



UNIVERSITY OF PISA

ABSTRACT

STORAGE MANAGEMENT AND
ACCESS IN LHC COMPUTING GRID

by Dr. Flavia Donno

Supervisors:

Prof. Gigliola Vaglini
Dr. Andrea Domenici

Department of Computer Engineering

One of the big challenges in Grid computing is storage management and access. Several solutions exist to store data in a persistent way. In this work we describe our contribution within the Worldwide LHC Computing Grid project. Substantial samples of data produced by the High Energy Physics detectors at CERN are shipped for initial processing to specific large computing centers worldwide. Such centers are normally able to provide persistent storage for tens of Petabytes of data mostly on tapes. Special physics applications are used to refine and filter the data after spooling the required files from tape to disk. At smaller geographically dispersed centers, physicists perform the analysis of such data stored on disk-only caches. In this thesis we analyze the application requirements such as uniform storage management, quality of storage, POSIX-like file access, performance, etc. Furthermore, security, policy enforcement, monitoring, and accounting need to be addressed carefully in a Grid environment. We then make a survey of the multitude of storage products deployed in the WLCG infrastructure, both hardware and software. We outline the specific features, functionalities and diverse interfaces offered to users. We focus in particular on StoRM, a storage resource manager that we have designed and developed to provide an answer to specific user request for a fast and efficient Grid interface to available parallel file systems. We propose a model for the Storage Resource Management protocol for uniform storage management and access in the Grid. The black box testing methodology has been applied in order to verify the completeness of the specifications and validate the existent implementations. an extension for storage on the Grid. We finally describe and report on the results obtained.

TABLE OF CONTENTS

List of Figures.....	v
List of Tables.....	vii
Acknowledgements.....	viii
Glossary	ix
Preface.....	xv
1. Introduction	1
1.1 Problem Statement.....	2
1.2 Contribution of this Thesis.....	2
1.3 Outline.....	3
2. Grid Computing and Architecture	5
2.1 Grid Computing.....	6
2.2 Grid Computing vs. Distributed Computing.....	9
2.3 Grid Architecture.....	10
2.3.1 Security.....	12
2.3.2 The Information System.....	14
2.3.3 The Workload Management System.....	14
2.3.4 Data Management and Replication	16
2.3.5 Storage Management and Access	18
2.4 Current Middleware Solutions.....	19
2.4.1 Condor.....	19
2.4.2 Globus	21
2.4.3 Legion and Avaki.....	25
2.4.4 gLite	28
2.4.5 ARC	33
2.5 Current Grid Infrastructures.....	34
2.5.1 Worldwide Large Hadrons Collider Computing Grid (WLCG).....	34
2.5.2 Open Science Grid (OSG)	35
2.5.3 Nordic Data Grid Facility (NDFG)	35
3. The Challenge of Data Storage in WLCG.....	37
3.1 Introduction.....	37
3.1.1 Constrains for Distributed Computing and Storage.....	37
3.1.2 Data Model.....	37
3.1.3 Storage Systems and Access Patterns.....	39
3.2 Multi-Tier Architecture.....	40
3.2.1 Discussion of the Tier Model.....	43
3.3 High Level Physics Use Cases	43
3.3.1 Reconstruction.....	43

3.3.2 Main Stream Analysis.....	44
3.3.3 Calibration Study.....	45
3.3.4 Hot Channel	45
3.4 Grid Data Access in Practice – The Client Side.....	46
3.5 Storage Requirements	48
3.5.1 Overview.....	49
3.5.2 The Storage Interface.....	49
3.5.3 The Storage Classes.....	52
4. Storage Solutions	54
4.1 A motivating example.....	54
4.2 Hardware storage technologies.....	55
4.2.1 Disk based technologies.....	55
4.2.2 Tape based technologies	56
4.3 Network based technologies.....	57
4.4 Software solutions.....	59
4.4.1 Disk Pool Managers	59
4.4.1.1 dCache.....	60
4.4.1.2 LDPM.....	60
4.4.1.3 NeST.....	60
4.4.1.4 DRM.....	61
4.4.1.5 SAM.....	61
4.4.2 Grid Storage	61
4.4.3 Distributed file systems.....	62
4.4.4 Parallel file systems	62
4.4.4.1 GPFS	63
4.4.4.2 LUSTRE	63
4.4.4.3 PVFS.....	63
4.4.4.4 StoRM.....	63
4.4.5 Mass Storage Systems.....	64
4.4.5.1 CASTOR.....	64
4.4.5.2 HPSS.....	65
4.4.5.3 TSM.....	65
4.4.5.4 DMF.....	65
4.4.5.5 SRB.....	66
5. File Access and Transfer protocols	67
5.1 Data Transfer Protocols: GridFTP	67
5.2 File Access Protocols.....	69
5.2.1 RFIO.....	70
5.2.2 dCap	71
5.2.3 xrootd.....	72
6. The Storage Resource Manager Interface.....	74
6.1 Motivations and History.....	74

6.1.1 Motivations and requirements.....	74
6.1.2 History and related work	75
6.2 The Storage Resource Manager Interface	76
6.2.1 The SRM v2.2 methods	78
6.2.1.1 Space management functions	78
6.2.1.2 Directory functions.....	79
6.2.1.3 Permission functions	79
6.2.1.4 Data transfer functions.....	80
6.2.1.5 Discovery functions.....	80
6.3 The Data and Semantic Model.....	81
6.3.1 Basic definitions.....	81
6.3.1.1 Storage Element.....	81
6.3.1.2 Space.....	82
6.3.1.3 Copy.....	83
6.3.1.3 Handle	83
6.3.1.4 File	83
6.3.2 Functions and relationships.....	84
6.3.3 Constrains.....	86
6.4 Detailed discussion on the model	87
6.4.1 The Space	87
6.4.2 Files, Copies, and Handles.....	91
7. Information Model for SRM.....	98
7.1 The Storage Element	98
7.2 The Access Protocol.....	99
7.2.1 The Access Protocol Use Cases.....	100
7.3 The Storage Component.....	101
7.3.1 The Storage Component Use Cases.....	102
7.4 The Storage Area.....	102
7.5 The VO-Storage Area Association.....	105
7.6 The Storage Paths	106
7.7 Free and available space.....	106
7.8 The Storage Component GLUE class.....	107
8. Test and Evaluation of SRM Implementations.....	109
8.1 Theory of Testing.....	109
8.1.1 The Black Box Methodology	110
8.1.1.1 Equivalence Partitioning	111
8.1.1.2 Boundary-value Analysis	111
8.1.1.3 Cause-Effect Graphing.....	112
8.1.1.4 Error guessing.....	114
8.2 The SRM Case	114
8.3 Reducing the test set: a use-case based analysis.....	116
8.4 Experimental results	123

8.5 The S2 language.....	127
8.6 Conclusions: how to optimize test case design.....	128
9. Conclusions	130
9.1 Standardizing the Storage Interface.....	130
9.2 The SRM Protocol: current limitations and open issues	131
9.3 Toward SRM version 3.....	132
9.4 Protocol validation	133
9.5 Future work.....	134

LIST OF FIGURES

<i>Number</i>	<i>Page</i>
1.1 Virtualization of computing and storage resources	1
2.1 A Computing Cluster	7
2.2 Intra/Extra-Grids	8
2.3 The evolution of the Grid	8
2.4 Architecture of a Grid	10
2.5 The layered Grid Middleware architecture	11
2.6 Authentication, authorization, delegation on the Grid	13
2.7 Computing Element and Worker Nodes	15
2.8 Grid Filenames	17
2.9 Data Management Service Components	18
2.10 The Directory Information Tree	23
2.11 The GRAM Architecture	23
2.12 The GASS Architecture	25
2.13 The Legion Architecture	26
2.14 The main components of the gLite middleware	28
2.15 The MDS Information Service in WLCG	30
2.16 The R-GMA Architecture	31
2.17 The job flow in gLite	33
3.1 Data flow in a typical HEP experiment	38
3.2 Multi-tier architecture of distributed computing centers	40
4.1 From complex expensive to inexpensive storage solutions	55
4.2 A user's request for a file located on a tape system	56
4.3 Example configuration of a SAN installation	58
4.4 The StoRM Architecture	64
4.5 The CASTOR architecture	65
5.1 Third-party transfer	68
5.2 The xrootd system with data servers and load balancers	72
6.1 The SRM v2.2 Space, File, Copy, and Handle UML class diagram	89
6.2 The SRM v2.2 Space UML state diagram	90
6.3 Copies and handles of a file in a Storage Element	92
6.4 The SRM v2.2 UML File class state diagram	94
6.5 The SRM v2.2 UML SURL_Assigned File state diagram	95
6.6 The SRM v2.2 UML Copy class state diagram	96
6.7 The SRM v2.2 UML Handle class state diagram	96
6.8 The SRM File creation use case activity UML diagram	97
7.1 The SE description in the GLUE schema	99
7.2 The Storage Component in the GLUE schema	101

7.3 Shared Storage Area	103
7.4 Shared Storage Area with no dynamic space reservation	104
7.5 Storage components and Space Tokens	104
7.6 The Storage Area in the GLUE schema	105
7.7 The Storage Component Class is optional	107
7.8 The Storage Service description in GLUE v1.3	108
8.1 Cause-effect symbols	112
8.2 List of causes and effects for the srmReserveSpace method	120
8.3 Cause-effect graph for the srmReserveSpace method	121
8.4 The web pages associated to the test results	123
8.5 Results of the availability tests for 6 implementations	124
8.6 Basic tests	125
8.7 Interoperability tests	126
8.8 Use Case tests	126
8.9 S2 Example	128

LIST OF TABLES

<i>Number</i>	<i>Page</i>
3.1 CPU, Storage and Network resources provided by CERN (Tier-0)	41
3.2 CPU, Storage and Network provided by CERN (Tier 0) per experiment	41
3.3 CPU, Storage and Network resources provided by the 11 Tier-1 sites	42
3.4 CPU and Storage resources provided by the Tier-2 centers in 2008	42
4.1 Comparison between the main SAN and NAS features	59
8.1 Equivalence partitioning classes for srmReserveSpace	118
8.2 List of test cases for srmReserveSpace	120
8.3 The decision table for cause-effect graph analysis for srmReserveSpace	122

ACKNOWLEDGMENTS

I wish to express sincere appreciation to Prof. Gigliola Vaglini and Dr. Andrea Domenici for their assistance in the preparation of this manuscript and for their supervision during my PhD studies. In addition, special thanks to Prof. Mirco Mazzucato and Dr. Antonia Ghiselli from INFN (Italian National Institute of Nuclear Physics) who made possible my participation to several Grid projects. I am especially grateful to the following people who are a big source of inspiration and from whom I learned everything I know about SRM and much more: Jean-Philippe Baud and Maarten Litmaath from CERN (Switzerland), Arie Shoshani, Alex Sim and Junmin Gu from LBNL (U.S.), Timur Perelmutov from FNAL (U.S.). I can never stop admiring the code written by Jirí Mencák from RAL (UK) for the S2 interpreter: it is really a must for all students approaching object oriented and C++ programming. A big thank you to the colleagues of the StoRM project from INFN and ICTP (Luca Magnoni, Riccardo Zappi, Ezio Corso and Riccardo Murri) for the many discussions and for sharing with me the difficult steps involved in the design and development of a software product for a production Grid environment. Finally I would like to thank my bosses at CERN, Les Robertson, Ian Bird and Jamie Shiers who always encouraged and supported me in the difficult task of “making things work”!

A special thank goes to the father of my son, Heinz, who has shared with me the difficult moments and has strongly supported me discussing many technical issues, while providing Kilian with many “Unterhosenservices”. A big hug and a huge “Bussi” to my little son Kilian, who allowed me to work on my PhD project during the first months of his life.

Finally, I would like to thank my mother and my father who have always been by my side, encouraging me in pursuing this career and giving me the possibility to live this wonderful and exciting experience.

GLOSSARY

AC	Attribute Certificate
ACL	Access Control List
AFS	Andrew File System
API	Application Programming Interface
ARC	Advanced Resources Connector
BDII	Berkeley Database Information Index
BESTMAN	Berkeley Storage Manager
CE	Computing Element
CA	Certification Authority
CAS	Community Authorization Server
CASTOR	CERN Advanced STORage Manager
CERN	Conseil Européen pour la Recherche Nucléaire
CLI	Command Language Interface
DAP	Data Access Point
DAS	Direct Attached Storage
DCACHE	Disk Cache
DESY	Deutsches Elektronen-SYNchrotron
DIT	Directory Information Tree
DLI	Data Location Interface

DM	Data Management
DMF	Data Migration Facility
DN	Distinguished Name
DPM	Disk Pool Manager
DRM	Disk Resource Manager
DRS	Data Replication Service
DST	Distributed Storage Tank
EDG	European Data Grid
EGEE	Enabling Grid for E-ScienceE
FNAL	Fermi National Accelerator Laboratory
FQAN	Fully Qualified Attribute Name
FTS	File Transfer Service
GAA	Generic Authorization and Access
GASS	Global Access to Secondary Storage
GDMP	Grid Data Mirroring Package
GFAL	Grid File Access Library
GG	Grid Gate
GIIS	Grid Information Index Service
GLUE	Grid Laboratory for a Uniform Environment
GOC	Grid Operation Center
G-PBOX	Grid Policy Box

GPFS	General Parallel File System
GRAM	Globus Resource Allocation and Management
GRIP	Grid Resource Information Protocol
GRIS	Grid Resource Information Service
GSI	Grid Security Infrastructure
GSM-WG	Grid Storage Management – Working Group
GSS-API	Generic Security Service Application Programming Interface
GT	Globus Toolkit
GUID	Grid Unique Identifier
HEP	High Energy Physics
HPSS	High Performance Storage System
ICTP	Abdus Salam International Center for Theoretical Physics
INFN	Istituto Nazionale Fisica Nucleare
IS	Information Service
JDL	Job Description Language
LB	Logging and Bookkeeping Service
LBNL	Lawrence Berkeley National Laboratory
LCAS	Local Centre Authorization Service
LCG	LHC Computing Grid
LCMAPS	Local Credential Mapping Service
LDAP	Lightweight Directory Access Protocol

LDPM	LCG Disk Pool Manager
LHC	Large Hadrons Collider
LFC	LCG File Catalogue
LFN	Logical File Name
LOA	Legion Object Address
LOID	Legion Object IDentifier
LSF	Load Sharing Facility
LRMS	Local Resource Management System
LRU	Least Recently Used
MDS	Monitoring and Discovery Service
MSS	Mass Storage System
NAS	Network Attached Storage
NDGF	Nordic Data Grid Facility
NFS	Network File System
OGF	Open Grid Forum
OGSA	Open Grid Service Architecture
OPR	Object Persistent Representation
OSG	Open Science Grid
PAT	Policy Administration Tool
PDP	Policy Decision Point
PEP	Policy Enforcement Point

PKI	Public Key Infrastructure
PR	Policy Repository
PVFS	Parallel Virtual File System
RB	Resource Broker
RFIO	Raw File I/O
RFTS	Reliable File Transfer Service
RGMA	Relational Grid Monitoring Architecture
RLS	Replica Location Service
RSL	Resource Specification Language
SAM	Storage Access Manager
SAN	Storage Area Network
SE	Storage Element
SOAP	Simple Object Access Protocol (original meaning)
SQL	Structured Query Language
SRB	Storage Resource Broker
SRM	Storage Resource Manager
SSE	Smart Storage Element
STORM	STOrage Resource Manager
SURL	Storage URL
SUT	System Under Test
TDR	Technical Design Report

TLS	Transport Layer Security
TSM	Tivoli Storage Manager
TURL	Transport URL
UI	User Interface
VDC	Virtual Data Catalog
VDL	Virtual Data Language
VDT	Virtual Data Toolkit
VOMS	Virtual Organization Management System
WLCG	Worldwide LHC Computing Grid
WMS	Workload Management System
WN	Worker Node
WSDL	Web Service Description Language
WSRF	Web Service Resource Framework

PREFACE

Grid Computing is one of the emerging research fields in Computer Science. The Grid aims at providing an infrastructure that enables the sharing, selection, and aggregation of geographically distributed "autonomous" resources dynamically depending on their availability, capability, performance, cost, and users' quality-of-service requirements. Users belonging to a "Virtual Organization" can establish policies of usage, requirements, a working environment and even a set of virtual resources for operation. From the time of the first proposal made by Ian Foster and Carl Kesselman with the publication of the book "The Grid: Blueprint for a new computing infrastructure" and the development of the Globus Toolkit, the Grid has gone through major evolutions, attracting industry partners as well. In Europe, the projects European DataGrid (EDG), Worldwide Large Hadrons Collider Computing Grid (WLCG) and Enabling Grid for E-SciencE (EGEE) have promoted the development of Grid middleware and the creation of a worldwide computing infrastructure available for science and research.

Even though current middleware solutions are much more complete than the first prototype proposed by the Globus Toolkit, there are many areas that still need investigations and development. One of these is certainly storage management and access. There are many research challenges: applications running on the Grid need to transparently access data on the specific local storage device, exploiting a set of needed features such as space and quota management, POSIX file access, security and policy enforcement, reliable file movement, without being aware of the specific hardware/software solutions implemented at a site.

At the time of writing a complete, self-contained and coherent solution to storage management is missing in many of the existing Grid research infrastructures today: WLCG in Europe, Nordic Data Grid Facility (NDGF) in the Northern European countries, Open Science Grid (OSG) in USA, etc. One of the issues that complicate the task is the heterogeneity of storage solutions used in computing centers around the world. This work aims at providing a proposal for v2.2 Storage Resource Manager (SRM) protocol, a Grid protocol for storage systems that provides for uniform storage management capabilities and flexible file access. In particular a formal model has been designed and used to check the consistency of the specification proposed. The test modeling approach has been used to generate a test suite to validate the implementations made available for the storage services deployed in the WLCG infrastructure. The study of the black box testing methodology applied to SRM has allowed us to find

many inconsistencies in the specifications and to deeply test the behavior of the SRM systems. In order to converge toward a first real implementation of SRM in version 2 we left uncovered issues and features that will be dealt with in version 3 of the SRM protocol. The first deployment in production of SRM v2.2 based storage services is foreseen in June 2007. Among the solutions SRM v2.2 based there are CASTOR developed at CERN and Rutherford Appleton Laboratory (RAL), dCache developed at Deutsches Elektronen-Synchrotron (DESY) and Fermi National Accelerator Laboratory (FNAL), LDPM developed at CERN, BeStMan developed at LBNL and StoRM developed in Italy by Istituto Nazionale di Fisica Nucleare (INFN) and International Centre of Theoretical Physics (ICTP). StoRM is a disk-based storage resource manager designed to work over native parallel filesystems. It provides for space reservation capabilities and uses native high performing POSIX I/O calls for file access. StoRM takes advantage of special features provided by the underlying filesystem like ACL support and file system block pre-allocation. Permission management functions have also been implemented. They are based on the Virtual Organization Management System (VOMS) and on the Grid Policy Box Service (G-PBox). StoRM caters for the interests of the economics and finance sectors since security is an important driving requirement.