

dCache

The dCache project was started by the Deutsches Elektron-Synchrotron and Fermi National Accelerator Laboratory and is aimed at larger grid sites (Tier-1) [6]. It is capable of managing petabytes of storage and supports the Storage Resource Manager standard. In comparison to DPM it also offers an optimized throughput and fault-tolerant service, and it can interface with storage managers like Enstore, Open Storage Manager and IBM's Tivoli Storage Manager, thus providing the means to integrate large amounts of existing storage.

CASTOR

A third SRM in use within EGEE is CASTOR, CERN Advanced STORage manager [3]. It is currently less frequently used compared to dCache and DPM and it is not used within VL-e. At SARA dCache is currently used as the SRM, while at other locations in VL-e DPM is used. For a developer the differences normally aren't an issue, however older "classic SE" grid storage (not using an SRM) limits the functionality. These storage elements are being phased out and SRM-accessible storage is widely deployed within VL-e.

Storage Element

By this term a server is meant that provides data storage resources. In order to provide advanced storage management a Storage Resource Manager is required, with which the storage element is connected.

Virtual Organization Management Server

The Virtual Organization Management Server is used within the VL-e grid infrastructure to group the users of grid resources into multiple Virtual Organizations (VO). Each VO contains members from a number of normal organizations and are subdivided depending on access control needs and area of research. The VL-e Medical Virtual Organization contains members from universities, hospitals and companies that share information with each other. Resources are allocated depending on the Virtual Organization; a company might be willing to offer data resources only if they are used within the area of interest, for example.

A non-technical aspect of a VO are (possibly) semi-regular meetings and discussion groups, allowing people to coordinate research, build trust and learn from each other both in the area of research and the use of grid resources. VOs are indispensable from a technical, organizational and a collaboration point-of-view.

4.2 VL-e Proof-of-Concept Client Architecture

The VL-e client image provides a complete CentOS Linux distribution, together with all the tools in the following diagram. This provides researchers an easy starting-point for trying out the grid infrastructure and porting their application to make use of this infrastructure. The distribution provides a number of visualization, workflow and data mining toolkits that can be used [30]. There are two interfaces available which developers can use to access the grid data infrastructure.

LCG File Management API

The LCG File Management API and utilities provide a means for querying the LFC and transferring files to and from the grid. The utilities and the API are

closely related and are limited to use on a Linux-based workstation. The API is closely mirrored to the default Unix POSIX system calls, but operates in subtly different ways. Transferring files using the default options means the operations are performed using GridFTP (a multi-stream secure FTP protocol).

The main problem with the API is that a lot of 'grid logic', i.e. accessing the LFC, communicating with SRM, is located within the client API. The API uses a number of database protocols for accessing different types of LFC servers, for instance. This is a problem because porting this API to other platforms or to other languages becomes increasingly difficult with each additional protocol. A possible solution would be to create a single web service that allows access to the grid via a single protocol, but this might decrease performance.

Grid File Access Library API

Next to the LCG File Management API there is also a separate Grid File Access Library API. This library provides a single interface for RFIO (Remote File Input/Output) commands. This allows a grid client to open, read and write portions of a file, instead of having to retrieve the file entirely.

RFIO and GridFTP have been compared at CERN. RFIO provides a much lower latency, while GridFTP provides a much higher throughput. Details can be found in [19].

4.3 Privacy

Privacy on the grid is always an issue of trust. There always has to be a degree of trust in an external party where the resource is physically hosted. For medical deployments, a hospital has to be aware that the data is leaving the internal network.

There is a difference between research data and clinical data, with the former patients have given their consent that the data can be used for research purposes, with the latter this is not the case.

In practice, research data stored on the grid is retrieved only for research purposes by the researchers running the trials. The data from the grid doesn't enter the systems used for clinical healthcare. In a clinical setting this is possibly the case, with the grid being used to exchange data between different systems from different organizations. Integrity (and its verification) then becomes vastly more important.

Privacy is closely related to the type of files being transferred to the grid. An interesting example is DICOM. DICOM files contain both the image data and the meta-data, with the latter containing personally-identifiable information. Both for research and clinical data it is required that the meta-data remains confidential.

There are a number of options available for anonymization/de-identification in such a case:

- Strip DICOM-headers entirely, and be aware that you can't retrieve the link back to the patients.

- Strip DICOM-headers, and maintain a copy of the meta-data internally with the ability to link a DICOM-image to the meta-data (digital or on paper)
- Hash the DICOM-headers on the grid, allowing anyone with a plain-text copy of the meta-data (stored internally) to verify that the image data on the grid belongs to the person specified in the meta-data
- Leave out selected DICOM-header fields (i.e.. only allow a patient ID, which allows a researcher with access to the local Hospital Information System to link the DICOM-file back to the patient).

In practice the type of anonymization depends on many factors. Within VL-e there are no specific services related to privacy, except for the basic access controls available using Virtual Organizations. Instead, it is up to the developers of the grid-enabled application to make sure that no personally-identifiable information ends up in the wrong hands.

4.4 Security in VL-e

Privacy and security are related concepts, however they are not the same. Privacy means that no other user should have access to sensitive personally-identifiable information, while security is a much wider concept that combines privacy, integrity (is the data I retrieve from the grid the same data that I stored yesterday?), access control (who can read, modify or delete my data?) and encryption (is the data safe in transport from and to the grid, and is the data stored safely at a potentially vulnerable remote location?).

Security isn't confined to these topics, however we consider physical or operating-system security of the servers that make up the grid to be out of scope. It is always possible that these are vulnerable to malicious attackers due to incorrect configurations or flawed implementations, however this is a topic that isn't confined to grids or VL-e.

Integrity, Certificates and Hydra

At this moment there is no way of storing file hashes or other integrity/validity information within the VL-e infrastructure. There is however a new grid service being evaluated: Hydra.

Hydra offers a means of distributing certificates and offers increased security via replication: a pool of hydra-servers can be set up, where the certificates are stored. If a majority of servers can verify a certificate then that certificate is valid, even if a minority of the hydra pool is compromised.

Hydra itself only offers a means of distributing certificates, and in itself doesn't provide integrity. However if Hydra and the LCG File Catalog are integrated it would be possible to automatically obtain a unique file hash for an uploaded file and verify the integrity of the file upon retrieval.

Fine-grained Access Control

Currently everyone within a VO has the same rights (to modify files, or to delete them). It thus isn't yet possible to define fine-grained access control within a VO.

What is being worked on are attribute-certificates. These are attached to a user and provides means to distinguish between roles within a single VO. Support for this has to be done at the SRM-level, but at least DPM and dCache support attributes. If deployed, this would allow fine-grained access control, but this does depend on the functionalities that can be defined at an attribute-level.

Encryption

As with integrity discussed above, there is no support within the VL-e infrastructure for automatic encryption and decryption of files. Any file stored currently is done so in plain view on the storage element. In case such a storage element is compromised, all data will be visible to an attacker.

Hydra offers a possible solution by distributing certificates as a means for encryption and decryption of files. If encryption is only required on a small number of hosts then it is also feasible to simply store the encryption keys locally and distribute them off the grid.

On the type of encryption used, asymmetric public key infrastructure-encryption is very costly in terms of computation, and this computation is required at the grid-client. Encrypting gigabytes of data using such a scheme would simply take too long to do on-the-fly.

Symmetric key encryption is more feasible, as a smaller encryption key can be used, make encryption/decryption a lot faster and still offer comparable security. An AES-CBC cipher algorithm is likely to be a good option. The security of such an implementation does depend on the availability of the symmetric keys, thus if a single Hydra-node or if a single key-owning host were to be compromised the attacker would be able to decrypt any encrypted data he retrieves.

4.5 Infrastructure Strengths and Weaknesses

Strengths

The whole EGEE-infrastructure was built to support very large amounts of research-data (petabytes), and to transfer these using high-speed network connections to many locations around the world. Replication and remote file transfers are well supported, thus providing advanced features to users of the grid.

The programming interfaces of EGEE are extensive and offer a high degree of control compared to other interfaces (which often only offer an upload/download mechanism). Although the documentation of these interfaces isn't ideal, a competent programmer can quickly learn how to use the grid resources via the EGEE interfaces.

Weaknesses

For actual data management and data storage, no native integration is available via (for example) FTP, NFS or the file system. Such an interface would allow a much easier integration of grid data storage and management compared to the current command-line utilities and APIs.

Latency

Another problem is that, due to the distributed nature of the grid and the many services required, the grid isn't well-suited to small fast read- and write-operations. These type of operations will be very slow compared to the normal, native operations on a network file system. This also means that databases (which depend on small fast read- and write operations) will be impossible to economically host using the current grid middleware infrastructure. If this would be possible using a reduced number of services or a new interface using (for example) the OGSA-DAI standard has yet to be determined and is subject for further study. The topic of distributed databases is one with many existing products and with a large amount of research currently taking place.

Availability

During testing of the VL-e infrastructure it is alas a frequent occurrence that grid resources are unavailable (due to outages, maintenance and other issues). In a research proof-of-concept environment like VL-e this is not a very serious problem: there is a group at SARA that supports grid users and answers questions and they are informed enough to know what is going on. In a production environment however such outages would be reason enough to not use the resource due to its unreliability, as availability of the data is often critical. After implementation we will therefore perform availability tests over a number of weeks in order to properly evaluate the current reliability of the services involved.

Service Management

For true production-grade usage of the VL-e infrastructure work has to be done in order to provide redundant grid services and minimize outages during working-hours. Service level agreements have to be available in order to provide users of the grid certainty in the degree of support. Notification services (via the web or a mailing list) should also be tailored to users of the grid in order to provide a means of informing users about maintenance and outages.

4.6 Related Grid technology

Globus

As mentioned in the previous section on grids in general, Globus is the most commonly-used grid middleware toolkit and the Globus project was initiated by Grid-pioneers Ian Foster and Carl Kesselman. Backed by the Open Grid Forum, Globus quickly contains support for the latest grid standards. The weakness

of Globus is that in recent years it has changed a lot, breaking backwards-compatibility. The EGEE group decided to use parts of Globus in their gLite middleware, combined with CERN-developed services.

Condor

Condor is a project used for smaller-scale scavenger grids, harnessing unused resources from computers within an organization. Multiple Condor instances can be set up to cooperate, however Condor is mostly focused on computational resources.

5 Background: IBM's grid technology

IBM has a long history in high-performance computing and as a result has been a major contributor to the field of grid computing [4].

In this section we will take a look at the current involvement of IBM in grid computing and at an IBM product that has been identified as potentially applicable in the described case. Information about the IBM product has been gathered through a number of channels (technical manuals, product descriptions and personal correspondence with developers) but in order to maintain confidentiality these resources have not been listed.

In Appendix C two other investigated IBM products are reviewed, but as these have been found to be less applicable for the AMC case they are not listed here. However, they might be useful for other cases and can to a certain degree be combined with the VL-e infrastructure.

5.1 Current IBM Grid involvement

In recent years IBM has focused its efforts in grid computing into two areas.

The first is to support the Globus Toolkit in various ways and integrate it into IBM solutions. IBM is a key member of the Globus Consortium [27] together with a number of members of industry. Together with Globus authors Ian Foster and Steve Tuecke as board members, the consortium backs the Globus Alliance and supports the use of the Globus Toolkit in industry.

The second main effort of IBM is in standardization, mainly done via the Open Grid Forum [28]. In cooperation with the Globus Alliance the OGF develops open (web)-standards that vendors adhere to.

Together, IBM supports a Grid infrastructure that is both based on open standards and developed using open source middleware.

Although IBM backs the Globus Toolkit, no products exist that both make use of the toolkit and are useful in the context described in the introduction. Instead, we have identified a number of IBM Healthcare and Life Sciences products that provide an additional benefit in our situation in the following sections. In addition we discuss the possibilities of interoperability with the VL-e infrastructure. This is further discussed in the chapter Problem Analysis and Discussion.

5.2 IBM Grid Medical Archive Solution (GMAS)

In this section we take an in-depth look at the functionalities of the Grid Medical Archiving Solution, the architectural design and a number of published case studies. We also take a look at possible pitfalls with relationship to this solution, the placement of this solution in comparison to the VL-e environment and related IBM technology for further investigation.

The Grid Medical Archive Solution combines multiple data sources into one large-scale storage grid, providing virtualized data storage across an organization. GMAS combines multiple distributed heterogeneous storage resources into a single user-accessible network drive. It is scalable, robust, secure (HIPAA-compliant) and is configured to have zero points of failure. GMAS offers fast streaming access to large amounts of data (petabyte-scalability) of any kind, with load-balancing and smart caching of objects to speed up transfers even more, regardless of the distance between the data sources and their users.

Other features of GMAS are the built-in hardware decommission-functionality (allowing administrators to upgrade, replace or disconnect nodes without influencing the storage grid as a whole), the high-level real-time centralized management facilities and the flexibility of determining intelligent rules to control data distribution across the storage grid. GMAS-functionality is split into various nodes, allowing functionalities where they are required together with redundancy.

The Grid Medical Archive Solution is a combination of Bypass StorageGRID software, IBM Tivoli Storage Manager and IBM services.

Functional overview

GMAS storage is visible to the user as a normal network-accessible drive (for both Windows and UNIX systems). This allows existing PACS (Picture Archiving and Communication Systems) to easily be migrated to the storage grid. GMAS automatically generates meta-data, giving users and administrators more flexibility to manage their data.

Non-functional overview

Security and Authentication

All data stored and transmitted within GMAS is encrypted using 128-bit AES data encryption, but only when the encryption module is used. This provides protection against security breaches and hardware theft from unsecured environments. Data from and to the Gateway Node can be encrypted using Secure NFS and CIFS encryption. Together this provides complete encryption both within GMAS and outside, straight to the client accessing the data objects stored on GMAS.

In order to provide data integrity, digital signatures are used to sign every data object stored on the storage grid. These signatures are stored on the Control Node. Also, an audit trail is available for every action taken on the

storage grid. GMAS offers Global File System Protection, which when used allows data only to be written once and not deleted or modified in any way after it has been written. GMAS also allows File-Level Protection; files in a certain directory can then not be deleted or modified for a fixed period of time. GMAS also allows user-level access via NFS and CIFS.

GMAS however has to be able to access sites outside of the local infrastructure. Existing firewalls can complicate this and administrators have to explicitly allow GMAS to contact the other sites. This weakens existing security slightly. Trust has to exist between GMAS sites in order to effectively distribute data objects on the storage grid.

The Grid Access Manager gateway nodes support the access controls inherent in NFS and CIFS.

Scalability

GMAS has shown that it can scale to various sites with many different Storage and Archive Nodes, providing access to hundreds of terabytes of storage with hundreds of gigabytes of throughput each day.

High Availability

Due to the integration of Bycast StorageGRID software with IBM Tivoli Storage Manager, GMAS can be deployed with complete redundancy and automatic self-healing. Lacking a single point of failure, secondary software nodes in GMAS automatically take over if a primary node fails. Using policy rules, data objects are stored at multiple servers and multiple sites, providing complete disaster recovery.

Performance

Nodes in GMAS are deployed in pairs, providing improved performance due to users being able to access data objects via two (or more) gateways at once. Data objects are also cached at these gateways, improving performance for frequently used data objects. Data objects are streamed to the user, providing direct access to large data objects without having to completely retrieve them. Performance for the end-user always depends on the local infrastructure which connects the user to GMAS. In production environments, GMAS has shown that it can handle over 600GB of throughput a day. Other competing solutions failed to handle this large amount of data.

Maintainability

Through a web interface, GMAS administrators are instantly able to review the state of the storage grid and are able to pin-point problems from any location on the network. Real-time and historic reports are available via this interface, including information on storage, bandwidth and CPU-utilization. Administrators on-site can directly manage each individual server via the Server Manager interface. This allows administrators to view the status of each software service and perform operations like restarting or shutdown of the node without specialized product knowledge. GMAS also allows administrators to put storage

hardware into a decommissioned state, after which GMAS will automatically ensure all data objects on the hardware are moved to a different location. In production environments, GMAS is known to require less than 0.1 FTE, reducing maintenance costs while providing high availability.

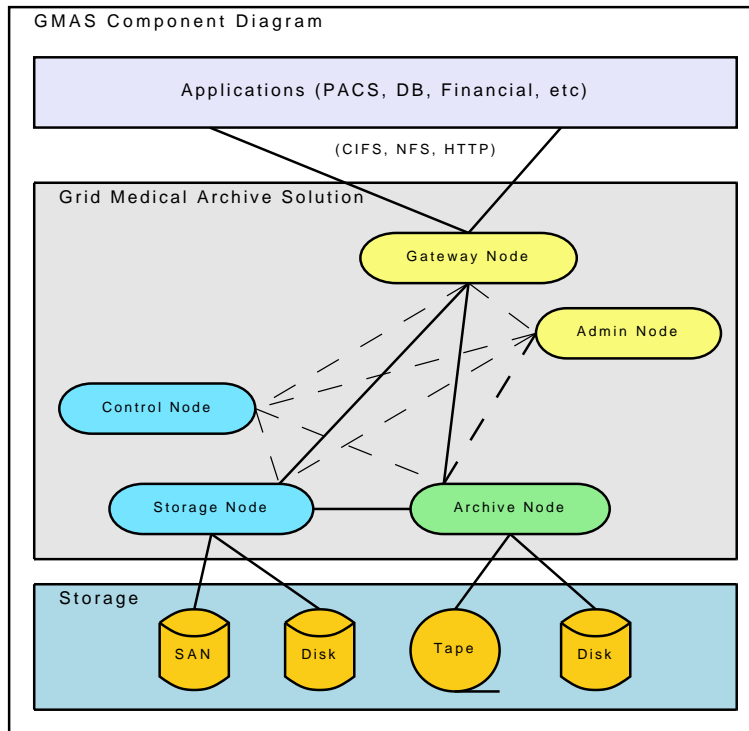
Ease-of-use, Ease-of-integration

The biggest advantage of GMAS is the ease of integration into an existing environment, with existing applications. Existing storage hardware can be used by GMAS, reducing initial deployment costs. Deployment of GMAS has been done in a matter of days. Deployment of GMAS is also possible for single-site installs. Users see GMAS as a normal remote filesystem, thus providing direct availability of data objects. End-users require little to no training in order to use GMAS.

Architecture

An overview of the GMAS nodes can be found in Figure 7.

Figure 7: The architecture of the IBM Grid Medical Archive Solution



Available GMAS Software Nodes:

- Gateway Node
- Storage Node
- Control Node
- Admin Node
- Archive Node

In the GMAS component diagram we see the various GMAS nodes in relationship to each other, and in relationship to both the application-layer and the storage-layer. Solid lines indicate that data objects are passed between these two nodes, dashed lines indicate that some form of control, monitoring or management is possible. Nodes with an equal color can be located on the same physical server.

Gateway Node

Each Gateway Node allows users to access the storage grid; all operations of the users on the grid go through the Gateway Node. Each Gateway Node supports the CIFS, HTTP and NFS protocols, providing a file system application interface for all types of client systems. Data objects are cached at the Gateway Node, which improves performance for end-users.

At each Gateway Node the administrator can specify the Information Lifecycle Management (ILM) policy, fine-tuning various properties of the data entering the grid. Depending on this policy, data is written to Storage and/or Archive Nodes.

At each location a Gateway Node is required. Multiple Gateway Nodes are possible, and this provides redundant access to the storage grid together with improved data throughput. For further redundancy optional High Availability Gateway Nodes (the High Available Gateway Cluster configuration) are available, providing continuous access to the storage grid.

Control Node

The Control Node controls all information about the data objects in the grid. It stores the meta-data assigned to each data object, together with the object-specific ILM policy. This policy contains information like the minimum number of copies a data object has across the storage grid, and how this number is split between Storage Nodes and Archive Nodes. An ILM policy can also for instance determine at what age a data object should move from a Storage Node to an Archive Node.

Additionally each data object has a digital signature assigned to it, safeguarding the integrity of the data object. The Control Node also determines the status of each Storage and Archive Node, and redirects data objects if a Node goes into a stand-by or an off-line state. Data object meta-data (including the digital signature) is stored on the Control Node.

A minimum of two Control Nodes are required for a storage grid, providing redundancy in case either goes down. Optionally a Control Node and a Storage Node can be situated on the same server.

Storage Node

Each Storage Node provides data objects, which are accessed by users through the Gateway Nodes. Based on the ILM policy, the Storage Node can migrate data to other nodes if required. At least one Storage Node is required in the storage grid. It is not possible to update the amount of storage on a Storage Node; in order to do this the Storage Node would have to be rebuilt in order to increase the amount of storage. Optionally, objects can be encrypted in the Storage Node using the Encryption Module.

Archive Node

The Archive Nodes of the storage grid allow for less active data objects to be placed into long-term storage. An Archive Node interfaces with the IBM Tivoli

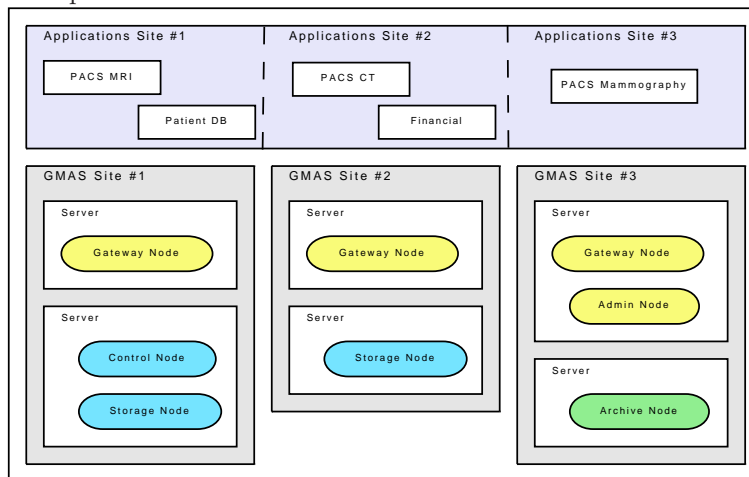
Storage Manager and IBM System Storage Archive Manager, and depending on the ILM-policy data objects are placed temporarily or permanently on these systems. The Archive Node interfaces with the IBM Tivoli Storage Manager to enable Hierarchical Storage Management functions. Archive Nodes are an optional part of the storage grid.

Admin Node

The Admin Node provides a Web-based interface for system administration. Via this interface, all grid node activities can be monitored and managed. All nodes and their status are visible through the Admin Node, allowing administrators to determine which nodes are online, and what the cause is of nodes no longer being available. Further more, reports can be generated showing how the grid has performed over a period of time.

At least one Admin Node is required, and it can be installed on the same server as a Gateway Node. Each Admin Node can optionally contain an Audit Module, which stores information on all transactions on the storage grid. It can be read and analyzed by third-party applications.

Figure 8: Example deployment of the IBM Grid Medical Archive Solution across multiple sites



In Figure 8 we show a possible deployment diagram. Users from each of the 3 sites can access any of the 5 applications, regardless of their location. GMAS nodes are split over the three sites, interconnected by a wide area network, with certain nodes being on the same physical server.

GMAS Case: Health Alliance

GMAS was installed at a number of locations of the Health Alliance Network, located in North-Central Massachusetts and Southern New Hampshire. They chose GMAS because of the open standards and flexible infrastructure, which allowed for existing third-party PACs to be migrated to the grid storage. The Health Alliance had to cope with large amounts of data (30 TB) from a variety of systems (CT, MRI, PET, digital mammography and the hospital information system Soarian). Their existing DVD-jukebox method of long-term storage was slow, limited and required manual retrieval of data.

GMAS allowed Health Alliance to reduce management effort and allow immediate access by surgeons, radiologists and other health-care professionals. Health Alliance further also used GMAS for storage of financial data. GMAS was delivered and the project was up and running on time and within budget, delivering 100% availability in nearly a year of operation.

GMAS Case: University Health Care System

UHCS is a non-profit network of hospitals throughout Georgia and parts of South Carolina. UHCS turned to GMAS in order to improve scalability and resiliency of the existing IT infrastructure, which supports a number of PACS systems. The cardiology PACS alone produces 15 TB a year, and this number continues to grow. GMAS allows storage virtualization throughout the hospital network and this allows UHCS to cost-effectively scale the required storage.

UHCS has also used GMAS for storage of database snapshots, increasing resiliency of business systems next to the clinical systems. Previous archiving of the cardiology department required 3 hours of downtime overnight, every night, for transferring of the clinical images. GMAS solved this problem, reducing the transfer time to 45 minutes without downtime and without manual intervention.

Possible issues

GMAS might sound like the next big thing; however it is necessary to keep the context in mind. Accessing storage via a VPN isn't new and such a setup allows administrators for each site determine who gets access to what. Network-wide storage doesn't depend on a Grid-like infrastructure; depending on the site and amount of data handled a central server with a secondary backup server might be more than enough. Not every application generates terabytes of data, and depending on the situation deployment of GMAS might be over-kill.

Security and privacy are very important issues in healthcare, and via GMAS many more users get access to confidential information. Before deployment, it is wise to carefully determine who will get access to information and to make arrangements between GMAS sites to verify security policies in place. You can't be careful enough when lives depend on the correct data being available at the correct location.

Users might get the idea that GMAS will help in compatibility between different PACS- and DICOM standards. This is not the case. GMAS provides access to the files from various locations, however it doesn't convert or manipulate these files. Other IBM solutions are available for this problem and these will be discussed later on.

Relation to VL-e

It is clear that there is quite a bit of overlap when comparing GMAS and the current VL-e Proof-of-Concept infrastructure. GMAS isn't limited to medical-applications, as it allows storage of any type of data. The term "Medical Archiving Solution" is thus more limited compared to what GMAS can actually deliver.

Both do have their strengths and weaknesses. IBM has a number of high-profile test cases using GMAS, and from all accounts GMAS has performed remarkably. What hasn't been tested is distributed storage over a large number of locations, as all test cases are about a relatively small number of sites. Also, the nature of GMAS limits its use to a single distributed organizational unit with central control. On the other hand, GMAS provides a natural interface for existing applications (a network file system), which allows a migration to GMAS be accomplished within a matter of days.

The VL-e environment doesn't have many of the features GMAS currently has. The main difference is the lack of central control in a VL-e environment, which possibly leads to a less stable environment. The main benefit of deploying GMAS is also the transparent interface to existing users.

Related IBM Technology

- GMAS is the extension of the IBM Medical Archive Solution.
- GMAS can be deployed on the General Parallel File System (GPFS), which is a high-performance shared-disk file system. GPFS provides fast reliable data access from multiple nodes in a cluster of Linux or AIX

servers. Combining GMAS and GPFS allows adding scalability to the gateway nodes.

In conclusion, the Grid Medical Archive Solution excels in combining resources from multiple locations and providing an interface for existing PACS systems. The management features allow a high degree of control and flexibility with relation to data object policies.

6 Problem Analysis and the implemented solution

In this section we analyze the problem described in Chapter 2 using the background from the previous chapters on grid technology and discuss a number of possible solutions. We then discuss the implemented solution in detail.

What has to be made clear is that the AMC case is, in principle, a simple question of data storage and data management. The VL-e grid infrastructure is capable of much more, however these aspects are the only ones required for this particular case. There are multiple solutions possible, each with advantages and disadvantages.

6.1 Alternative solution: local storage

IBM and its competitors have a lot of experience with setting up dedicated network-accessible storage. With the cost of storage dropping rapidly and considering the complications of using grid-enabled remote storage, keeping the storage on-site can be both cost-effective and much easier to implement. There are however a number of benefits that grid-enabled data storage provides compared to using local storage:

- Automatic replication. With each sequencer run costing thousands of euros, losing the generated data can become very costly. With the inherent property of automatic replication over multiple sites, the chance of data-loss is decreased dramatically.
- Decreased maintenance costs. By using grid storage, the AMC is able to tap into the resources at large government-sponsored grid storage facilities. Due to economies of scale, far less employees are required to run the same amount of systems required for the storage of the sequencer data.
- Remote accessibility. Via strict authentication protocols, it is possible to obtain the sequencer data from different sites, where grid storage to be used. Agreements with other facilities can be made in providing the sequenced data via the grid, while high-level access control is available to only allow access to the data from specific institutions using a virtual organization.
- Automatic scaling. For the large grid data facilities, it doesn't matter in principle how much data is being stored. For the option of local storage, regularly an assessment will have to be made if the currently-available storage is sufficient.

6.2 Alternative solution: proprietary off-site storage

Using off-site storage is the next option, but instead of doing this via the grid infrastructure it is also possible to contract a single data storage provider. Hosted

data storage solutions are available from many providers, however this comes with other drawbacks:

- Reliancy on a single corporate entity. With a typical off-site storage solution all the eggs are in one basket. This doesn't have to be an issue, as long as things go well. If the company decides to unilaterally increase rates, the customer doesn't have a lot of choice (short of removing all the data and moving to the next supplier, in spite of all the costs involved). With grid-enabled storage the data storage of multiple sites are being used. If one supplier raises its rates, the customer can relatively easy switch to another site.
- Open standards and open source software. The VL-e infrastructure makes use of European-sponsored open source software, which allows users to avoid vendor lock-in. In the case of proprietary off-site storage switching to a different provider can be coupled with high costs and a troublesome migration due to the previous dependency on proprietary software and standards.
- Expansion to computing resources. The VL-e infrastructure provides more than just data storage. If for research or clinical purposes a large amount of data has to be processed, then this can also be done using the grid facilities. Such flexibility is currently not available in most commercial offerings

One exception to the last item is that of Amazon S3, in combination with Amazon EC2 services. Amazon [1] is currently hard at work at developing online webservice-accessible resources, however the first two drawbacks apply also to Amazon.

6.3 Implemented solution: off-site EGEE/VL-e grid storage

Because of the issues mentioned above we have decided to implement a grid-storage solution. The next question that then follows is: "which grid-storage solution?". There are many different possibilities, however within the VL-e project there were two choices: use of the (older) SRB storage service or the (newer) SRM-based storage service described in Chapter 3. As the SRB storage service is being phased out the only future-proof solution, for the time being, is the use of the EGEE-developed SRM grid interface.

The problem that quickly surfaced however is that VL-e offers a collection of services without one single user-interface. The interfaces available that are used for accessing the SRM-based storage are a collection of older and newer command-line tools and APIs, available solely for Linux. After some investigating it was clear that the EGEE-developed SRM grid interface doesn't use web service protocols or even has a clear selection of non-web protocols; much of the

unified EGEE grid logic, for accessing the LFC for instance, is hidden into the **client-side** grid libraries.

This issue greatly reduces the number of options available in providing non-technical users a clear interface with which to work. Instead of being able to develop a platform-independent Java API, for example, users of the SRM grid interface within VL-e are bound to the low-level C APIs and their command-line counterparts available only on Linux. This might be acceptable in an environment with highly technical Linux powerusers and developers, however this is not acceptable for ordinary non-technical users: they would always require assistance if they needed to perform an action on a grid storage resource.

In overcoming this problem the solution was relatively straight-forward: develop a network-accessible interface using the low-level C grid APIs and have users access the Linux machine with a grid interface using standard network protocols. This manner of accessing the grid has both advantages and disadvantages, as will be explained below. After some time we dubbed our server the Grid Access Point (GAP), after a similar solution for accessing the older SRB storage.

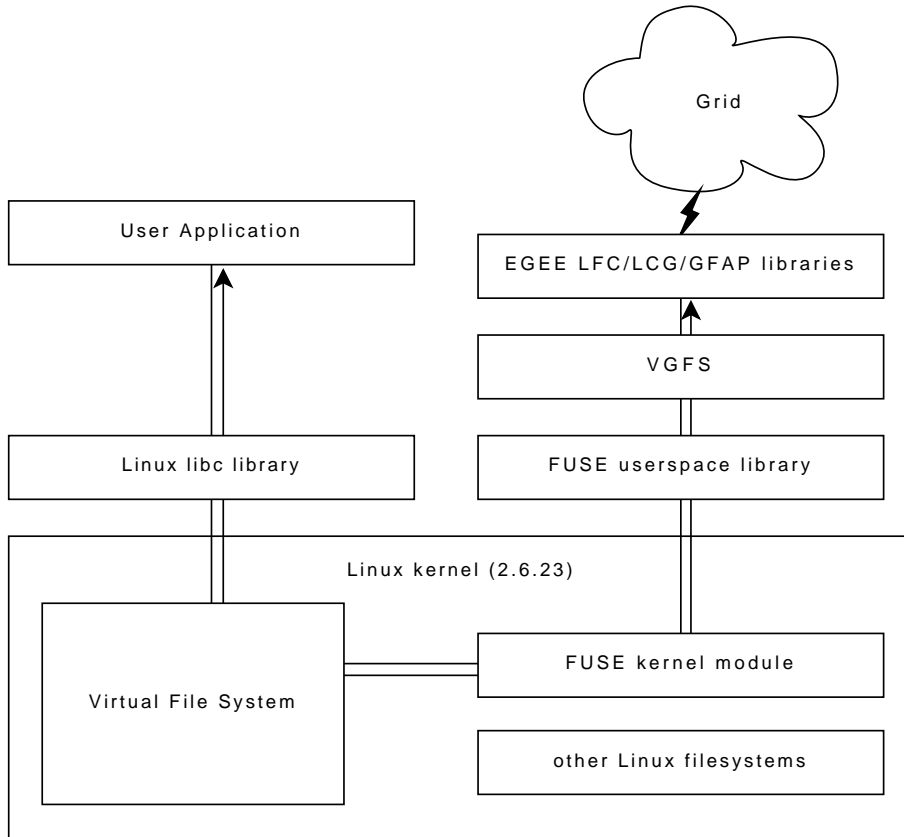
There were a number of choices possible in developing a network-accessible interface. Developing an FTP or HTTP server for instance were possibilities, however at the time we started development the applicable network protocols were not yet known. Thus instead we made use of the Filesystem in Userspace (FUSE) Linux kernel module [8] in order to develop a server-independent interface of accessing the remote grid storage, which we call VGFS. See Figure 9 for an overview of how an application accesses the grid via Linux, FUSE, VGFS and the EGEE libraries.

FUSE allows a Linux user to 'mount' a remote filesystem as if it were part of the local machine, making existing Linux network services available to access these remote files as if they were part of the local network. To be more precise, FUSE allows the development of such a filesystem-interface to be done much more efficiently (from userspace instead of directly in the Linux kernel). This allows us to link the normal POSIX system calls with the C APIs used to access the grid storage. In effect the filesystem grid-interface, together with a NFS or CIFS/SMB server, allows ordinary Windows, Linux and OS X users to map the grid filesystem from the GAP, thus letting them access the grid storage as if it were a normal local network accessible storage device. Screenshots showing the grid accessed from a Windows PC in this way are shown in Appendix B together with further in-depth details about VGFS and the current status of development. Figure 10 shows how VGFS allows users to access files on a grid as if they were part of a normal Linux filesystem.

A further point of concern was the lack of file encryption in the existing command-line EGEE interface. Especially in the medical and life sciences community sending any data outside the local network must be done with caution. Privacy regulations are strict, but even with the consent of patients having files stored at a remote location without any encryption or control is not acceptable.

To overcome this an option was implemented in VGFS for it to enable complete transparent CBC AES 256-bit file encryption and decryption. The key

Figure 9: Technical overview of the GAP/VGFS components



for encryption is stored in a file on the GAP and multiple keys can be used in order to provide access to other sites. Files sent via the GAP are automatically stored in an encrypted state on the remote storage element, files retrieved from the grid are automatically decrypted using the same key.

In summary, the EGEE GAP/VGFS characteristics:

- Integrates grid data storage into existing heterogeneous networks and products
- Mountable and exportable via a single Linux-based Grid Access Point
- Transparent usage via a local network on all modern operating systems (NFS, SMB)
- Supports all EGEE-style (LFC/LCG/SRM) high-capacity grids
- Supports transparent AES file encryption

Figure 10: Example Linux filesystem with VGFS enabled

```
\
  \bin
  \etc
  ...
  \mnt
    \grid <- VGFS-mountpoint
      \astron
      \atlas
      ...
    \vlemed <- VO directory containing grid files
      \alex <- Files of a grid-user
      \amc-ng <- Files of neurogenetics department
      \amc-ng-enc <- Encrypted files
      \silvia
      \tristan
```

- Supports both high-speed GridFTP transfers and real-time RFIO operations

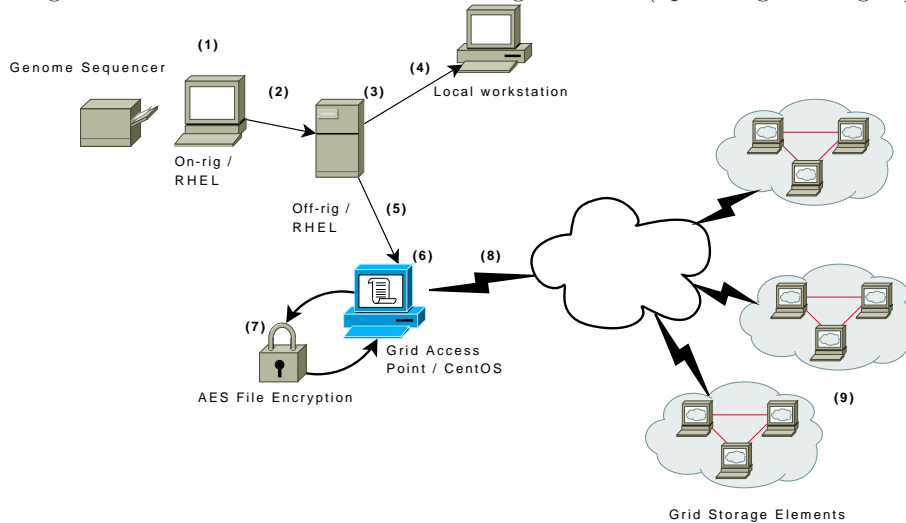
6.4 Deployment of the GAP/VGFS solution at the AMC

During the last month of the internship deployment of the VGFS-based GAP solution was conducted on-site at the AMC. Particular issues like the AMC firewall and questions like security and encryption were addressed and the solution was tuned to support the processes of the neurogenetics department.

A schematic of the deployed solution can be found in Figure 11.

- 1) A researcher finishes the DNA sequencing. He runs a local script that asks which data files need to be preserved, whether encryption is to be used and which other research centers need to have access to the files.
- 2) The script transfers the files via NFS to the off-rig server.
- 3) Once the transfer is complete, the off-rig server places the files at a location where local workstations can access the files. The same files are transferred to the GAP server.
- 4) The local Windows/Unix workstations have access to a Samba drive with only the last couple of sequences. The researcher can directly process the available raw data.
- 5) The data is transferred from the off-rig server to the GAP via one of a number of NFS-exported directories.

Figure 11: AMC GAP-based off-site storage solution (uploading to the grid)

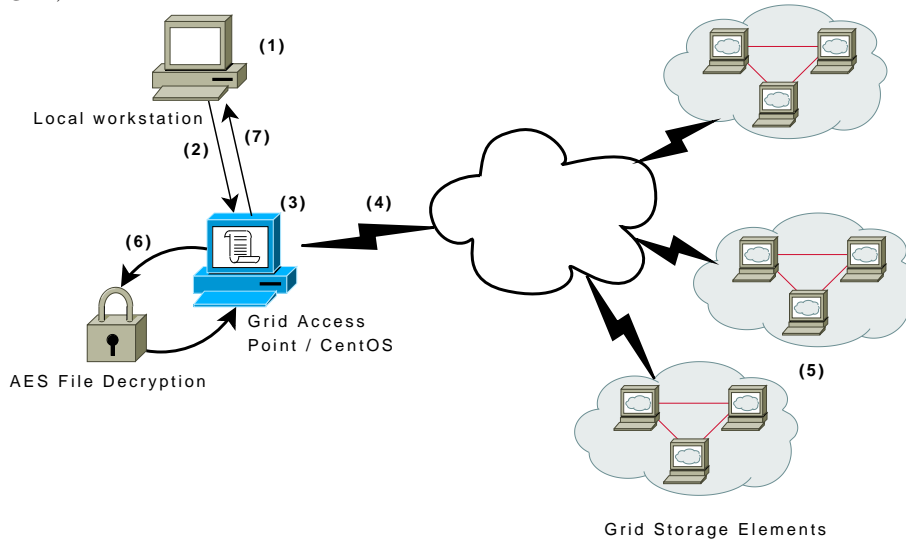


- 6) The GAP receives the data in a directory that has VGFS mounted. VGFS runs continuously and authenticates itself to the grid when necessary.
- 7) The GAP optionally encrypts incoming files automatically, using 256-bit AES symmetric key encryption. In our setup the choice of the NFS-exported directory determines if encryption is used, and if so which key is used in encrypting the files. Multiple encryption keys are used, one for encrypting private AMC data, one for encrypting data that is also used at other research centers, etcetera.
- 8) The GAP, via the VGFS grid interface, adds the required LFC-file entries and sends the files via the SRM interface using GridFTP.
- 9) The files are stored on the grid storage elements.

Retrieval of the files from the grid is somewhat simpler (Figure 12), as the workstations can access the GAP directly.

- 1) A researcher requires raw image data from previous genome sequences and requests retrieval of these files from the Windows/Unix workstation via a shared drive.
- 2) The request is sent via CIFS/SMB to the GAP.
- 3) The GAP receives the request in a VGFS-mounted directory.
- 4) The GAP requests the file using the LFC and obtains the files via the SRM interface using GridFTP.

Figure 12: AMC GAP-based off-site storage solution (downloading from the grid)



- 5) The grid storage element with the file opens the GridFTP-connections to the GAP and transfers the file.
- 6) The GAP optionally, depending on which VGFS-mounted directory is used, decrypts the file using the appropriate 256-bit AES key.
- 7) The (decrypted) file is sent back to the researcher.

6.5 Issues encountered

The main problems encountered are related to the strict requirements on the firewalls at both the AMC and the SARA. At the SARA (which hosts most of the grid services for the VL-e project) RFIO access from outside was initially not accepted. During development the services were at times not available, further limiting the progress made. These problems were dealt with thanks primarily to the dedicated grid support staff at SARA. The AMC is first and foremost a medical center. Being a medical university as well a lot of research goes on, which leads to differences in focus by, for example, the IT support staff. The fact that a similar-sounding VL-e GAP had been set up at the radiology department did make the IT support staff a lot less anxious, and thanks to them the deployment of our solution went relatively smoothly.

During development and testing a number of smaller problems surfaced. The VL-e CentOS image was meant to be for demonstration and research purposes, but for production purposes it is rather out-dated. Some of the included libraries lacked proper multi-threading support, for instance. This isn't a problem for

simple linear actions to and from the grid, however a filesystem-interface has to be able to support multiple threads reading and writing. This is especially the case with network filesystems. We attempted to solve the problem by replacing the effected libraries with modified versions, however other multithread-issues appeared. A temporary solution we enabled for deployment was switching the FUSE layer to a single-threaded mode, which gave improved stability. This however isn't a viable option in environments where multiple users are accessing the grid simultaneously; they would have to wait in turn in order to access the grid.

For proper production use either a new VL-e CentOS image would be required or an updated single-purpose VL-e GAP image would be necessary, with VGFS pre-installed and configured.

Getting the GAP to export the VGFS-mounted filesystems wasn't as easy as initially expected. Graphical file browsing utilities, like Windows Explorer and Gnome's Nautilus, tend to make many calls to the filesystem. Explorer even attempts to automatically download files for certain types of files. For normal (network) filesystems this isn't a problem, however when dealing with gigabyte-sized files and a very high latency this behavior slows down browsing considerably.

A related problem was found when dealing with reading via NFS. Due to the typical filesystem semantics used in NFS servers at the setup at the AMC files were repeatedly requested for retrieval, so making reading large files via NFS impossible. As files only had to be written via NFS, support for NFS filesystems via FUSE is relatively recent and read/write support worked using Samba/SMB we decided to not persue this issue further.

A final problem that was discovered much too late was the use of 32bit signed integers (instead of 64bit integers) for determining the filesize within VGFS. This caused problems for files with a filesize of more than 2GB (2^{31} bytes). Once discovered it was easily dealt with.

7 Assessment and Testing

In this section we look at various aspects of the deployed solution described in the previous section. For each of the non-functional criteria mentioned in Chapter 2 we investigate if the solution is good enough and if not what would be required in order to improve the solution. We also test the measurable criteria and determine if the results are acceptable in our situation.

7.1 Grid certificates and Security

The VL-e grid infrastructure is focused on providing secure grid resources. In order to gain access a researcher has to obtain access via the Dutchgrid certificate authority and meet with an authorized registration authority in-person with identification papers. Access to the VL-e data resources is then available using a pass phrase and the grid certificate obtained from the certificate authority. Access is limited to the resources that have been allocated to the Virtual Organization of which you are a member. All protocols used in accessing the grid resources use the supplied user certificate and are done over a secure channel.

In a normal situation, each user of the grid resource has his/her own certificate. In our case however this requirement has been relaxed: the IT administrator has a single certificate installed on the GAP. He controls which users have access to the grid resources, effectively in his name. Use of a user certificate in this manner is against the grid policy and thus would have to be changed in a production environment.

There have been discussions with key members of the VL-e project concerning so-called 'Robot certificates'. In our case the user certificate allows much more than is actually used; it also allows access to computation resources, for instance. A user certificate is also transferable to another host, something that in our case shouldn't be allowed. An ideal certificate would be tied to a single host and allow long-lasting access to a limited set of data resources. At this point robot certificates aren't available, thus a user certificate is being used as a stop-gap.

7.2 Privacy, Access Control and Encryption

Although a lot of work has been done in setting up a secure grid, there is a lack of attention within VL-e on the question of privacy. By default, users within a single Virtual Organization are allowed to view each others data, although the implemented basic Unix access control allows users to restrict this. What is worse is that, in spite of the recent developments on using Hydra (see the previous section on security and privacy in VL-e), there is no single means for encrypting privacy-sensitive data.

At this time the best way to protect privacy is to simply not use the grid for any privacy-sensitive data. In our case at the AMC we put raw image data of the genome sequencer on the grid, which effectively is the DNA sequence of humans. The data used is marked as research-data, as the patients have given

their consent, but even if consent has been given it is the obligation of IT to be careful with the supplied data.

Efforts are underway in VL-e in order to use more fine-grained access control when storing data on the grid, however at this moment only VO-based read/write access is available. In a production setting a separate VO would have to be used for all sites where the raw image data has to be available. At this moment however the VL-e Medical VO is being used, which gives more users access to the files than strictly required.

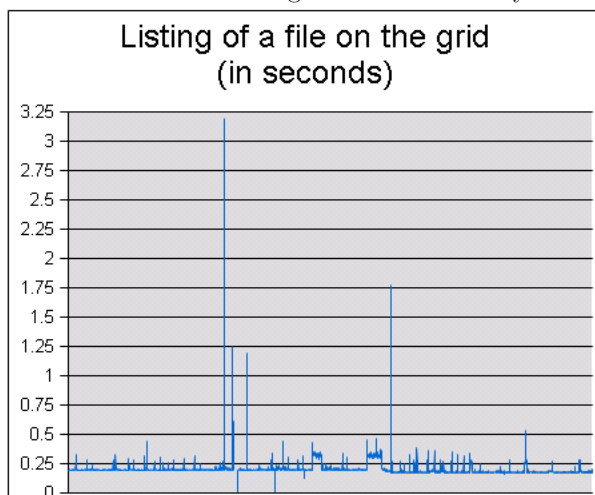
The second means of gaining access is via IT administrators where the data is physically being hosted. If privacy were able to be ensured from these IT administrators the problem would be solved, however experience shows that a lack of security can also lead to a lack of privacy at these grid sites. Proper encryption of privacy-sensitive data is the only way forward on this level and an initial solution has been included within VGFS. A solution for company-wide secure, encrypted file storage (both on and off the grid) is the IBM GMAS product.

7.3 Availability

In order to be production-ready, the user has to be ensured of a high degree of availability. Naturally, in certain situations it is required to take resources off-line, however these would have to be kept to a bare minimum and communicated properly with the IT administrators.

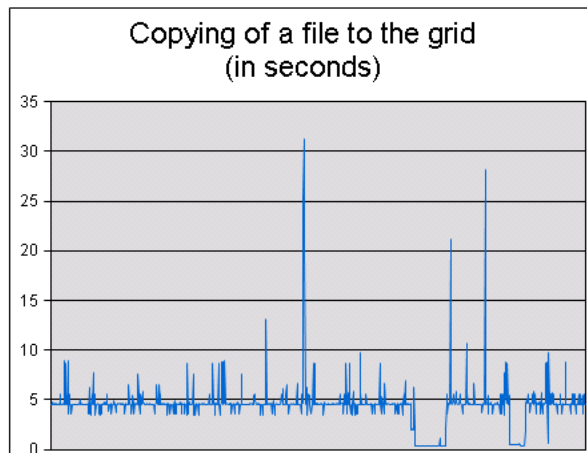
Although statistics and personal descriptions are available from other sources on the availability of VL-e grid resources, for this study we did our own availability tests in the period of February - March 2008. Stakeholders of the VL-e grid services were notified beforehand. The following statistics are the result:

Figure 13: Availability test results



The results from Figure 13 are from 2780 points in time, conducted over a period of three weeks. Every few minutes the LCG File Catalog was queried for a file. Only twice did it fail, giving it a 99.93% uptime over this period. The average time required to query the LFC was 0.2043 seconds. At 5 points in time the query time was higher than half a second.

Figure 14: Availability test results



The results from Figure 14 are from 1383 points in time over the same period. Instead of querying the file catalog, the SRM was contacted and the request was sent to store a file on the grid. The average time to store this small (100KB) file was 4.846 seconds, with a much larger variance compared to the file catalog test. The three dips in the graph are from two intervals where the SRM interface was unavailable or unable to store the file. Together, these dips were 91 time points, leading to an uptime of 93.34%.

The difference in uptime is remarkable, but understandable. In the first test only the LCG File Catalog is contacted, while in the second test the file catalog the SRM/dCache server and the storage element all have to work together in order to deliver the correct file. If only one grid service is down, due to a single point of failure, grid storage grinds to a halt. Even so, an uptime of 93% is unreliable (one in twelve copy-requests fail). This might be acceptable for a research environment, but for a production environment this clearly wouldn't be the case.

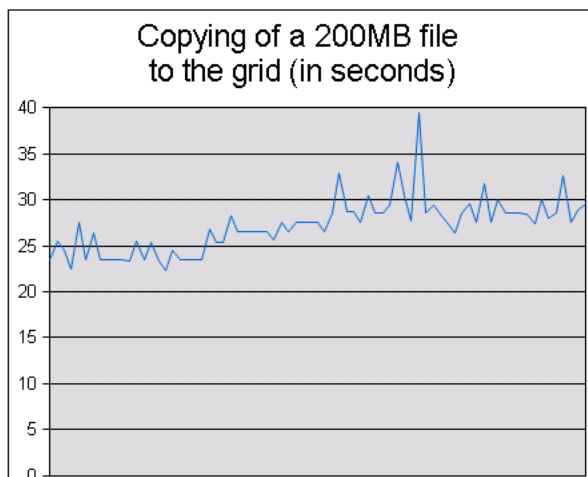
These results also show the issue of latency resulting due to a request having to contact a number of different services. A latency of 0.2 seconds is acceptable for listing a file, and this is due largely to the fact that only the LCG File Catalog is contacted, however when the overhead of storing a file is 4-8 seconds this does limit the usefulness of grid storage to only large files.

7.4 Performance

The grid infrastructure has been developed from a performance-perspective; if CERN can handle the petabytes generated by the Large Hadron Collider then it can handle the amount of data generated by the AMC genome sequencer. In our case performance is not nearly as critical; as long as the 30GB/day are stored at the end of the day on the grid all is well. Retrieval of the data for local analysis does mean that the user has to wait until the data has been transferred back, however this would also be the case in any other situation.

What has been found from experience (and confirmed in the previous section) is that latency (and not through-put) is the largest problem for users storing and accessing the files on the grid in real-time. There is a considerable amount of overhead in order to transfer a single file. During our availability tests we also performed a number of performance tests, which were timed as to not interfere with each other. The following statistics are the result:

Figure 15: Performance test results



In Figure 15 the results are shown of 75 200MB files successfully stored using the grid interface. These tests were conducted over a period of 48 hours during normal conditions. The average time required to store a 200MB file within this period was 27.23 seconds, which can be translated to 7.4MB/s including the latency required to store a file on the grid. The variance of the transfer times is low compared to the results from the latency and availability tests, as temporary network issues are smoothed out over a longer period of time. Also, these results are from fewer trials and over a shorter period of time, thus biased results are more likely.

Extrapolating these results to the 14GB file that has to be stored after every run, each transfer would take about half an hour to be transmitted.

In Figure 16 the results are shown of 75 retrievals of the same 200MB files

Ensuring integrity doesn't have to be difficult, the LFC could easily be modified to support it. When uploading a file, a checksum could be stored together with the location of the file in the LFC. This could then be checked upon retrieval of a file, automatically informing an administrator in the case a corrupted file has been found. This hypothetical solution would only provide integrity as long as the LFC hasn't been compromised.

A second solution would be to use Hydra for storing data checksums. As Hydra offers multiple redundant servers, it offers a more resilient way to store the data checksums. An even better modification would be to combine the LFC and Hydra services, thus also ensuring that the location pointed to by the LFC is the proper file.

A third solution would be to use the IBM GMAS product. With GMAS, the checksum data could be stored on-site and the actual data off-site. The problem with this approach however is that the checksum data would not be available to other authorized VO members that have access to a particular set of data.

7.6 Integration with IBM products

Of the 3 IBM products reviewed, the IBM Grid Medical Archive Solution offers the most in terms of being able to enhance the services provided by the VL-e infrastructure. GMAS is especially useful in providing security and integrity in ways that the VL-e infrastructure has not been designed to facilitate. GMAS also provides a number of features that overlap with the features of the VL-e infrastructure, however it does not provide cross-administration grid storage. Together, IBM GMAS and the VL-e infrastructure complement each other. Due to time restrictions however, we have not been able to test IBM GMAS on-site at the AMC.

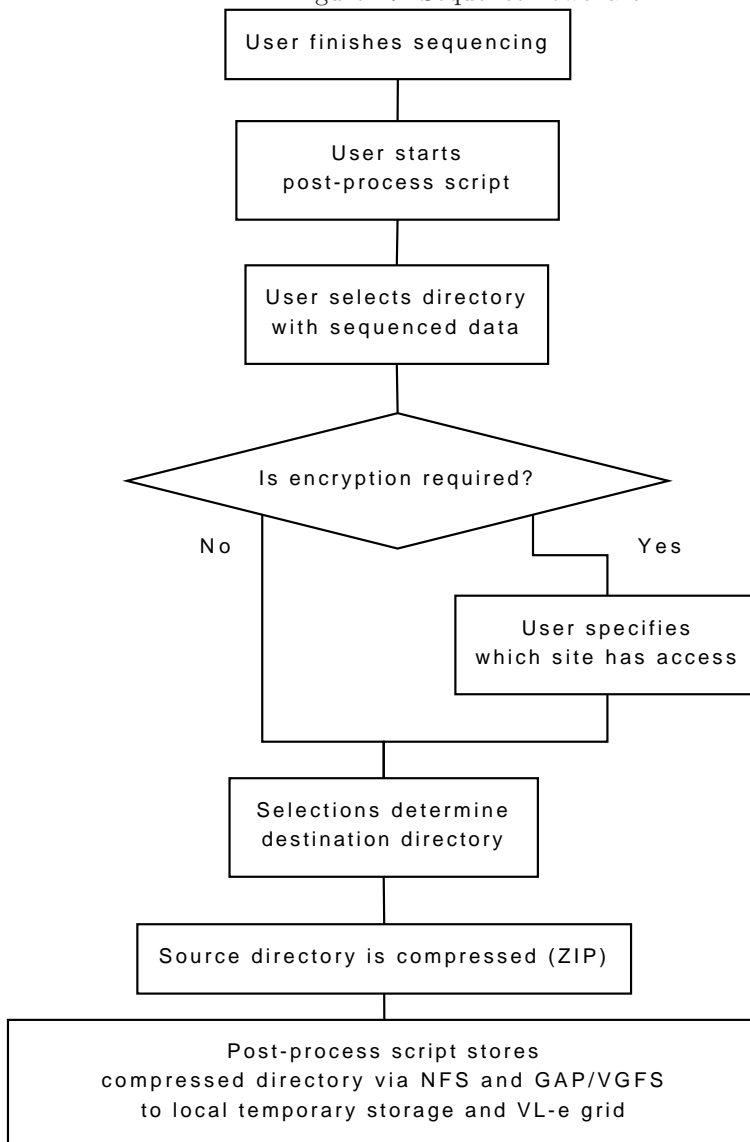
7.7 Integration into AMC sequencing process

The final step is also one of the most important ones: integrating the complete solution with the current practice at the AMC. In an ideal world the user wouldn't have to do anything differently; in practice there are a number of choices that the user can make that determine where the sequence data has to be stored. Figure 17 shows the steps taken after a user has performed a sequence run. Screenshots of the script developed to support this process are shown in Appendix A.

The step of determining which site should have access to the data is the only step that requires strict manual input. We can't determine 'a priori' if the data should be restricted to only the AMC, if it should be shared between the AMC and other sites or if the data is allowed to be stored unencrypted on the grid.

For each different site-selection option a separate exported NFS directory from the GAP is available, each running its own VGFS-instance, possibly with encryption using a site-specific key.

Figure 17: Sequence flowchart



8 Conclusions

The main question we set out to answer was what the state of grid computing is and if the current grid infrastructure is suitable for production environments. It is clear that there has been a lot of progress with relation to grid computing in the last few years, and with a recent peak in interest in similar areas this will likely remain to continue. The current VL-e Proof-of-Concept infrastructure provides technical users with useful tools and thanks to the development that took place within this internship we hope to make a subset of these tools available to non-technical users via integration in normal IT infrastructures.

However, in our experience and during the test-phase of this project we have determined that there are other issues that have not been solved and can not easily be solved from a grid-user point of view. Availability, or the lack thereof, is simply the critical issue for all production systems. Although our test-phase only was done over a limited period of time, we believe it is an indication of the problems with both the grid infrastructure within VL-e and grids in general. Due to the high degree of complexity required, and the lack of attention in solving single points of failure, the whole service can not be deemed reliable enough for production environments (at least 99.9% uptime would be necessary). Currently availability of the VL-e grid infrastructure is done on a 'best effort' basis. We therefore can not recommend the use of the current VL-e grid infrastructure for production environments at this point in time.

Figure 18: VL-e evaluation

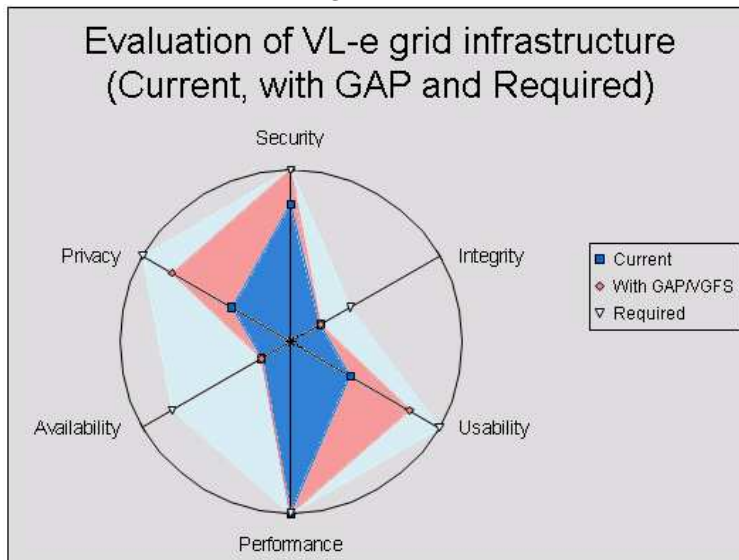


Figure 18 shows the status of the 6 evaluated non-functional criteria before and after our project, and what would be required before the VL-e grid

infrastructure can be used in production environments:

- **Security:** A focus-point of the VL-e project, security is strict and access to the grid is closely regulated. The inclusion of client-side data encryption improves this slightly.
- **Integrity:** No efforts have been done to verify the integrity of data within the VL-e project. This hasn't been an issue for us during the project, but for certain commercial and production usages verification of integrity would be required.
- **Usability:** Usability from the viewpoint of the non-technical grid end-user has not been a serious focus-point for the VL-e project as a whole. During our project we have made a serious attempt to improve this to a level where grid resources can be seamlessly integrated into an existing infrastructure, allowing ordinary non-technical users to access the grid transparently.
- **Performance:** A focus-point of the VL-e project, performance of the current infrastructure is optimal. As performance was not a serious criteria during this project we have not focused on this issue.
- **Availability:** Unfortunately availability of the grid infrastructure is proving to be its 'Achilles heel'. As reliability is the most crucial aspect for production environments the issues surrounding availability will hinder any serious usage of the grid for such purposes.
- **Privacy:** An issue that receives moderate attention within the VL-e project, but as higher standards of privacy are required for Healthcare and Life Sciences environments we have improved this via use of client-side data encryption.

In order to make this whole issue clear in a single sentence I have, in the light of Moore, Murphy and Amdahl, summarized this as follows:

Alex's observation:

Unreliable data storage is only cheap if your data is worthless

Do note that this is not specific to VL-e or grids. Amazon's S3 service was down for 3 hours on Friday the 15th of February 2008 [20]. Many small websites were unavailable, customers were outraged and the reputation of Amazon web services and Cloud Computing took a hit. Even with better than 99.9% uptime, unreliability depends both on the expectations of your users and the criticality of your service.

8.1 Recommendations to VL-e

- The current grid is certainly useful enough for research purposes. VL-e should focus on getting more users in this area, so fostering a more active community and gaining more experience and goodwill.

- The barrier to entry (certificates, software) has to be lower, but there is interest and there are possibilities.
- The main advantages of the current data grid infrastructure lies in performance, security and scalability.
- The secondary advantages are the combination of data and computational resources and the online/offline collaboration within virtual organizations.
- Selling the grid in situations where other solutions would be a better fit leads to disappointed users and a loss of credibility. Realize that it should always be about the best tool for the best job.
- For production-usage of the grid there are a number of non-technical issues still to be tackled: service level agreements, fees associated to grid usage and administrator-level notification and support.
- There are also a number of technical issues that remain to be addressed: Improved availability would be required of the infrastructure as a whole by removing single points of failure, host/robot certificates would need to be deployable for GAP servers and there are opportunities for improving integrity and usability.
- Make sure that the interfaces to the grid remain stable and communicate updates clearly.
- Determine if having two separate grid middleware software infrastructures (both gLite and Globus) is necessary, as from a technical perspective this is counter-productive. Investigate combining the two software infrastructures, for instance via the JavaGAT/SAGA project, and if found practical facilitate this development as much as possible.

8.2 Recommendations to AMC

- The main advantage of using the grid infrastructure is the high-speed interface. Together with VGFS and the GAP approach, there is a possible solution that provides remote data storage with transparent authentication and encryption.
- Replication of the data currently isn't done transparently. This is required for production usage. A second site would have to be determined (possibly the NIKHEF or an internal cluster) and modifications would be required to both the AMC firewall and VGFS in order to make this possible.
- Realize that the grid infrastructure is a complicated matter and a work-in-progress. The current level of availability would require manual verification of the remote files before they can be deleted locally. If used, monitor the performance and availability closely.

- Discuss usage of the grid infrastructure with VL-e and IBM. Determine the level of service required and if the other parties can supply this level. Don't focus blindly on only the grid infrastructure; both SARA and IBM can offer alternatives.
- Discuss where GAP responsibility lies; the current grid certificate can only be used for testing, thus an administrator would require his/her own certificate for production usage.
- Only after the above recommendations have been addressed would it be possible to look at using a GAP-based solution for client-institutions that wish to make frequent use of the genome sequencer. The described solution can be easily deployed at other sites but would require coordination in order to make use of cross-site encryption.
- Investigate other use-cases where the VL-e infrastructure can be of use within the AMC.

8.3 Recommendations to IBM

- Choose more strongly the role within the VL-e project. One possible opportunity would be to facilitate grid usage at other research institutions, so growing the scale of VL-e and facilitate it into becoming an infrastructure that can be part of certain IBM solutions.
- Position the current VL-e infrastructure well. Realize that it has drawbacks, but it has major benefits as well.
- Actively test IBM GMAS with the GAP solution. It is the most suitable IBM Healthcare and Life Sciences product technology-wise.
- Try to bridge the gap between recent cloud computing initiatives and the grid computing initiatives of the first half of this decade.

8.4 Recommendations to developers using grids

- There is a large difference in maturity and features between various grid middleware layers. Take the time to evaluate and choose wisely.
- Determine where using the grid would make the most sense. Use its strengths.
- Grids are inherently unreliable, this is their greatest weakness. When developing against grid middleware, assume the worst (as current grid middleware won't help you on this issue). Your program should either fail loudly or be able to buffer unavailability-issues.

References

- [1] Amazon Web Services. <http://aws.amazon.com>.
- [2] H. Bal et al. The distributed ASCI supercomputer project. *SIGOPS Oper. Syst. Rev.*, 34(4):76–96, 2000.
- [3] J.-P. Baud, B. Couturier, C. Curran, J.-D. Durand, E. Knezo, S. Occhetti, and O. Barring. Castor status and evolution, 2003. <http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0305047>.
- [4] G. Bell and J. Gray. What’s next in high-performance computing? *Commun. ACM*, 45(2):91–95, 2002.
- [5] BIG GRID. <http://www.biggrid.nl>.
- [6] *dCache, Storage System for the Future*, pages 1106–1113. Springer Berlin / Heidelberg, 2006.
- [7] Distributed.net. <http://www.distributed.net>.
- [8] Filesystem in Userspace. <http://fuse.sourceforge.net>.
- [9] I. Foster. *Grid Computing: Making the Global Infrastructure a Reality*. Wiley, 2003.
- [10] I. T. Foster and C. Kesselman. Computational grids. In *VECPAR*, pages 3–37, 2000. <http://citeseer.ist.psu.edu/article/foster98computational.html>.
- [11] F. Gagliardi, B. Jones, F. Grey, M.-E. Bgin, and M. Heikkurinen. Building an infrastructure for scientific grid computing: status and goals of the EGEE project. *Philosophical Transactions of the Royal Society A*, 363(1833):1729–1742, 2005.
- [12] Genome sequencer FLX system. <http://www.454.com/products-and-reagents/>.
- [13] S. Graham, C. Patterson, and M. Snir. *Getting up to speed: the future of supercomputing*. National Academy Press, 2005.
- [14] Grid operations get ready for LHC launch. *CERN Computer Newsletter*, 2007. <http://cerncourier.com/cws/article/cnl/31097>.
- [15] Grid storage management working group. <https://forge.gridforum.org/projects/gsm-wg>.
- [16] N. S. Hekster et al. *Numerically Intensive Computing Environment Evaluation*. IBM Redbook ZZ81-0245-01, 1991.
- [17] L. Hertzberger et al. Virtual Laboratory for e-Science: Mid-term progress report, 2007.

- [18] K. Howard. The bioinformatics gold rush. *Scientific American*, 283(1), 2000.
- [19] R. Kalmady and B. Tierny. A comparison of GSIFTP and RFIO on a WAN. CERN Technical Report.
- [20] M. Krigsman. ZDNet: Amazon s3 web service down. <http://blogs.zdnet.com/projectfailures/?p=602>.
- [21] Ölund, Lindqvist, and Litton. BIMS: An information management system for biobanking in the 21st century. *IBM Systems Journal*, 46(1), 2007.
- [22] B. Segal. Grid computing: the european data grid project. *IEEE Nuclear Science Symposium Conference Record*, 2000.
- [23] SETI@Home. <http://setiathome.berkeley.edu/>.
- [24] A. Tanenbaum and M. van Steen. *Distributed Systems: Principles and Paradigms*. Prentice Hall, 2007.
- [25] D. Thain and M. Livny. Building reliable clients and servers. In I. Foster and C. Kesselman, editors, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2003.
- [26] The Globus Alliance. <http://www.globus.org>.
- [27] The Globus Consortium. <http://www.globusconsortium.org>.
- [28] The Open Grid Forum. <http://www.ogf.org>.
- [29] Virtual Laboratory for e-Science. <http://www.vl-e.nl>.
- [30] VL-e proof of concept environment. <http://poc.vl-e.nl>.
- [31] World Community Grid. <http://www.worldcommunitygrid.org>.

A Genome sequencer transfer script

As mentioned in the section 'Integration into AMC sequencing process', we attempted to automate the transfer of data from the genome sequencer to the grid. In order to do so and allow the possibility of encryption we wrote a small script to assist the user of the sequencer.

Each of the following screens shows a step in the sequence flowchart in Figure 17.

Figure 19: Step 1

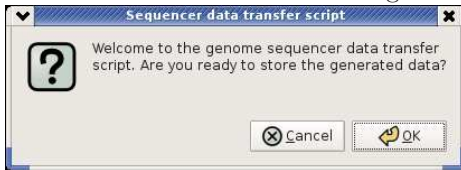


Figure 20: Step 2

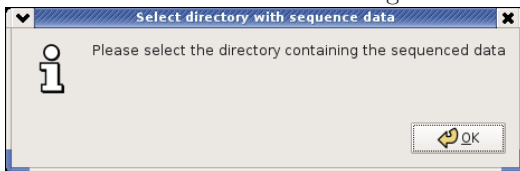


Figure 21: Step 3

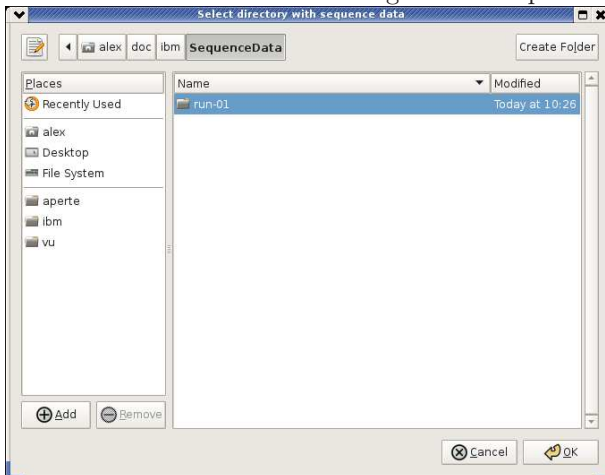




Figure 22: Step 4

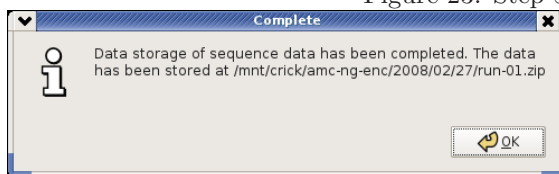


Figure 23: Step 5

B VGFS technical details

VGFS (Virtual Grid FileSystem) started as a project to test the VL-e infrastructure and grew into a full-scale FUSE filesystem. In order to work properly VGFS requires Linux 2.6.18 or newer, FUSE 2.6.5 or newer, the EGEE gLite APIs (liblfc, liblfcg, libdpm and libgfal) and for encryption it uses the ncrypt tool. VGFS also requires the C libc and glib libraries for compilation.

VGFS can be run from the command line (and in practice this is automated using a shell script). In order to use VGFS, the command line arguments that are supported are listed in Figure 24.

Figure 24: VGFS command line arguments

FUSE-related arguments:

```
-d DIRECTORY    - Specify the directory where the grid fs is mounted
-oallow_other   - Allow other users to access the grid fs
-oallow_root    - Allow root to access the grid fs
--singlethreaded - Maps to -s in the fuse layer,
                 doesn't background calls
```

EGEE-related arguments:

```
--help          - Merely print some helpful information to stdout
--host HOSTNAME - Specify the LFC grid hostname
                 (default: $LFC_HOST)
--path PATH     - Specify the start path (default: /grid)
--vo VO         - Log on using this virtual organization
                 (default: $LCG_GFAL_VO)
--transfer TRANSFER - Specify the default file transfer method
                 (either DIRECT_IO or BUFFERED, default: X)
--streams NR_STREAMS - Specify the number of parallel streams used for
                    GridFTP file transfers
                 (only does anything with --transfer BUFFERED)
--verbose       - Turn on verbose commandline output (to stderr)
```

VGFS can be easily used from the GAP by a Linux user or remotely via FTP, SFTP or CIFS/SMB. Getting it to work with NFS requires more patience, due to the filesystem semantics that NFS uses (see the chapters 6.4 and 6.5 for details). A screenshot of a Windows PC using VGFS via Samba can be found in figure 25.

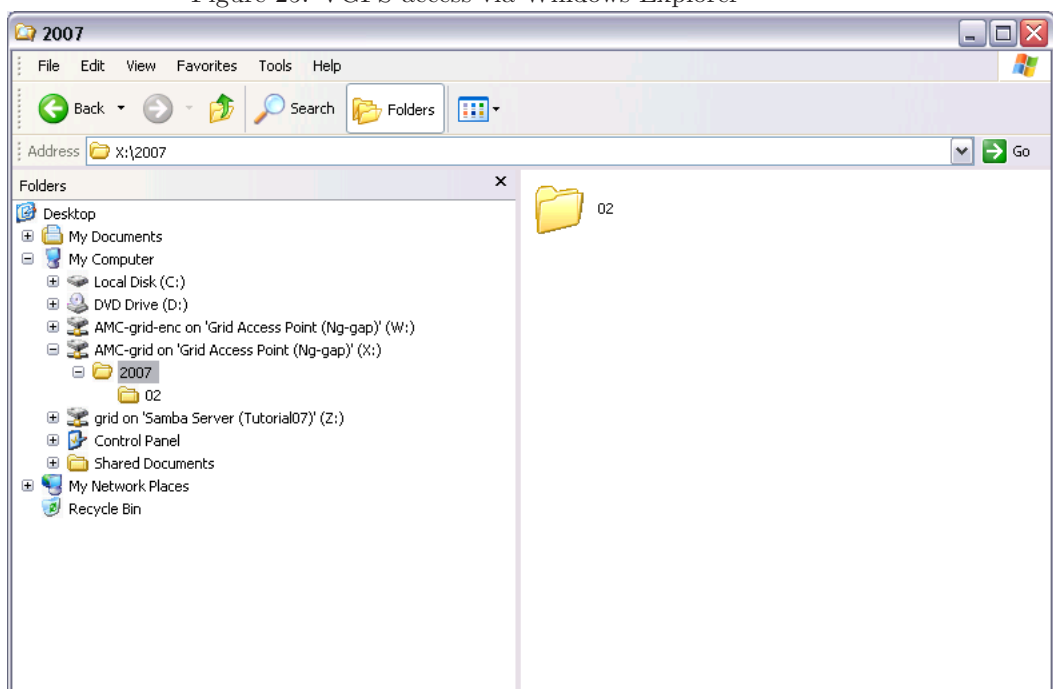
B.1 Status of VGFS development

VGFS, the grid filesystem interface layer used in the GAP solution, should be seen as a proof-of-concept. It works, but not all required features have been

integrated and in spite of the testing done it is not yet a tried-and-proven tool. There are a number of improvements still required:

- **Automatic retries at a later point in time.** Currently VGFS retries up to 10 times to store a file, after that the transfer fails. It should be possible to retry this at a later time automatically, so buffering the availability issues. As the availability issues were not discovered in time, this has not been implemented.
- **NFS read support.** Currently only writing via NFS works as expected. As FUSE matures and with enough testing read support via NFS should be possible, but at this point in time this is not the case.
- **Replication.** Currently there is no support in VGFS for automatically replicating data to multiple grid sites. There was not enough time to integrate this properly. As such, one of the key advantages of the grid has not been utilized.
- **Integrity-verification.** Currently the integrity of files is not checked after transfer. This can be done either locally via a file-hash database or via a Hydra-based grid service.
- **Notification.** Currently a lot of data is being logged. In an ideal case an IT administrator would receive a simple email after a file has been transferred through VGFS.
- **Distributed encryption.** Currently encryption is done using a local encryption key. These local keys would have to be distributed in a separate secure channel to other GAPs in the case of distributed encryption.
- **Complexity.** The GAP at the neurogenetics department currently runs a highly-customized Linux distribution. In order to proliferate the usage of GAP servers there would have to be a single properly-packaged set of VGFS software that can be easily installed and configured.

Figure 25: VGFS access via Windows Explorer



C IBM Healthcare and Life Sciences products

In this section we describe two of the other investigated IBM products. These products have been found to be less usable for the AMC case, but are useful in combination with the VL-e infrastructure in other cases.

C.1 IBM Content Management Offering (CMO)

In this section we take an in-depth look at the functionalities of the IBM Content Management Offering, the architectural design of the offering and the combination of this offering with other IBM solutions. We also take a look at possible pitfalls with relationship to this offering and the placement of this offering in comparison to the VL-e environment and other IBM technology.

The Content Management Offering (CMO) is a HIPAA-compliant centralized solution for archiving and storing medical information. Using IBM Content Manager and IBM Tivoli Storage Manager, CMO provides a flexible and high-capacity storage solution for medical data that allows easy integration and automation of storage tasks.

CMO uses DICOM to facilitate interoperability between PACS's (Picture Archiving and Communication System), and ensures long-term storage that easily interfaces with existing PACS systems. CMO is an open system, supporting HL7.

CMO is scalable, secure and can use High Availability platforms in order to ensure 24/7 access. CMO allows flexible deployment, on systems ranging from single workstations to a cluster of SMP Unix servers.

Functional overview

PACS integration and data formats CMO allows data in varying formats and standards to be collected and stored in a centralized storage facility. Data from multiple PACS of different vendors can be integrated and queried. Data that doesn't conform to standards, or data that needs to be transformed before storage, can be modified in-transit using a Curator. If invalid data is sent, then this is rejected by CMO in order to maintain coherence.

By default, the DICOM interface agent is supplied with CMO. Other interface agents can easily be integrated, letting CMO speak other data formats with ease. A HL7 interface is available, allowing DICOM and HL7 data to be combined and queried via one central system.

CMO handles both static, referenceable data and dynamic transactional data.

Administration

Administrators can monitor and manage the CMO via the web-based CMO Console. Through this interface the administrator can also configure users and access rights to the console, configure access to DICOM and HL7 devices, usage of Curator data-transformation scripts and configure upload rules. Through

the web-based CMO Console an administrator can view all current and historic alarms, and obtain an audit trail.

Querying

The CMO provides basic querying of medical data based on either patient information or studies performed. Meta-data, like a description of who performed the study, what the study was about and when the study took place, is easily obtained via the CMO Console.

Non-functional overview

Security and Authentication CMO provides access based on user/group-rights and access control lists. Access control can be determined on a per-object level. Default rights can be given on a group or institution level.

Encryption in CMO is done using an external library via the user configurable workflow, together with SSL/TSL protocols in order to ensure secure connections. DICOM messages have built-in support for digital signatures, which CMO supports. Signatures are also used in recent HL7 standards in order to ensure integrity.

Privacy

CMO provides an anonymity/pseudonym layer which filters out all data that can be used to identify a patient, for example from DICOM data. This layer is essential when sharing research data with other organizations and provides a means to ensure patient confidentiality.

Scalability

CMO is based on IBM Content Manager and Websphere Application Server, both which can be scaled vertically and horizontally. Together with scalable storage solutions, this provides the ability to scale and upgrade CMO depending on the performance required and amount of users that access the system.

High Availability

CMO utilizes existing high availability platforms, such as IBM AIX with High Availability Cluster Multi-Processing. CMO can also make use of non-IBM high availability platforms, like Sun's SUN Cluster and Microsoft's Cluster Server. This decreases the initial costs of CMO deployment and allows an easier integration within the existing infrastructure.

Performance

In performance tests, a single IBM Blade server was shown to perform fine under a load of 50-100GB per hour. The actual performance naturally depends on the specific production environment and server specifications. CMO components scale horizontally and vertically, and the solution is able to address very high loads.

Maintainability

Most of the work is completed once CMO has been deployed, but there is also a small degree of maintenance required. This can be outsourced to external IBM administrators. In practice, users frequently discover new ways that CMO could be leveraged and in the beginning developing extensions for CMO is a common task. This too is often done in collaboration with IBM.

Stored objects that are less likely to be accessed can be automatically moved to a different storage pool, reducing the need for manual backups and archiving. This can be configured via the CMO Console.

Auditing

CMO provides a logging of all user interactions, providing administrators a means of determining which users accessed what data.

Ease-of-use, Ease-of-integration

Via the CMO Console, integration with existing PACS systems is simple for an administrator. Users of the PACS systems don't have to switch to a different interface, which will ease the efforts of integration.

Existing IBM and non-IBM storage can be integrated into CMO using IBM Tivoli Storage Manager. By using existing infrastructure, deployment and integration of CMO can be done without having to replace existing systems.

Architecture

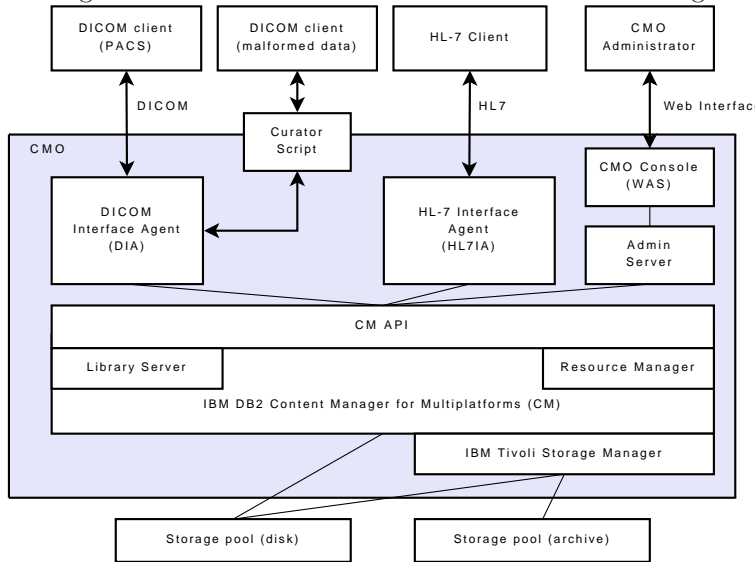
The architecture used in CMO is split into two layers: the top layer is a collection of distributed loosely coupled components (called Interface Agents) and the bottom layer consists of IBM Content Manager. Communication between these two layers is done using the (open) CM API. An Admin Server controls and monitors the Interface Agents.

Due to the modular architecture (Figure 26), it is possible to extend CMO by using the CM API directly. Doing so, data can be integrated from non-DICOM/non-HL7 while enabling this data to be queried and retrieved via the various interfaces.

Internally, the Content Manager is split into a Library Server and a Resource Manager. These nodes can be deployed on a single or on multiple servers, depending on the performance required. For High Availability and performance reasons, these nodes can also be duplicated. Using a load balancer it is then possible to scale the Content Manager to any required level.

CMO can use various options for storage via the IBM Tivoli Storage Manager. Grouped together in storage pools for either direct or archive access, CMO can intelligently move stored objects from one storage pool to another after a fixed period of inactive use.

Figure 26: The architecture of the IBM Content Management Offering



CMO Case: Merck

Merck is an internationally-operating research-driven pharmaceutical company that aims to produce next generation therapy for diseases such as Alzheimer's, cancer and Parkinson's Disease. In clinical trials biomarkers, indicators for special conditions and diseases, are tracked. Before implementing a solution with CMO the images generated at these trials were burned on CD or DVD, and sent from the imaging sites to Merck. This process was time-consuming and error-prone, leading to trial drop-outs. With increasing image sizes, this was starting to lead to even larger problems.

The key component of the solution IBM Global Business Services implemented was CMO. The imaging sites now send the large images electronically to Merck in a matter of seconds. Contracted research organizations can now also be given fine-grained access to the images stored within the organization. The meta-data automatically organized when storing images allows easy querying and retrieval of images, which speeds research up considerably compared to manually searching through DVDs. All in all, trial times have been reduced significantly and Merck estimates a reduction of trial administration costs of up to US \$1M per year.

Possible Issues

What should be considered when implementing CMO is that a solution would have to avoid having a single point of failure. By integrating all these systems many users become dependent on CMO, thus the reliability has to be second-to-none. Due to these factors, deployment of CMO must be done together with

one of the mentioned High Availability solutions. The speed of accessing the various PACS will also become dependent on the speed and reliability of the internal network infrastructure, thus it is important to do extensive stress-tests before CMO is put into production use.

Relation to VL-e

CMO provides a useful service to hospitals by integrating different versions of, for example, DICOM. Storage of different PACS systems can thus be centralized. VL-e provides a central storage infrastructure which is capable of storing large amounts of data off-site. CMO can be used without IBM Content Manager and instead use GMAS as a storage layer, however Content Manager itself is unable to use GMAS. VL-e could also be used for archiving of backups of CMO, or a combination of the two.

Related IBM technology

CMO and the Grid Medical Archive Solution (GMAS) provide some overlap in functionality, however they complement each other when implemented correctly. The main differences between the two solutions are:

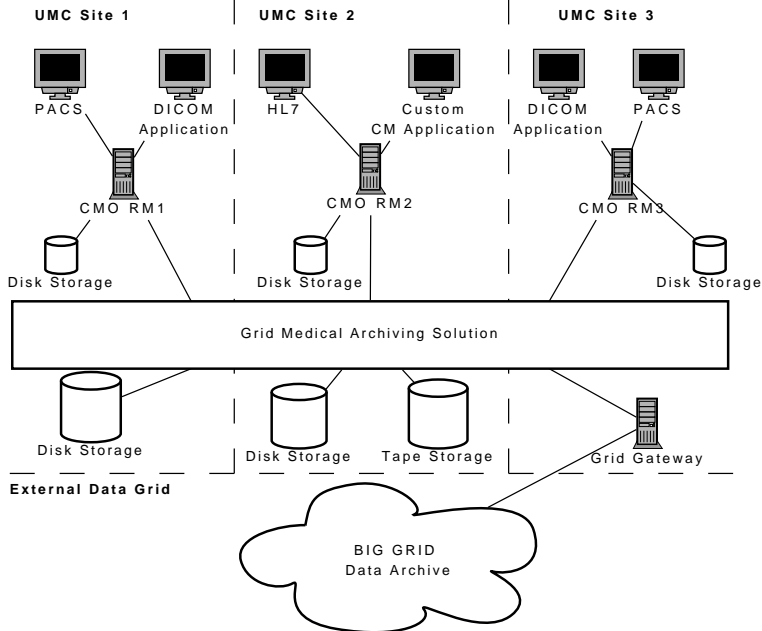
- CMO is content-aware, GMAS is not. CMO can thus translate differing standards on-the-fly, allowing integration of previously incompatible systems.
- GMAS provides integration of storage on multiple sites, CMO uses FileNet for this purpose.

By combining CMO and GMAS, organizations obtain the possibility to combine storage across multiple sites while integrating existing incompatible PACS into one coherent system. Integration of CMO and GMAS is possible, but this solution would lack IBM's Content Manager. How CMO in practice could use GMAS depends on the situation. CMO could for example use GMAS for archiving only, or CMO could be installed at multiple sites with a single GMAS storage layer underneath. The latter example would require a fast interconnect between the sites in order to provide fast access to the resources.

We have determined that GMAS could also use the VL-e infrastructure, thus a combination of the three is also possible. An example of such a solution is shown in Figure 27.

In this example, three sites are connected together using GMAS, which allows users from one site to access storage on another site. With a CMO server at each site, existing PACS systems can be connected together to allow querying of the combined data. GMAS is connected to the VL-e infrastructure (via a Grid gateway), which allows transparent use of BIG GRID for archiving and backup purposes. The distribution of data objects across the different methods of storage can be managed centrally via the GMAS interface, while configuration and integration of PACS systems is handled via the web interface of the CMO systems

Figure 27: A possible combination of CMO, CMAS and the VL-e infrastructure



In conclusion, IBM's CMO allows the integration of differing existing PACS systems into one centralized storage system, enabling functionality's like querying across different, previously incompatible, systems. Alone, CMO provides a solution for a single site of an organization, however this can be combined with GMAS and/or VL-e to provide a more flexible cross-organization solution.

C.2 IBM Clinical Genomics Solution (CGS)

The Clinical Genomics Solution (CGS) allows healthcare institutes to combine various types of medical research data with clinical care data, providing researchers a means to access and analyze previously-incompatible genomics data via a single, easy-to-use data mining interface.

CGS is a complete solution, combined from a number of industry-proven components allowing a flexible deployment depending on the case-by-case needs of the institution. Many different types of healthcare-related data formats are supported, including the usage of phenotypic data from clinical databases and genetic studies from microarrays.

CGS is able to integrate many different types of data formats from a vast number of sources, and provides the flexibility to easily extend the means of data collection to new data formats.

Researchers, instead of having to manually correlate and combine sources, can now easily build their own data queries covering the many data sources in CGS without having to program. Queries can be saved for later use, speeding up the time and reliability of research even more.

Finally, the de-identification of data sources is done directly at the start. Patient data is removed before any data is stored in CGS, ensuring patient confidentiality. Physical security is ensured via IBM's Tivoli-based security framework, allowing security policies, monitoring and auditing of CGS.

Functional overview

Data formats

Via the data processor engine, which transforms various types of files into the Clinical Genomics database format, CGS is able to handle the following data formats out of the box:

- HL7-CDA version 3 (allowing interoperability of electronic health records)
- MAGE-ML (Microarray Gene Expression Markup Language)
- BSML (Bioinformatic Sequence Markup Language)
- HAPMAP (used for determining genetic similarities)
- ODM (Ontology Definition Meta-Model)

A set of graphical user-interfaces is provided to allow manual uploading of data into CGS. The so-called shredders for each data format collect all the required information from the data, after which this data is stored into the Clinical Genomics database. Additional shredders can be easily integrated into CGS, allowing any type of existing data to be integrated. These also allow the integration of plain-text information, upon which users can do a full-text search.

User interface

The researcher accesses CGS via either the included Data Discovery and Query Builder (DDQB) or a third-party analysis, data mining or visualization tool like SAS, Spotfire or deCODE CGM-D. DDQB allows researchers to easily create complex queries via a web interface, without the need to program. The output from a DDQB query can be exported to XML or CSV, or it can simply be viewed via the web browser. As an example, a researcher can create a query specifying all kidney-biopsy procedures where the diagnosis included the finding of blood in the patient's urine, where the patient is a non-caucasian man with an age of 60 or more. With the availability of genomic data, the researcher can search for specific biomarkers combined with clinical findings in order to obtain correlations between genotypic and phenotypic information.

Non-functional overview

Security and Authentication

Data access via DDQB can be configured by the administrator to restrict access to a particular data source, or even to a certain subset of a data source. Access control can be defined at a user or a group-level, giving the administrator a range of access control methods.

Privacy

All privacy-sensitive data sources are routed via the IBM Universal De-identification Platform (UDiP). This rule-based engine is configured to anonymize identifying data and provides a range of algorithms to do so. Via UDiP, encryption, scrambling, deletion and masking of sensitive information is possible. Non-supported data sources can be integrated, providing that the data source is based on XML.

Scalability and High Availability

Each component of the Clinical Genomics Solution can be scaled horizontally and vertically, providing a high degree of redundancy and availability due to the inherent scalability of the solution. The solution can be expanded with additional data sources over time, thus allowing a small-scale deployment to be grown over time to encompass other types of information.

Due to the mission-critical nature of this type of infrastructure, availability of CGS must be guaranteed. This is done due this solution being built upon tried-and-proven IBM technologies like IBM DB2 and IBM WebSphere and IBM pSeries systems.

Auditing

Audit logs can be stored in the Clinical Genomics database, containing all queries performed together with information on who performed them. These logs too can be set to be browsed and queried by administrators via the DDQB web interface, providing complete knowledge about who has had access to which data.

Ease-of-use, Ease-of-integration

Due to the availability of data shredders for many different types of data formats, integration of the Clinical Genomics Solution is easy to accomplish. Due to the ease-of-use of the Data Discovery and Query Builder, researchers can make the most out of CGS with a minimum of training.

Architecture

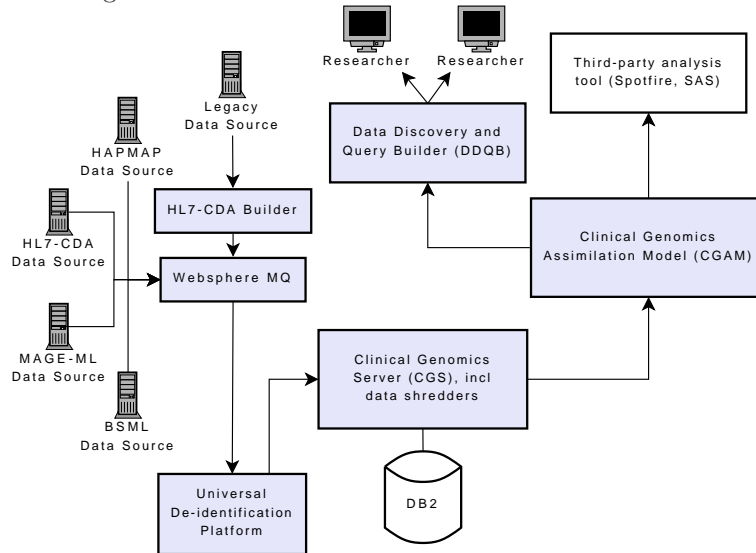
Components

- The HL7-CDA Builder provides a collection of interfaces for building the two most common HL7 messages: Clinical Document Architecture (CDA) and UpdatePatient. This tool dramatically decreases the time required to integrate legacy systems into the Clinical Genomics Solution.
- The Universal De-identification platform is the IBM solution for protecting patients privacy. By using a flexible rule-based engine, the specific type of de-identification is customized according to the needs of the healthcare institute where CGS is deployed.
- The Clinical Genomics Server provides a means for translating (shredding) XML documents into a relational database. This database holds the complete contents of the data extracted from formats like HL7, MAGE-ML and BSML.
- The Clinical and Genomics Assimilation Model (CGAM) contains the data model for the operational data store, upon which analysis tools like the Data Discovery and Query Builder perform their queries. The model used was developed by biomedical scientists, academic healthcare institutions and hospitals in collaboration with IBM and provides the basis for usage of CGS by third-party tools.
- The Data Discovery and Query Builder is a flexible user-oriented interface, developed in collaboration with Mayo Clinic, that allows researchers and physicians to easily access the data stored in the CGAM component. Via this single interface, users have access to a multitude of previously-separated data sources.
- IBM WebSphere MQ is the Enterprise Service Bus used in the Clinical Genomics Solution.

CGS Case: Karolinska Institutet

The Swedish Karolinska Institutet is one of Europe's largest medical universities and a well-known leader on medical and genomics research. Together with IBM, Karolinska deployed Sweden's first IT-enabled biobank that allows researchers to identify links between genetics, environment and diseases. With

Figure 28: The architecture of the IBM Clinical Genomics Solution



the new biobank, Kaorlinska is able to improve healthcare for its patients using information-based personalized medicine and accelerate research into mechanisms of disease by linking phenotypic and genotypic data.

With the new biobank, Karolinska has combined the following data sources:

- Laboratory tests and drug responses
- Family histories
- Lifestyle information
- Genetic data and gene variances
- Gene-environment interactions

Key components of the complete IBM solution were the deployment of components of CGS, IBM consultancy and services and the usage of IBM WebSphere Portal Server for information access and project collaboration.

CGS Case: Mayo Clinic

The Mayo Clinic comprises a network of hospitals and clinics across 3 states in the United States. Mayo Clinic employs over 46800 and is pushing the limits of proteomic research.

The key reason that the Mayo Clinic deployed components of CGS was that researchers and physicians had to be able to access to clinical, genetic and proteomic data from all of the 64 communities that the Mayo Clinic serves. The Mayo Clinic had over 4 million electronic patient records dispersed across the

institute in different formats, with various third-party analytical and clinical decision-support tools. This led to inefficient processes and laborious attempts in manually combining data sources.

The IBM solution for the Mayo Clinic combined the electronic patient records with lab test results, billing and demographic data. Key components of the solution were the deployment of WebSphere Application Server, services of IBM Healthcare & Life Sciences and Systems Group and components of the Clinical Genomics Solution. The Data Discovery and Query Builder was developed in close collaboration with the Mayo Clinic to aid researchers and physicians, which today is one of the key components of the IBM Clinical Genomics Solution.

Possible Issues

The Clinical Genomics Solution is a combination of various proven IBM products (UDiP, DB2, DDQB) which have been shown to be easy to integrate. Although each of these products have been shown to work well, the solution doesn't appear to be deployed anywhere in its entirety. The Clinical Genomics Solution is more of a framework, which IBM in collaboration with the clinical or research center can modify and expand depending on the needs of the organization.

The Clinical Genomics Solution is used primarily for integrating the various data sources across an organization. One problem that is frequently encountered is the usage of legacy, proprietary and even undocumented protocols and data formats. This will greatly complicate the integration of the solution with the existing systems. Deployment of CGS should be focused primarily on the most easily integratable data sources in order to provide 'quick wins' and demonstrate the user interface to users, allowing them to experiment with the system and provide feedback. In a next iteration, more complicated data sources can then be integrated together with modifying the system according to the wishes of stakeholders and early adopters.

Relation to VL-e

What has to be realized is that the CGS is not primarily a high-capacity data storage solution; it provides a means for integrating multiple data sources. The prime usage for VL-e is distributed access across a virtual organization to large amounts of data.

One possible scenario is that multiple sites have CGS deployed, and periodically that the data models from all sites are uploaded to the grid and integrated via the grid at each site. This would be a possible situation, as long as synchronization problems are avoided.

There is however a discrepancy: VL-e is most useful for distributing very large amounts of data, however CGS works on a large number of relatively small messages. No doubt that combined all the data in CGS is large, however the question is if this amount would be large enough to even justify the usage of a grid infrastructure.

A more useful scenario is that multiple sites send their data to a central Clinical Genomics Server, possibly through a dedicated intra-site network. Users can access the central DDQB server and retrieve the results for analysis. In such a case, the grid infrastructure could be useful, for example for CGS-backups, archiving and long-term storage of the data. In any other situation a grid-hosted database service would be required.

Related IBM technology

- The IBM Content Management Offering is also used for combining multiple data sources, however this offering is focused on distributing (high-resolution) medical images throughout an organization. In such a case a grid infrastructure is more useful due to the large amount of data collected.
- The IBM WebSphere Federation Server is a comparable solution to CGS, but it provides a broad, non-healthcare solution. CGS uses a number of the WebSphere Federation Server components, but they are two separate solutions for different cases. Because of this we will not discuss the IBM WebSphere Federation Server separately.

In conclusion, the IBM Clinical Genomics Solution is a fitting framework for the deployment of an IT-enabled biobank. By providing a means for combining many different data sources used in the healthcare and life sciences sector, CGS is a useful solution for integrating incompatible clinical and research systems.

Integration with the VL-e infrastructure is possible, but the options to do this effectively are limited to archiving and long-term storage due to the nature of the data stored in CGS.

D Other internship activities

In this section other activities are described that took place during the internship period at IBM.

D.1 VL-e Medical Presentation

Every few weeks the VL-e Medical Virtual Organization comes together and discusses a particular topic, often with a presentation. These talks are alternated between technical (grid-related) topics and healthcare topics, so broadening the perspective of all involved.

In November 2007 I was allowed to give a talk on the proposed solution for the case discussed in this document. The feedback and contacts gained because of this were invaluable in the completion of my internship. The slides are available via my personal website <http://www.alextrreme.org> and contain an overview of the solution as described in this document.

D.2 Project "Zorgkonijn"

Via my supervisor at IBM I was able to help in the "Zorgkonijn" project, an initiative of Achmea to make the stay of small children in a hospital more comfortable. The idea is that each child has a toy rabbit which has various means of in- and output (twisting the ears and pushing the nose, speakers, wifi). IBM developed the software for the rabbit, together with a web-based portal via which parents could send emails to the rabbit. These are then read to the child via tekst-to-speech software.

The missing part that I helped to develop was translating the tekst of such an email to a form suitable for playback via the rabbit. Using IBM Embedded ViaVoice I developed a small application that turned the tekst into speech and made this speech available in the format required for transmitting to the rabbit. I'd like to thank Arjo Poldervaart for his cooperation and I hope what I did was useful in completing this project.

D.3 Literature Study

Next to my internship I also successfully completed a literature study for the VU. The topic was "Web 2.0 and Grids" and the paper can be downloaded from my personal website at <http://www.alextrreme.org>. The paper is closely related to this document and gives an overview of how various 'Web 2.0' topics like Cloud Computing can be compared to grid computing. I'd like to thank Thilo Kielmann for being my supervisor for this study.

D.4 AMC and UvA

In order to facilitate communication and thanks to cooperation at the institutes I was able to work at both the University of Amsterdam (UvA) and the AMC.

At the UvA I worked one day a week for a period of 2 months in order to collaborate with key VL-e Medical members and learn from their experiences with the grid. During the last month of my internship I worked full-time at the AMC in order to deploy, test and integrate the GAP solution.

D.5 IBM, workshops and other activities

For the remainder of my time I was active at the headquarters of IBM Netherlands in Amsterdam. During this time I took part in workshops, regular meetings, informal discussions and accompanied my supervisor to customers of IBM. The experiences gained were valuable and insightful and often included topics where I previously knew little about (IBM mainframes, IBM software products, IBM xSeries/pSeries hardware). All in all my time as an intern at IBM was a valuable experience.