Management and Analysis of Large Research Data Sets

Rob Allan
Computational Science and Engineering Department,
STFC Daresbury Laboratory, Daresbury, Warrington WA4 4AD

Contact e-Mail: r.j.allan@dl.ac.uk

December 2, 2011

Abstract

This report looks at a wide range of issues and solutions relating to data management and analysis in the context of high performance computing for scientific simulation and modelling. It considers policies, data set formats, cluster and wide area distributed or hierarchical and high performance storage systems. To set the scene it also considers Research Council requirements to publish and curate research data and outcomes.

This report does not address storage solutions, backup or checkpointing which are nowadays considered to be part of the infrastructure provision. It also does not address the implementation of archival, curation and discovery technologies for which we refer the reader to work of the UK e-Science Programme and in particular the Digital Curation Centre.

Contents

1 Introduction

Managing scientific data has been identified as one of the most important emerging needs of the scientific community because of the sheer volume and increasing complexity of data being created or collected. This is particularly true in the growing field of computational science where increases in computer performance permit ever more realistic simulations and the potential to automatically explore large parameter spaces, e.g. using tools based on workflow. Bell *et al.* [6] noted that *As simulations and experiments yield ever more data, a fourth paradigm is emerging, consisting of the techniques and technologies needed to perform data intensive science. ... The demands of data intensive science represent a challenge for diverse scientific communities.*

Effectively generating, managing and analysing the data and resulting information requires a comprehensive, end-to-end approach that encompasses all the stages from the initial data acquisition to its final analysis. This is sometimes referred to as Information Lifecycle Management or ILM. For a discussion of the research activity lifecycle in the context of data management see [36].

A SciDAC project [25] has identified three significant requirements based on community input. Firstly, access to more efficient storage systems – in particular, parallel file system improvements are needed to write and read large volumes of data without slowing a simulation, analysis, or visualisation engine. Secondly, scientists require technologies to facilitate better understanding of their data, in particular the ability to perform complex data analysis and searches over large data sets in an effective way – specialised feature discovery, parallel statistical analysis and efficient indexing are needed before the data can be understood or visualised. Finally, generating the data, collecting and storing the results, data post-processing, and analysis of results is a tedious, fragmented process – workflow tools are required for automation of this process in a robust, tractable and recoverable fashion to enhance scientific exploration adding provenance and other metadata in the process. To this we could add stages of pre-processing which bring similar requirements, for instance mesh generation in engineering. We consider workflow tools in a separate report.

Here we review the requirements and tools for managing large data sets of interest to the UK research community involved in computational simulation and modelling. This in part extends work in a previous report [5] prepared for the UK High End Computing project `http://www.ukhec.ac.uk`.

We do not address archival, curation and discovery for which we refer the reader to work of the UK e-Science Programme and in particular the Digital Curation Centre. Nevertheless we first survey the expectations of public funding bodies in terms of publishing science outcomes and making related data available. This makes it essential not only to be able to manage and interpret large data sets at the time of the original research, but to create accurate metadata at the time of the original data creation [1] and ensure that data sharing and subsequent analysis is possible many years in the future.

A second report will focus on data intensive computing with requirements and examples of approaches to data analysis [2].

[1] Information which describes significant aspects of a resource. Most discussion tends to emphasise metadata for the purposes of resource discovery. Metadata are also required to manage and preserve digital materials over time and to assist in ensuring essential contextual, historical and technical information are preserved along with the digital object.

2 Data Centres, Curation, Publishing and Policies

A report [18] commissioned by JISC, the Joint Information Systems Committee and RIN, the Research Information Network, was published in Sep'2011 and surveyed the work of a number of data centres and services including: ADS, Archeology Data Service; BADC, British Atmospheric Data Centre; CDS, Chemical Database Service; EBI, European Bioinformatics Institute; ESDS, Economic and Social Data Service; NCDR, National Cancer Data Repository; NGDC, National Geo-science Data Centre; UKSSDC, UK Solar System Data Centre. These receive funding from the UK Research Councils plus CR-UK and the Wellcome Trust.

Rather than policies or technology, the report looked at the centres from a user perspective and was based on surveys by the Technolopis Group from Nov'2009-Jan'2010. It concluded that the re-use of curated data was high and it did indeed lead to improved research efficiency and quality with quantifiable impacts, followed by additional data deposits. Nearly all users were academic, with the exception of social data sets.

There are two main issues addressed briefly in the rest of this section: (1) open publication of research outcomes; and (2) the need for data retention (curation).

2.1 RCUK Position Statement

Policies arise from the seven core RCUK principles on data sharing, see `http://www.rcuk.ac.uk/research/Pages/DataPolicy.aspx`. Two of these principles are of particular importance: (1) that publicly funded research data should generally be made as widely and freely available as possible in a timely and responsible manner; and (2) the research process should not be damaged by inappropriate release of such data.

We note that these principles are themselves derived from a statement of the OECD (world wide Organisation for Economic Co-operation and Development) that publicly funded research data are a public good, produced in the public interest and, therefore, should be openly available to the maximum extent possible [34].

Policies must also reflect the principal UK legal provisions intended to assure access to publicly held information. These include: the Freedom of Information Act 2000 and the Freedom of Information (Scotland) Act 2002; Data Protection Act 1998; the Environmental Information Regulations 2004; and the Environmental Information (Scotland) Regulations 2004. These Acts allow any person to ask any public authority (including universities) for any information they believe to be held by that authority, and require the authority to respond in writing stating whether or not they hold the information sought and, if so, to supply that information unless certain exemptions apply.

Any exemptions, which may be absolute or qualified, generally relate to considerations such as national security, law enforcement, commercial interests or data protection, all of which may be relevant to research data. Guidance is available to help researchers and their institutional representatives understand their obligations. See for example the JISC publication [37]. Note: the exemptions in Scotland differ from those in the rest of the UK.

2.2 UK Digital Curation Centre

Digital curation is about maintaining and adding value to a trusted body of digital research data for current and future use. It includes the active management of data throughout the research lifecycle. To be useful the data should be validated and the method by which it was generated should be recorded.

In the UK, the Digital Curation Centre (DCC) has experts in curating digital research data who promote best practice in storing, managing and protecting digital data, see `http://www.dcc.ac.uk`. They explain the principles of curation to primary stake holders and to the wider community, seek to inform and influence political positioning in the curation and preservation landscape and promote and publicise curation concepts.

The DCC has created a number of information brochures and a *Curation Reference Manual* [30]. There is also a collection of papers on *Managing Research Data* [31].

Developing a data management plan is now a core part of good research practice and has been shown to bring significant benefits in terms of more efficiently conducted research and avoiding the risk of data loss. The starting point in developing such a plan should be to consult the DCC's useful overview of research funders' requirements as also summarised below.

The widely praised *DMP Online* is the DCC's data management planning tool. The tool draws upon the DCC's analysis of funders' data requirements to help project teams create up to three iterations of a data management plan: a "minimal" version for use at the grant application stage; a "core" version to be developed during the project itself; and towards the end of the project a "full" version that addresses issues of longer term access and preservation.

2.3 BBSRC

The BBSRC states that publications should be deposited at the earliest opportunity and expects data to be made available in a timely and responsible manner. Timely release could be considered as no later than the release of main findings through publication, or three years as a general guide. Data should be maintained for a minimum of ten years after project completion through their home institutions.

The BBSRC encourages data sharing in all research areas where there is strong scientific need and where it is cost effective. They encourage researchers to make material openly available, in suitably accessible formats using established standards. A publications repository and financial support for data sharing is available to facilitate sustained access. Researchers are therefore required to submit a data sharing plan with their proposals.

A number of databases are recommended for depositing research data, several of which are hosted at EBI, the European Bio-molecular Institute at Hinxton near Cambridge.

Further information:

`http://www.dcc.ac.uk/resources/policy-and-legal/research-funding-policies/bbsrc`

`http://www.bbsrc.ac.uk/publications/policy/access-research-outputs.aspx`

http://www.bbsrc.ac.uk/web/FILES/Policies/data-sharing-policy.pdf

http://www.bbsrc.ac.uk/publications/policy/good-scientific-practice.aspx

http://www.ebi.ac.uk

2.4 EPSRC

EPSRC has mandated open access publication of research that it funds since 2009. From 1/5/2011, they introduced a new policy framework covering access to, and management of, research data arising from research sponsored by EPSRC. This sets out their expectations arising from the core RCUK principles.

The framework is not prescriptive about how the expectations should be met and gives freedom for institutions to develop policies and practices based on individual circumstances. Specific expectations are published on-line [33] and include the following. Publications should include information on how and under what terms the related data can be accessed. Organisations must ensure that structured metadata exists describing the research data they hold made freely accessible on the internet. The metadata must be sufficient to allow others to understand what research data exists, why, when and how it was generated and how to access it. Where the research data referred to in the metadata is a digital object it is expected that the metadata will include use of a robust digital object identifier (DOI).

Organisations must ensure that research data is securely preserved for a minimum of 10 years from the date that any "privileged access" period expires or from the last date on which access to the data was requested by a third party. They must also ensure that effective data curation is provided throughout the full data life cycle as defined by the Digital Curation Centre.

EPSRC do not provide data centres. Instead, research organisations must ensure adequate resources are provided to support the curation of data arising from publicly funded research. These resources must be allocated from within their existing public funding streams, whether received from RCs as direct or indirect support for specific projects or from HEFCE as block grants.

Further information:

http://www.dcc.ac.uk/resources/policy-and-legal/research-funding-policies/epsrc

http://www.epsrc.ac.uk/about/infoaccess/Pages/roaccess.aspx

http://www.epsrc.ac.uk/funding/managing/Documents/goodpracticeguide.pdf

http://www.epsrc.ac.uk/about/standards/researchdata/Pages/default.aspx

http://www.epsrc.ac.uk/about/standards/researchdata/Pages/expectations.aspx

2.5 ESRC

ESRC require applicants to consider what outputs will be created at the proposal stage and how these will be made available in the long term. Researchers are expected to make all outputs accessible as soon as possible. ESRC provide a publications repository and data service to facilitate this. Grant holders are then expected to deposit publications at the earliest opportunity and data must be offered to the Economic and Social Data Service, ESDS, based at the UK Data Archive in Colchester within three months of the end of the award. Planning to do this is part of the grant application process. ESRC's research data policy was last updated in Sep'2010.

Further information:

`http://www.dcc.ac.uk/resources/policy-and-legal/research-funding-policies/esrc`

`http://www.esrcsocietytoday.ac.uk/ESRCInfoCentre/Support/access/`

`http://www.esrc.ac.uk/about-esrc/information/data-policy.aspx`

`http://www.esrc.ac.uk/_images/Research_Data_Policy_2010_tcm8-4595.pdf`

2.6 NERC

NERC published a position statement on access to research outputs in 2006. To support access to environmental data, their data policy also requires that award holders offer a copy of any data set resulting from NERC funded activities to its data centres. A new version of the data policy was published in Jan'2011 with data management requirements expected to be implemented in 2012.

Long term curation is central to NERC and an extensive data centre support infrastructure is in place to facilitate this. Use of these centres is free to NERC funded researchers who are expected to consider aspects of data creation and management prior to beginning research. Over arching data plans are produced for each thematic programme. BADC's *Data Management Plans* template is indicative. Grant applicants must include a plan for their work drawn up and implemented with an appropriate data centre.

The current NERC data centres are as follows.

- Atmospheric science – BADC, the British Atmospheric Data Centre;
- Earth sciences – NGDC, the National Geo-science Data Centre;
- Earth observation – NEODC, the NERC Earth Observation Data Centre;
- Marine Science –BODC, the British Oceanographic Data Centre;
- Polar Science – PDC, the Polar Data Centre;
- Science based archaeology – NERC users are encouraged to use the Archeology Data Service which is part of the Arts and Humanities Data Service, AHDS;

- Terrestrial and freshwater science, hydrology and bio-informatics – EIDC, the Environmental Information Data Centre is based at CEH, Wallingford and comprises the National Water Archive, the Biological Records Centre and the UK Environmental Change Network. It also includes NEBC, the Environmental Bio-informatics Centre.

Information on all data held within the centres will be made available through the NERC Data Discovery Service which provides an integrated, searchable catalogue.

NERC also support an e-Prints document repository. Publications should be made accessible through this or other institutional repositories. Publications resulting from NERC funding must be deposited at the earliest opportunity and data must be offered after a "reasonable period" of exclusive use, currently considered to be two years from the end of data collection.

Further information:

`http://www.dcc.ac.uk/resources/policy-and-legal/research-funding-policies/nerc`

`http://www.nerc.ac.uk/about/access/statement.asp`

`http://www.nerc.ac.uk/research/sites/data/`

`http://www.nerc.ac.uk/research/sites/data/policy.asp`

2.7 STFC

As an example, the full policy statement from STFC (as published Sep'2011) is re-produced in Appendix B.

Researchers are expected to make publications that arise from STFC funded research available at the earliest opportunity. An e-Pubs system has been set up for this purpose. Activities in the e-Science Centre have led to STFC's statement on data management or sharing and best practice, but there is currently no overall formal policy covering long term curation. It is suggested that a domain specific or institutional repository be used and that data should be retained for a minimum of 10 years. Data archives are being implemented for facilities such as Diamond and ISIS, see Secion 4.5. There is also separate provision for access and management of particle physics data through the GridPP consortium and UK Tier-1 centre.

Note that STFC, at the Rutherford Appleton Laboratory, host the BADC, the particle physics Tier-1 Centre and a data archive for BBSRC. STFC also formerly hosted the HPCx service at Daresbury Laboratory.

Further information:

`http://www.dcc.ac.uk/resources/policy-and-legal/research-funding-policies/stfc`

`http://www.scitech.ac.uk/rgh/rghDisplay2.aspx?m=s&s=64`

`http://www.stfc.ac.uk/stfcconsultation/sources/strategy/StrategyConsultationDocument.`

pdf

2.8 JISC

The JISC, the Joint Information Systems Committee, funded a programme on Managing Research Data from 2009-11, see http://www.jisc.ac.uk/whatwedo/programmes/mrd/outputs.aspx.

The Web page provides a narrative guide to outputs from the programme and some related JISC funded activities. This contains links to the projects and services mentioned below. It is intended as an easy point of introduction to key outputs that will be of interest to others seeking to improve research data management in universities, and therefore relevant to the discussion above. It is intended that this will be useful for institutions seeking to improve research data management.

This is an ongoing activity, with JISC funding for further projects in this area announced from time to time.

Research data management support for researchers

The *Incremental* project has produced Web pages providing support and guidance for managing research data: Support for Managing Research Data at the University of Cambridge; and Data Managing Support for Researchers at the University of Glasgow.

The *EIDCSR* project, sister project to *SUDAMIH*, created a similar Research Data Management site for the University of Oxford. Likewise, University of Edinburgh Information Services has put together a site providing Research Data Management Guidance.

Introductions and "How To" guides

The UK Data Archive has recently revised its guide *Managing and Sharing Data – Best Practice Guide for Researchers*, in part as a result of work undertaken by the JISC funded project *Data Management Planning for ESRC Research Data Rich Investments* (DMP-ESRC).

Building on its wide ranging set of briefing papers, the Digital Curation Centre is also producing a series of "How To" guides which provide a working knowledge of curation topics, aimed at people in research or support posts who are new to curation, but are taking on responsibilities for managing data, whether at local research group level or in an institutional data centre or repository. The first two guides in this series deal with how to appraise and select research data and how to license research data.

In its early stages, the *ERIM* project produced a *Review of the State of the Art of the Digital Curation of Research Data* which serves as an introduction to and overview of the issues.

Model data management plans and guidance

The DMP-ESRC project produced a substantial and detailed set of *Data Management Recommendations for Research Centres and Programmes* as well as a summary guide to two key recommendations, relating to research data management strategies and the maintenance of a resources library. Although targeted at large ESRC research investments, these guidelines are widely applicable and could be

useful for data management planning in other disciplines.

The *ERIM* project, examining research data management and sharing issues for researchers at the University of Bath's *Innovative Design and Manufacturing Research Centre* produced a *Draft Data Management Plan for IdMRC Projects*. The work builds on a set of high level *Principles for Engineering Research Data Management*; a *Thematic Analysis of Data Management Plan Tools and Exemplars*; and a *Requirement Specification for an Engineering Research Data Management Plan*.

Case studies and requirements analyses

A number of the projects in the programme produced requirements analyses. Some projects took a broad institutional view while others focused on the requirements of specific disciplines.

Institutional Data Management Blueprint examined research data management challenges across a number of departments at the University of Southampton to produce a report.

Incremental carried out a scoping study and produced an implementation plan. It took a similarly broad approach covering researchers' requirements in a range of departments at both Cambridge and Glasgow.

SUDAMIH produced a requirements report for a research data management infrastructure to support humanities researchers.

I2S2 performed in depth case studies of practice in various forms of structural science to produce its main requirements report and a supplementary report.

ERIM conducted a detailed and methodologically rigorous study *Understanding and Characterizing Engineering Research Data for its Better Management* and also examined opportunities for and barriers to engineering research data re-use.

HALOGEN at the University of Leicester, was a pilot project to demonstrate how a central IT services department could provide support for specific research projects, an approach which potentially can bring cost efficiencies and promote collaboration across all departments and disciplines. The project developed a sustainable and potentially extensible platform integrating archeology datasets. They produced a service specification, a technical design specification and a data glossary detailing the datasets to be integrated in the system.

Gravitational Waves produced an interesting draft report on research data management in "big science", taking the LIGO Gravitational Wave Astronomy collaboration as a case study.

MaDAM was a project at University of Manchester undertaken by John Rylands University Library, Research Computing Services and Manchester e-Research Centre and aimed at developing a pilot data management infrastructure for bio-medical researchers. The project ended in Mar'2011 and was organised in five stages: requirements capture; implementation of a demonstration technical architecture and supporting data management service; evaluation; sustainability analysis; and dissemination. Experience from the project has fed into a green paper [11]. To benefit fully from this project, it has been recommended that it be continued as the first element of the universitys research data management service.

Research data management platforms

Many projects in the Managing Research Data programme developed technical platforms and software to help researchers manage their data.

As an example, the core technical output of the *I2S2* project was to develop the I2S2 Information Model and to implement this within the STFC's ICAT Lite "personal workbench for managing data flows". This allows the user to manage data, to capture provenance information and to "commit data" for long term storage. The project has produced a useful implementation plan and a description of their pilot implementation. ICAT was formerly a product of the STFC e-Science Centre at Daresbury.

Research data management costing

Understanding how to model the full cost of research data management is a challenging area and one which will require further work at the institutional level. Material for understanding activity based costing has come out of the *Keeping Research Data Safe* project and a good starting point is the project's user guide.

The DMP-ESRC project produced a light weight activity based research data management costing tool for researchers in the social sciences.

Training materials

A number of projects in the JISC programme produced training materials which are available for re-use and adaptation. As an exmaple, working with the Humanities Division at the University of Oxford, the SUDAMIH project produced training materials for humanities researchers.

2.9 NHS

Management of research data produced in collaboration with or derived from the NHS falls under the *Research Governance Framework for Health and Social Care* [35]. This requires an organisation to have clearly documented standard operating procedures for the management of all research data. As an example, the University of Manchester ensures compliance with the framework through a Research Governance MoU with partner NHS Trusts, and a joint Research Governance Group meets regularly [11]. The framework states that *data collected in the course of research must be retained for an appropriate period, to allow further analysis by the original or other research teams subject to consent, and to support monitoring by regulatory and other authorities.*

3 Data Management

Scientists often consider "data management" to mean a physical data store with an access layer for movement of data from one location to another. The scope of scientific data management is however much broader, encompassing both its meaning and content.

The cutting edge of computational science involves very large simulations taking many hours or days on the latest high performance (and therefore expensive) computers. For business data, large companies implement enterprise wide data architectures, with data warehousing and data mining to extract information from their data. Can something similar be done for scientific data?

Some problems identified from current scientific projects include the following.

- Limited file and directory naming schemes. Many project data repositories are simply big flat directory structures. This makes it hard to catalogue, find and re-use data;
- No access to important metadata as information tends to be stored in scientists' notebooks and heads. Without preserved metadata, relevance of data and information extracted from it can be lost;
- Scientists retrieve entire files to ascertain relevance of their content. It is hard to pick out individual pieces of data, for instance a parameter as a function of a variable sweeping across multiple simulations. This is often because the metadata is missing;
- "Un-owned" data with dubious content after the end of project or Ph.D. thesis. This so called "grey literature" is often un-usable.

The increasing size of scientific data collections brings not only problems, but also opportunities. One of the biggest is the possibility to re-use existing data for new studies. This was one of the great hopes of the e-Science programme. Many projects investigated data curation, provenance and metadata definitions based on common ontologies. At STFC it was a goal of the Facilities e-Infrastructure Programme with a focus on SRB, ICAT and DataPortal for SRS, Diamond and ISIS.

Scientific data is composed not only of bytes, but also of workflow definitions, computation parameters, environment setup and provenance. Capturing and using all this associated information is a goal of, among others, the JISC funded myExperiment project which focusses on workflows encapsulated in "research objects", see `http://www.myexperiment.org`.

Other aspects of data management, particularly virtualisation of the underlying storage, has been tackled in projects such as SRB, the Storage Resource Broker from SDSC, now superceded by iRODS.

Much of the work mentioned above was aimed a creating catalogues such as ICAT which reference large data collections, see below. A similar project for high energy physics data is LFC, the LCG File Catalogue. Entries in the catalogues may point to the outputs of a facility or long term research programme which will have consumed extensive public funding and are deemed to be of lasting and sometimes national importance.

We will not attempt to fully describe or re-produce this e-Science work here, the background to which is discussed in a report [14]. We also do not consider issues of data curation for which appropriate standards and processes have been developed and documented by the Digital Curation Centre as noted above. Instead, we will now focus on ways to manipulate and analyse large scientific data sets. There are lists of open data repositories on-line, for instance `http://oad.simmons.edu/oadwiki/Data_repositories`. A few examples are as follows.

Astrophysics: Virtual Observatories, e.g. SLOAN Sky Survey, data from LSST (Large Synoptic Survey Telescope) 16TB per 8 hours;

Biology, Bio-chemistry and Bio-physics: EBI, the European Bio-informatics Institute;

CFD and Computational Engineering: CAD models, flow fields, etc.

Environment and Atmosphere: BADC, the British Atmospheric Data Centre;

Facilities Data: ICAT and associated tools developed at STFC;

Geo-physics: e.g. FAGS: Federation of Astronomical and Geo-physical Data Analysis Services, `http://www.icsu-fags.org/`;

High energy physics: Analysis of data from CERN LHC, 1.6GB/s while operating, e.g. UK Tier-1 Centre at RAL;

Meteorology: e.g. Met Office examination of longitudinal data sets for climate trends;

Oceans: BODC (British Oceanographic Data Centre) and NOC (National Oceaonography Centre);

Protein Data Bank: 3D biological macro-molecular structure data widely used by projects such as CCP4, PDB hosted at the SBI;

Social and Geo-spatial: e.g. ESDS (Economic and Social Science Data Service), EDINA, CESSDA (Council of European Social Science Data Archives), digital multi-media libraries and cinema.

The LHC data analysis represents an extreme case as it will generate upwards of 14PByte of data a year, which has to be distributed across the EGEE, OSG and NorduGrid Grids for analysis. Such data volumes cannot be handled easily with current production networks, so have required the provisioning of optical private networks (an unusual form of SAN) linking CERN's Tier-0 centre to the key national Tier-1 computing centres around the world, the one for the UK being situated at RAL. Dedicated 10Gb/s network links are provided in this way for data movement.

We do not consider the middleware aspects of such data grids in this report, but focus more on the requirements of a high end data centre focusing on computational simulation and modelling. The following figure illustrates the architecture implemented in the SciDAC Data Management Center [25].

Here activities are organised in three layers that abstract the end-to-end data flow. The layers are: Storage Efficient Access (SEA); Data Mining and Analytics (DMA); and Scientific Process Automation (SPA). The SEA layer is immediately on top of the infrastructure, i.e. data intensive computing hardware, operating systems, file systems and hierarchical mass storage systems, and provides parallel data access technology and transparent access to archival storage. The DMA layer, which builds on the functionality of the SEA layer, consists of indexing, feature selection and parallel statistical analysis technology. The SPA layer, above the DMA layer, provides the ability to compose scientific workflows from the components in the DMA layer as well as application specific modules. This architecture has been used in the centre to organise its components and apply them to various scientific applications.

Important aspects in any data management system have been identified as follows.

- The most important thing is the metadata schema definitions (ICAT provides an example);
- Data ingestion process (metadata collection and organisation);
- Physical data access;
- User data access interface;

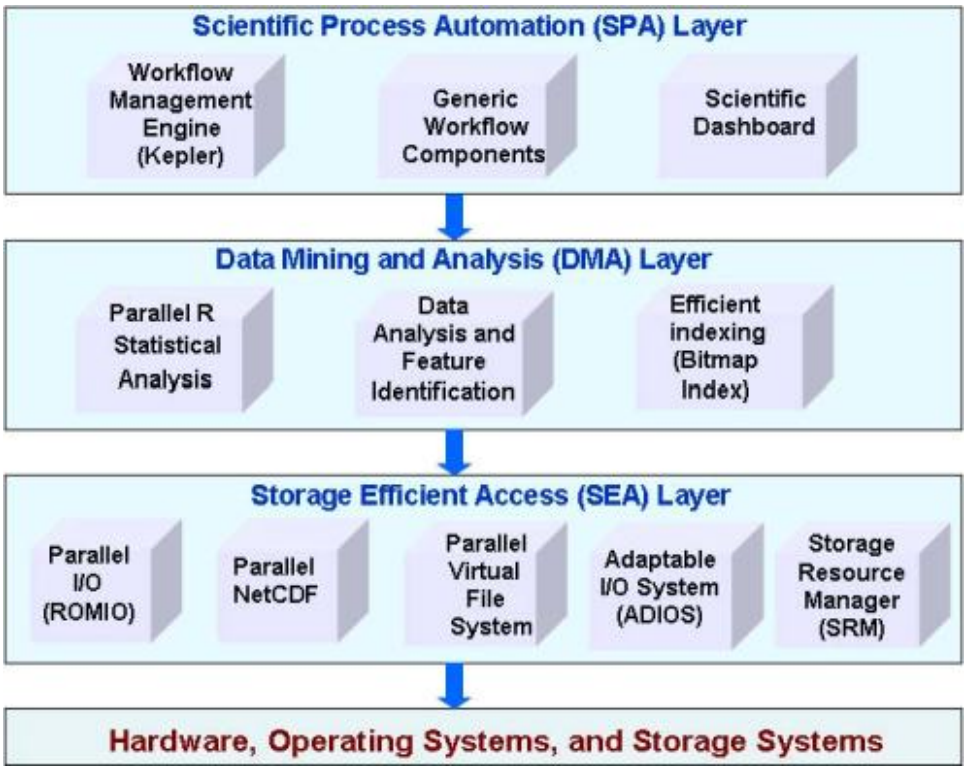

Figure 1: Architecture of SciDAC Data Management Center

- Metadata storage and management;
- Workflow definition and management;
- Rules for ownership and data lifetime definition;
- Data quality assessment process.

3.1 Data Formats

To be accessible, scientific data must be stored in a widely recognised format. Not surprisingly, many data format standards are concerned with multi-dimensional data or image processing. In some cases these are domain specific, but others are more generic.

Data management issues and processes are mostly independent of the data format. Nevertheless some formats are more suited to support metadata and more helpful in reaching the various data management goals. The format can also have a significant effect on i/o performance and the ability to search across and pull out sub-sets of data. An example of this is slicing through 3D data. For system independence, e.g. transferring data between big endian and little endian systems, formats such as XDR (eXternal Data Representation) can be used, although translation may be slow.

There have been several attempts to use XML to describe binary data formats, but these have been largely un-successful. Initiatives such as DFDL, Data Format Description Language and BINX attempt

to do this. Traditional formats like HDF-5, NetCDF and CGNS are widely used and newer formats using XML may be more suited for storing metadata. It is also possible to use relational or object databases for certain types of data. We note that usually the metadata is stored separately from the actual data, for instance in a catalogue (searchable database) which contains location references or URLs. There are issues of transaction management and consistency when data or metadata are distributed or replicated.

We do not address low level file systems such as FAT, EXT3, HPFS, etc. in this document, although we note that some performance improvements can be associated with an appropriate choice. This is particularly true of newer ones which are being developed to address scalability to large networked file stores, such as Oracle's BTRFS, B-Tree File System (fault tolerant) introduced in 2007, see `http://btrfs.wiki.kernel.org` and CRFS, the Coherent Remote File System. ZFS from Sun offers some similiar capabilities and is available on the newer NW-GRID clusters. This includes volume management functions, scalability, snapshots and copy-on-write clones plus built in integrity checking and repair, RAID and NFS-4 support. See Wikipedia for more information about file systems, `http://en.wikipedia.org/wiki/List_of_file_systems`.

The following list describes a number of widely used data set formats expanding on that in [5]. Section 3.2 goes on to describe storage and distributed high performance file systems which may be of interest.

ADF: Advanced Data Format used for CFD data. HDF-5 is now widely used instead because it offers parallel i/o and data compression support.

CDF: the Common Data Format is a library and toolkit developed by NASA. The software is an interface for the storage and manipulation of multi-dimensional data sets.

CGNS: the CFD General Notation System consists of a collection of conventions and software for the storage and retrieval of CFD data, see `http://cgns.sourceforge.net/index.html`.

The CGNS system is designed to facilitate the exchange of data between sites and applications and to help stabilise the archiving of aerodynamic data. The data are stored in a compact, binary format and are accessible through a comprehensive and extensible library of functions. The API is platform independent and can be easily implemented in C, C++ and Fortran. A data viewer is available.

CIF: the IUCr Crystallographic Information File is becoming standard for crystallography and related fields, see `http://www.iucr.org/resources/cif`. ICAT for example uses imgCIF for crystallographic binary image data. There is a related mmCIF for macro-molecular structures.

DEM: a Digital Elevation Model consists of a sampled array of elevations for ground positions that are normally at regularly spaced intervals. Information about this format, along with data availability, is available from USGS, the US Geological Survey. Note there are several related DEM file formats.

DLG-3: the Digital Line Graph format is used for cartographiy by USGS to store geographical vector data as part of a geographical information system.

FITS: the Flexible Image Transport System is a digital file format used to store, transmit and manipulate scientific and other images. It is the most commonly used digital file format in astronomy.

Unlike many image formats, FITS is designed specifically for scientific data and hence includes many provisions for describing photometric and spatial calibration information, together with image origin metadata. See `http://fits.gsfc.nasa.gov`.

A major feature of the FITS format is that image metadata is stored in a human readable ASCII header, so that an interested user can examine the headers to investigate a file of unknown provenance.

FITS is also often used to store non-image data, such as spectra, photon lists, data cubes, or even structured data such as multi-table databases. A FITS file may contain several extensions, and each of these may contain a data object. For example, it is possible to store X-ray and infrared exposures in the same file.

GRIB-1 and GRIB-2: GRid In Binary is a concise data format commonly used in meteorology to store historical and forecast weather data. It is standardised by the World Meteorological Organisation's Commission for Basic Systems (GRIB FM 92-IX). GRIB-1 is still used operationally world wide by many meteorological centres for numerical weather prediction output. Since the introduction of GRIB-2, data is slowly changing over to the new format. GRIB-2 is for instance used for derived forecasts distributed in Eumetcast of Meteosat Second Generation. Another example is NAM, the North American Mesoscale model. See `http://www.wmo.ch/pages/prog/www/DPS/FM92-GRIB2-11-2003.pdf`.

HDF-5: the Hierarchical Data Format, is a general purpose library and file format for storing scientific data. It is a self defining file format for transfer of various types of data between different machines. The HDF library contains interfaces for storing and retrieving compressed or uncompressed raster images with palettes and an interface for storing and retrieving N-dimensional scientific datasets together with information about the data, such as labels, units, formats and scales for all dimensions. HDF-5 can store two primary objects: data set and group. A data set is essentially a multi-dimensional array of homogeneous data elements and a group is a structure for organising objects (data sets or other groups) in an HDF-5 file. Using these two basic objects, one can create and store almost any kind of scientific data structure, such as images, arrays of vectors and structured and un-structured meshes. Data is accessed using a Posix style path notation. A data viewer is available.

HDF was originally from NCSA and is now supported by the HDF Group, see `http://www.hdfgroup.org/HDF5`. We note that HDF-4 still exists, but is significantly different both in design and API.

Note: Q5cost is an HDF-5 based format which has a Fortran API developed in an EU COST D23 project for computational chemistry [4], see `http://abigrid.cineca.it/abigrid/the-docs-archive/q5cost/`.

NetCDF: the Network Common Data Form is a set of software libraries and self describing, machine independent data formats that support the creation, access and sharing of array oriented scientific data. NetCDF implements a machine independent, self describing, extensible file. The project Web site is hosted by Unidata at UCAR, the University Corporation for Atmospheric Research. They are also the chief source of netCDF software, standards development, updates, etc. NetCDF is an open standard and is widely used for climate modelling and related studies.

The project is actively supported. The recently released (2008) version 4.0 greatly enhances the data model by allowing the use of the HDF-5 data file format. See `http://www.unidata.ucar.edu/software/netcdf`.

NeXuS: is a common data format for neutron, X-ray, and muon science. It is being developed as an international standard by scientists and programmers representing major scientific facilities in Europe, Asia, Australia and USA in order to facilitate greater co-operation in the analysis and visualisation of neutron, X-ray and muon data. See `http://www.nexusformat.org`.

Like HDF-5, NeXuS is a hierarchical format with a directory style structure. Some metadata in NeXuS files is encoded as XML with standard tag names making them easy to interpret. NeXuS is used in ICAT.

OpenMath: aims at developing a standard exchange format for mathematical objects such as formulae processed by computer algebra systems. See `http://www.openmath.org`.

PDS: the Planetary Data System is an archive which has been responsible for storing space mission data on CD-ROM media, using its own self describing data format, variously know as PDS or ODL, Object Description Language. At least some of the current projects (e.g. Magellan, Galileo) are using the PDS format as a "pointer" to detached VICAR image format on the mission CD-ROM volumes. See `http://pds.nasa.gov`.

SAIF: Spatial Archive and Interchange Format is a Canadian standard for the exchange of geographic data. It uses an object oriented data model and consists of definitions of the underlying building blocks, including tuples, sets, lists, enumerations, and primitives. A company, Safe Software, was formed to provide tools and training for the SAIF data standard.

SDTS: the Spatial Data Transfer Standard is US Federal Information Processing Standard (FIPS) 173 for transfer of geological and other spatial data. Documentation and examples are available from the USGS. There are SDTS versions of DEM and CLG.

VICAR: Video Image Communication and Retrieval is a collection of image processing programs supported by the Multi-Mission Image Processing Laboratory (MIPL) at the US Jet Propulsion Laboratory (JPL), for use in manipulating and analysing images from spacecraft. The image format used by VICAR programs and for all or most data from JPL managed missions, is referred to as VICAR format. An independent third party description of the VICAR image format is available.

Miscellaneous graphics formats: include formats for storing graphics files – TIFF, GIF, JPEG, FLI, CGM, MPEG, etc.

There are many other formats, some proprietary or application specific, see also Section **??**.

3.2 Data Transfer Tools and Wide Area File Systems

Distributed File Systems are sometimes called Distributed Datastore Networks – see Wikipedia. In this report we consider only those which work on a wide area network and are thefore suitable for Campus or inter-site computing and data management. Normally many implementations have been made, they are location dependent and they have access control lists (ACLs), unless otherwise stated below.

We separate the rest of this section into server centric storage systems and those supporting distributed file servers. Most systems do however have some dependency on one or more central services, such as

metadata services, database or catalogues, which are noted. We assume that all systems reviewed can access distributed storage or provide storage to distributed clients in some way.

3.2.1 Keywords and Definitions

Keywords include: data migration; hierarchical storage management; information lifecycle management; storage area network; tiered storage.

Data Migration

Data migration is the transferring of data between storage types, formats, or computer systems. Data migration is usually performed programmatically to achieve an automated migration. It is required when organisations or individuals change computer systems or upgrade to new systems. Migration is a key issue in data curation, see the Digital Curation Centre `http://www.dcc.ac.uk`. Migration is thus a means of overcoming technological obsolescence by transferring digital resources from one hardware or software generation to the next. The purpose of migration is to preserve the intellectual content of digital objects and to retain the ability for clients to retrieve, display and otherwise use them with evolving technology. Migration differs from the refreshing of storage media in that it is not always possible to make an exact digital copy or replicate original features and appearance and still maintain the compatibility of the resource with the new generation of technology.

ILM

Information Lifecycle Management (ILM) is a comprehensive approach to managing the flow of information, data and metadata from creation to obsolescence. ILM encapsulates potentially complex criteria for storage management going beyond age of data and access frequency.

ILM thus refers to a wide ranging set of strategies for administering storage systems on computing devices. Specifically, four categories of storage strategies may be considered under the auspices of ILM. These concern: policies including SLAs around data management; operational aspects including backup and data protection; logical and physical infrastructure; and definition of how the strategies are applied.

ILM products automate the processes involved, typically organising data into separate tiers (see below) according to specified policies, and automating data migration from one tier to another based on those criteria. As a rule, newer data, and data that must be accessed more frequently, is stored on faster, but more expensive storage media, while less critical data is stored on cheaper, but slower media. ILM can specify different policies for data that declines in value at different rates or that retains its value throughout its life span.

SAN

A Storage Area Network (SAN) is a high speed network designed to attach computer storage devices such as disk array controllers and tape libraries to servers. SANs became widely used in enterprise (campus) storage from around 2006.

A SAN allows a machine to connect to remote targets such as disks and tape drives on a network usually for block level i/o. From the point of view of the class drivers and application software, the

devices appear as locally attached devices.

There are two variations of SAN:

1. A network whose primary purpose is the transfer of data between computer systems and storage elements. This SAN consists of a communication infrastructure, which provides physical connections, and a management layer, which organises the connections, storage elements and computer systems so that data transfer is secure and robust.
2. A storage system consisting of storage elements, storage devices, computer systems, and/ or appliances, plus all control software, communicating over a network.

Tiered Storage

Tiered storage is a data storage environment consisting of two or more kinds of storage differentiated by at least one of four attributes: Price; Performance; Capacity; Function. Any significant difference in one or more of the four defining attributes can be sufficient to justify a separate storage tier.

Examples include the following.

- Disk and Tape – two separate storage tiers identified by differences in all four defining attributes.
- Old technology disk and new technology disk – two separate storage tiers identified by differences in one or more of the attributes.
- High performing disk storage and less expensive, slower disk of the same capacity and function – two tiers with different access capabilities.
- Identical enterprise class disk configured to utilise different functions such as RAID level or replication – a separate tier for each set of unique functions.

HSM

Hierarchical Storage Management is related to tiered storage. It is a data storage technique that automatically migrates data between high cost and low cost (and probably higher capacity) storage media. HSM systems exist because high speed storage devices, such as hard disk drives, are typically more expensive (per byte stored) than slower devices, such as optical discs and magnetic tape drives. Whilst it would be ideal to have all data available on high speed devices all the time, this would be prohibitively expensive for most installations. HSM systems instead store the bulk of the organisation's data on slower devices and copy data to faster disk drives only when needed. In effect, HSM turns the fast disk drives into caches for the slower mass storage. The HSM system monitors the way data is used and makes best guesses as to which data can safely be relegated to slower devices and which data should stay on the fast disks.

HSM thus implements policy based management of file backup and archiving in a way that uses storage devices economically and without the user needing to be aware of when files are being retrieved from backup storage media. Although HSM can be implemented on a standalone system, it is more frequently used in the distributed network of an enterprise. Using an HSM product, an administrator can

establish and state guidelines for how often different kinds of files are to be copied to a backup storage device. Once the guideline has been set up, the HSM software manages everything automatically.

File Migration

Assuming a file based storage system, efficient file migration services are at the heart of HSM. It is also relevant when moving data from one vendor's product to another, but see under Data Migration above.

File migration thus arises from an ILM strategy that relegates data to less expensive devices as the data decreases in value to the enterprise. Migration may also be driven by a need to simplify or standardise environments, to improve storage space utilisation, to balance workloads between file systems, or to consolidate storage management.

File migration services are present in many commercial products, e.g. from CommVault, HP, LSI and Symantec.

3.3 Server Centric Storage Systems

This section lists some "traditional" storage systems aimed principally at backup, restore and archival, but now increasingly including business logic. These are typically aimed at managing (e.g. by indexing) many small files to produce a searchable archive store sometimes known as a collection.

CASTOR: the CERN Advanced Storage Manager is used as an interface to storage systems for high energy physics, including the Atlas Data Store at RAL. This is a HSM system in which files can be stored, listed, retrieved and accessed using command line tools or applications built on top of the different data transfer protocols like RFIO (Remote File IO), ROOT libraries, GridFTP and XROOTD. CASTOR manages disk cache(s) and the data on tertiary storage or tapes. CASTOR provides a Posix like directory structure with a single name space per site, but the CASTOR CLI must be used. All files are staged to allow for retrieval from tape, etc. on demand. Metadata is contained in a central database. See `http://castor.web.cern.ch`. This is also discussed by Stewart *al.* [21] in the context of EGEE storage management and Brown *et al.* [**?**] in the context of STFC experimental facilities .

CommVault: Simpana-9 HSM product has modules for backup, archive, replication, de-duplication, resource management and search built on a common software platform. Modules can be individually licensed, see `http://www.commvault.com/simpana.html`.

HDS: Hitachi Data Systems offer a range of products for HSM tiered storage management and virtualisation. See `http://www.hds.com/solutions/infrastructure`.

HP: HP Neoview data warehousing is aimed at the commercial sector and includes data analysis and customer relationship management. It is typically delivered alongside HP storage solutions and other business intelligence products. See `http://h71028.www7.hp.com/enterprise/w1/en/software/business-intelligence-neoview.html`.

IBRIX Fusion: from HP is based on a patented segmented file system architecture that, unlike other file systems, does not require a central metadata server or a distributed lock manager.

Good speed up and scalability is achieved by "parallelising" the data as well as the metadata. Available for Linux under a proprietary software license.

ICAT: from STFC is a metadata catalogue which provides a web service end point for the registration and retrieval of metadata. There are a number of clients available for registration, searching and retrieval of data. ICAT provides the clients with a single point of service. The deployment of ICAT allows for the distribution of the storage of the metadata. A typical usage scenario is the following: a data producer uses a tool to register the existence of data in ICAT and stores information such as the data location for retrieval; at the same time, additional information is stored so that searches can locate the data. In general, there are different tools for catalouging, searching and retrieval; however the tools share a common web service end point which defines the ICAT instance.

Isilon: clustered storage system architecture consists of independent nodes that are all integrated with the OneFS operating system software. The systems can be installed in standard data centre environments and are accessible to users and applications running Windows, UNIX or Linux and Mac operating systems using industry standard file sharing protocols over standard Gigabit ethernet. The OneFS operating system software is designed with file striping functionality across each node in a cluster, a fully distributed lock manager, caching, fully distributed metadata and a remote block manager to maintain global coherency and synchronisation across the cluster. See Wikipedia.

LSI: offers traditional high performance networking and storage for HPC systems, see `http://www.lsi.com/storage_home/high_performance_computing/index.html`.

StoreAge MultiMigrate product, now from LSI, see `http://www.lsi.com/storage_home/products_home/storage_virtualization_data_services/storeage_multimigrate`. This enables the on-line migration of data from any storage device to any other storage device, regardless of vendor. The migration takes place while the applications remain on-line without any interruption. It is aimed at migrating critical applications from older storage devices onto newer platforms.

ONStor: See `http://www.onstor.com/` for clustered NAS storage gateways. Among other things, it offers an SMB implementation that also supports NFS protocol so users can access the same data through both protocols, see Section 3.4. Note that ONStor is now part of LSI.

RelData: UnitedStorage is an IP storage gateway which consolidates and virtualises open storage resources, providing a storage "pool" over an existing IP network NAS or SAN infrastructure. RelData has an open back end connectivity enabling exsting storage to be re-used and also permits new fibre channel disk arrays to be added. There is no vendor tie in.

Symantec: VERITAS enterprise vault HSM product offers a range of storage and backup solutions aimed at Microsoft servers and typically used for archiving e-mail and Sharepoint files. It has tools for legal and compliance testing for commerce.

Tivoli: product line from IBM, see `http://www-03.ibm.com/systems/storage/software/`. Tivoli includes StorageManager-6 HSM for Microsoft Windows and Sharepoint. Virtualisation is available for storage consolidation. For high performance cluster and networked storage see GPFS in Section 3.5.5.

3.4 Systems with Distributed Services

High performance computing environments require parallel file systems and access to data from multiple clients. Traditional server based file systems such as those exported via NFS are unable to scale efficiently to support hundreds of nodes or multiple servers. Parallel file systems are typically deployed for dedicated high performance storage solutions within clusters, usually as part of a vendors integrated cluster solution. These file systems are often tightly integrated with a single clusters hardware and software environment making sharing them impractical. Recently, several parallel file systems have been introduced that are designed to make sharing a files between clusters feasible in the presence of hardware and software heterogeneity.

In this section we review distributed file systems, which are also possibly also parallel and fault tolerant, stripe and replicate data over multiple servers for high performance and maintain data integrity.

All file systems listed here focus on high availability, scalability and high performance unless otherwise stated. Whilst these provide distinct advantages over more traditional file systems they may be more or less complicated to install, configure and manage and may require specific Linux kernel patches.

AFS: Andrew File System is scalable and location independent, has a large client cache and uses Kerberos for authentication. AFS is a distributed networked file system which uses a set of trusted servers to present a homogeneous, file name space to all the clients. It was developed by Carnegie Mellon University as part of the Andrew Project and is named after Andrew Carnegie and Andrew Mellon. See Wikipedia.

Ceph: a scalable, distributed, open source file system from the Storage Systems Research Center, UC Santa Cruz. It is aimed at petabyte storage with replication and fault tolerance. Ceph has been included as "experimental" in Linux kernels since v2.6.34 in Mar'2010. Earlier versions used FUSE [2]. V0.33 is available for Linux under LGPL from SourceForge, see `http://ceph.newdream.net`.

Coda: is a fault tolerant distributed file system from Carnegie Mellon University which focuses on bandwidth adaptive operation, including disconnected operation using a client side cache for mobile computing. It is a descendant of AFS-2, see the AFS architecture section below. The client side caching can give this good performance for multiple read operations. It is available for Linux under GPL. See Wikipedia.

dCache: from FermiLab and DESY is part of the EGEE data management architecture and aims to provide a mechanism for storing and retrieving huge amounts of data shared among a large number of heterogeneous servers. It provides a single namespace view of all of the files that it manages and allows access to those files using a variety of a protocols. By connecting dCache to a tape back end, it becomes a hierarchical storage manager (HSM). See `http://www.dcache.org` and also Stewart *al.* [21].

DCE/ DFS: Distributed File System from IBM (earlier Transarc) is similar to AFS with a focus on full Posix file system semantics and high availability. Available for AIX and Solaris under a proprietary software license. See the AFS architecture section below.

[2] FUSE: File System in User Space, allows file systems to bridge to the Linux VFS kernel via libfuse and a kernel module thus allowing user space deployment. See `http://fuse.sourceforge.net`.

DFS: fault tolerant Distributed File System from Microsoft focuses on location transparency and redundancy for high availability. Based on SMB and available for Windows under a proprietary software license. See Wikipedia.

eXludus: High performance data management for optimisation within a cluster. This relies on multiple multi-cast routes on the internal cluster network and is implemented as a storage server and client on each node. Benchmarking has reported good scalability but overall performance somewhat slower than GPFS. Such solutions can improve application startup times. See separate reports [12, 16].

FraunhoferFS: from the Fraunhofer Society Competence Center for HPC. Available free of charge for Linux under a proprietary license. See `http://www.fhgfs.com`.

FraunhoferFS is written from scratch and incorporates results from previous experience. It is a fully Posix compliant, scalable file system including features as follows.

- Distributed metadata: Although parallel file systems usually distribute the file contents over multiple storage nodes, the metadata is often bound to single nodes. This leads to performance bottlenecks and limited fault tolerance. FhGFS distributes the metadata across all the available storage nodes in a way that keeps the lookup time at a minimum.
- Easy installation: FhGFS requires no kernel patches, is able to connect storage nodes and servers with zero configuration and allows you to add more clients and storage nodes to a running system.
- Support for high performance technologies: FhGFS is built on a scalable multi-threaded architecture with native InfiniBand support. Storage nodes can serve InfiniBand and ethernet clients at the same time and automatically switche to a redundant connection path in case of failure.

GAM: IBM's Grid Access Manager [17] software delivers a virtualisation and data protection layer that creates a unified, fixed content storage interface across multiple facilities and heterogeneous storage media. See `http://www-03.ibm.com/systems/storage/software/gam`.

GAM software enables formation of fixed content storage systems that can scale to petabytes of data across numerous sites. It has an efficient wide area replication to deliver a storage system spanning sites linked together with differing bandwidth networks. GAM software optimises broad availability of data across sites through network file system interfaces (CIFS and NFS) and as an object store delivering a global name space.

GFS: Google File System has a focus on fault tolerance, high throughput and scalability. It was originally developed for internal use, particularly to support a MapReduce style of data processing, to provide efficient, reliable access to data using large clusters of commodity hardware, but is now freely available as GFS-2. It is based on the concept of a 64MB chunk server and replica management. See Wikipedia. GFS currently does not have a Posix API.

Gluster: GlusterFS is a general purpose distributed file system for scalable storage based on clustering. It aggregates various storage bricks over Infiniband RDMA or TCP/IP GigE interconnect into one large parallel network file system. It has a stackable user space design.

GlusterFS has client and server components. Servers are typically deployed as storage bricks, with each server running a glusterfsd daemon to export a local file system as a volume. The

glusterfs client process, which connects to servers with a custom protocol over TCP/IP, InfiniBand or SDP, composes remote volumes into larger ones using stackable translators. The final volume is then mounted by the client host through the FUSE mechanism to provide a Posix interface. Applications doing large amounts of file i/o can also use the libglusterfs client library to connect to the servers directly and run in-process translators, without going through the file system and incurring FUSE overhead.

Most of the functionality of GlusterFS is implemented as translators, including: file based mirroring and replication; file based striping; file based load balancing; volume failover; scheduling and disk caching; storage quotas.

The GlusterFS server is kept minimally simple – it exports an existing file system as-is, leaving it up to client side translators to structure the store. The clients themselves are stateless, do not communicate with each other, and are expected to have translator configurations consistent with each other. This can cause coherency problems, but allows GlusterFS to scale up to several petabytes on commodity hardware by avoiding bottlenecks that normally affect more tightly coupled distributed file systems – there is no master node.

There seems to be good community support for Gluster, although most users seem to be from Web hosting companies, see `http://www.gluster.org`. It is being considered for use by Streamline Computing.

GPFS: the General Parallel File System is a high performance shared disk clustered file system for AIX and Linux developed by IBM. It is used by many of the supercomputers that populate the Top500 list. GPFS was evaluated in [8].

GPFS provides concurrent high speed file access to applications executing on multiple nodes of clusters. It can be used with AIX, Linux, Microsoft Windows Server 2003-r2 or a heterogeneous cluster. In addition to providing file system storage, GPFS provides tools for management and administration of the storage cluster and allows for shared access to file systems from remote GPFS clusters.

HDFS: Hadoop Distributed File System from Apache stores large files (an ideal size is a multiple of 64MB), across multiple nodes or machines. It is basically an open source implementation of GFS. It achieves reliability by replicating the data across multiple hosts, and hence does not require RAID storage on hosts or a SAN. With the default replication value of 3, data is stored on three nodes. For redundancy this is recommended to be two on the same rack, and one on a different rack in a cluster. Hadoop is a Java application which uses a variant of the Google MapReduce method to split data out, perform functions on the data in parallel, create replicas of the blocks and re-combine (reduce) it when required. In this way Hadoop can be used in the processing of very large distributed data sets, it is typically implemented in a functional programming language (the origin of MapReduce).

HDFS is thus built from a cluster of data nodes, each of which serves up blocks of data over the network using a protocol specific to HDFS. They can also serve the data over HTTP, allowing access to all content from a Web browser or other client. Data nodes can talk to each other to re-balance data, to move copies around, and to keep the replication of data high (which helps with access speed using peer-to-peer methods). It can also be used for storage scavaging.

A Hadoop file system requires one unique server, the name node. This is therefore a single point of failure for HDFS (or indeed GFS or KFS). When it comes back up, the name node must replay all outstanding operations. Another limitation of HDFS is that it cannot be directly mounted by an existing operating system as it does not have a Posix API. Getting data into and out of

the HDFS file system is an action that often needs to be performed before and after executing a job, so this can be inconvenient.

It was speculated in 2010 that Sun Grid Engine v6.2-5 would include support for Hadoop. Thus SGE, which is aware of HDFS, would be able to route processing jobs to where the data is already located in the nodes, which speeds up execution of those jobs. This is more efficient than starting up a job somewhere and then trying to move the data over to that node. It can also be of use for chaining multi-stage jobs which use large data sets. It was speculated in 2009 that Condor would take a similar approach from v7.4 onwards, how this has not yet been seen.

HDFS may also be valuable for data mining applications, see Section **??**. It is now available with support for running on Amazon EC2 or S3 clouds, although its performance has been questioned in this context.

A number of applications are becoming available built on Hadoop. Yale University's HadoopDB uses MapReduce and is aimed at petabyte databases. Hive is a data warehouse infrastructure with its own query language (called Hive QL). Hive makes Hadoop more familiar to those with a Structured Query Language (SQL) background, but it also supports the traditional MapReduce infrastructure for data processing. HBase is a high performance database system similar to Google BigTable. Instead of traditional file processing, HBase makes database tables the input and output form for MapReduce processing. Finally, PIG is a platform on Hadoop for analysing large data sets. PIG provides a high level language that compiles to MapReduce applications.

HP: StorageWorks file share for clusters is based on Lustre, see Section 3.5.4.

KFS: Kosmos File System is an open source implementation of GFS from Kosmix implemented in C++ with C++, Java and Python client support. It is also known as CloudStore. KFS exports a Posix file interface via FUSE. KFS can also be integrated with Hadoop MapReduce to replace HDFS. Kosmos is still not widely used, but see `http://sourceforge.net/projects/kosmosfs/`.

Lustre: originated as a project at Carnegie Mellon University in 1991 is an object based, distributed file system, generally used for large scale cluster computing. The project aims to provide a file system for clusters of 10,000s of nodes with petabytes of storage capacity, without compromising speed, security or availability. Due to the high scalability of Lustre file systems, deployments are popular in the oil and gas, manufacturing, rich media and finance sectors. Lustre was evaluated in [8].

Lustre is now developed and maintained by Oracle (previously Sun) with input from many other individuals and companies after the acquisition of Cluster File Systems, Inc. in 2007, with the intention of bringing the benefits of Lustre technologies to Sun's ZFS file system and the Solaris operating system. See `http://wiki.lustre.org/index.php/Main_Page`.

Lustre has failover, but multi-server RAID-1 or RAID-5 is still in their roadmap for future versions. Available for Linux under GPL. Largest current Lustre implementation is on the TeraGrid Data Capacitor [19].

MogileFS: from Danga Interactive is an open source distributed file system. It is not Posix compliant and is designed for archiving, i.e. write-once files which are read multiple times. This would be appropriate in a data collection scenario. It uses a flat namespace, application level, uses MySQL for metadata and NFS or HTTP for transport. Available for Linux (but may be ported) under GPL. See `http://www.danga.com/mogilefs`.

MooseFS: is an open source file system aimed at petabyte storage. It is fault tolerant and fully distributed. It is similar to GlusterFS in terms of being FUSE based and having built in replication, but different in other ways. It uses a "metadata logger" approach rather than true distributed metadata. Important features include: replica management; built in coherent file snapshots; dynamic expansion with new servers; retention of deleted files in a "trash bin". The Web site `http://www.moosefs.org/` provides a good explanation of the architecture and examples of usage world wide. Many applications are for image repositories.

NFS-4 Network File System (NFS) originally from Sun is an open standard in Posix based networked file systems. It is not itself a distributed file system, but may be used as an access mechanism. The NFS protocol allows clients on a network to mount shared file systems from one or more remote servers. NFS may use Kerberos authentication and a client cache. NFS v4.1 includes parallel file access and separates location metadata from the actual files. This is also referred to as pNFS. NFS-4 is developed and maintained by the IETF and is thus a standard rather than a full implementation – Panasas are taking a lead on implementation issues. See Wikipedia and also the AFS architecture Section 3.5.1 below.

OCFS: Oracle Cluster File System, currently OCFS-2 is a Posix compliant shared disk cluster file system for Linux capable of providing both high performance and high availability. It is available under a GPL license, see `http://oss.oracle.com/projects/ocfs2`.

Cluster aware applications can make use of parallel i/o. It can also be used as a fail over setup to increase its resiliance. Apart from being used with Oracle's Real Application Cluster (RAC) database product, OCFS-2 is currently in use to provide scalable Web servers, file servers, mail servers and for hosting virtual machine images.

Some of the notable features of OCFS are:

- Optimised allocations (extents, reservations, sparse, unwritten extents, punch holes);
- Inode based writeable snapshots;
- Indexed directories;
- Metadata checksums;
- Extended attributes (unlimited number of attributes per inode);
- Advanced security (Posix ACLs and SELinux);
- User and group quotas;
- Variable block and cluster sizes;
- Journaling (ordered and writeback data journaling modes);
- Endian and architecture neutral (x86, x86_64, ia64 and ppc64);
- Buffered, direct, asynchronous, splice and memory mapped i/o;
- In-built Clusterstack with a distributed lock manager
- Cluster aware tools (mkfs, fsck, tunefs, etc.)

PanFS: Panasas ActiveScale File System uses object storage devices available as a proprietary storage solution. The DirectFLOW capability offers users fully parallel i/o to allow high speed, direct communications between Linux clusters and Panasas storage. Panasas have won several industry awards and have had a large influence on pNFS, see Section 3.4. pNFS is expected to replace the DirectFlow client in time.

PanFS is used at LANL for RoadRunner and on the NW-GRID clusters and some other systems at Daresbury Laboratory. See `http://www.panasas.com`.

Parrot: Parrot is a virtual file system component of Condor that allows computation jobs to access data stored on remote servers. It is used for deploying campus Grids, see `http://www.cse.nd.edu/~ccl/software/parrot`.

PeerFS: from Radiant Data Corporation focusses on high availability and high performance and uses peer-to-peer replication with multiple sources and targets. Available for Linux under a proprietary software license, see `http://www.radiantdata.com`.

PVFS: Parallel Virtual File System for Linux based HPC clusters is available under GPL. It is designed to scale to petabytes of storage and provide access rates at 100s of GB/s. PVFS is supported by SciDAC through the Scientific Data Management Center and developed by a multi-institution team with an interest in reliability and performance supporting multiple hardware platforms and MPI-IO implementations. Currently PVFS-2, see `http://www.pvfs.org`. PVFS-2 was evaluated in [8]. These may in fact be just different software branches (sometimes referred to as orange and blue). GlusterFS may be an alternative.

SMB: Server Message Block originally from IBM operates as an application layer network protocol mainly used to provide shared access to files, printers, serial ports and miscellaneous communications between nodes on a network. It also provides an authenticated inter-process communication mechanism. Most usage of SMB involves computers running Microsoft Windows, where it is often known as Microsoft Windows Network. SMB is also known as Common Internet File System (CIFS) or Samba file system. SMB may use Kerberos authentication.

SRB and iRODS: Storage Resource Broker, originally from the San Diego Supercomputer Center, provides the capability to virtualise distributed storage resources and to provide standardised access to a broad range of underlying technologies, from flat file systems to databases and tape archiving systems. Through SRB, users are freed from concerns about the location of data and determining the correct procedures to recall or transfer data to their local or host compute environment. SRB abstracts these aspects of distributed data management away from the end user and provides a simplified and uniform way to recall data via indexing systems (meta-catalogues) which keep a logical mapping of the underlying distributed data. SRB has been widely adopted within large scale Grid applications, particularly in the science communities and provided the first data management backbone for the National Grid Service (NGS).

The developers of SRB have now developed iRODS, another data Grid software system with the important addition of a distributed rules engine. Users can execute rules and micro-services to automate the enforcement of management policies to control data access, manipulate data at distributed sites, etc. The use of rules provides iRODS with a flexibility that would have to be hard coded using SRB.

Tahoe: is a distributed file system from *allmydata.com*, which they claim safely stores files on multiple machines to protect against hardware failures. Cryptographic tools are used to ensure integrity and confidentiality, and a de-centralised architecture minimises single points of failure. Files can be accessed through a Web interface or native system calls via FUSE. Fine grained sharing allows individual files or directories to be delegated by passing short URI like strings through e-mail. Tahoe grids are easy to set up, and can be used by a handful of friends or by a large company for thousands of customers. Tahoe relies on distributing multiple copies of data split into blocks and using an erasure coding re-construction algorithm. It is coded in Python. See `http://allmydata.org/~warner/pycon-tahoe.html`.

3.5 Detailed Architectures

We have chosen to illustrate in more detail the architectures of six distributed file systems: AFS, Ceph, iRODS, Lustre, GPFS and Panasas. All are suitable for use in large data centres or experimental facilities.

3.5.1 AFS

AFS has several benefits over traditional networked file systems, particularly in the areas of security and scalability. It is not un-common for enterprise AFS cells to exceed 25,000 clients. AFS uses Kerberos for authentication and implements access control lists on directories for users and groups. Each client caches files on the local file system for increased speed on subsequent requests for the same file. This also allows limited file access in the event of a server crash or a network outage.

Read and write to an open file is directed only to the local copy. When a modified file is closed, the changes are copied back to the file server. Cache consistency is maintained by a callback mechanism. Clients are informed by the server if the file is changed elsewhere. Callbacks are discarded and must be re-established after any client, server or network failure, including a time-out. Re-establishing a callback involves a status check and does not require re-reading the file itself.

A consequence of the file locking strategy required to implement the above mechanism is that AFS does not support large shared databases or record updating within files shared between client systems. This was a deliberate design decision based on the perceived needs of the university computing environment at the time.

A significant feature of AFS is the "volume", a tree of files, sub-directories and AFS mount points (links to other AFS volumes). Volumes are created by administrators and linked at a specific named path in an AFS cell. Once created, users of the file system may create directories and files as usual without concern for the physical location of the volume. A volume may have a quota assigned to it in order to limit the amount of space consumed. AFS administrators can move that volume to another server and disk location as required without the need to notify users; indeed the operation can occur while files in that volume are being used.

AFS volumes can be replicated to read only cloned copies. When accessing files in a read only volume, a client system will retrieve data from a particular read only copy. If at some point that copy becomes unavailable, clients will look for any of the remaining copies. Again, users of that data are un-aware of the location of the read only copy; administrators can create and relocate such copies as needed. The AFS command suite guarantees that all read only volumes contain exact copies of the original read-write volume at the time the read only copy was created.

The file name space on an Andrew workstation is partitioned into a shared and local name space. The shared name space (usually mounted as `/afs` on the Unix file system) is identical on all workstations. The local name space is unique to each workstation. It only needs to contain temporary files needed for workstation initialisation and symbolic links to files in the shared name space.

AFS heavily influenced NFS-4. Additionally, a variant of AFS, the Distributed File System (DFS) was adopted by the Open Software Foundation in 1989 as part of their Distributed Computing Envi-

ronment.

There are currently three major implementations from Transarc (IBM), OpenAFS and Arla, although the Transarc software is losing support and is deprecated. AFS-2 is also the predecessor of the Coda file system.

A fourth implementation exists in the Linux kernel source code since at least version 2.6.10. This was committed by Red Hat, but is a fairly simple implementation in its early stages of development and therefore still incomplete.

Note: AFS is in use at University of Manchester and clients can be provided on the National Grid Service clusters if projects require them.

3.5.2 CEPH

Ceph has three main components: 1) a cluster of Object Storage Devices (OSDs), which collectively store all data and metadata; 2) a metadata server (MDS) cluster, which manages the namespace (file names and directories), consistency and coherence; 3) clients, each instance of which exposes a Posix like API.

Ceph storage consists of a potentially large number of OSDs, a smaller set of MDS daemons and a few monitor daemons for managing cluster membership and state. The OSDs handle data migration, replication, failure detection and recovery. They rely on the new Linux BTRFS object store and use an algorithm like hashing to compute the data location rather than using lookup tables (the data distribution function is referred to as CRUSH, Controlled Replica Under Scalable Hashing). Replicas are used in CRUSH to improve access and reliability. The storage cluster is simple to deploy, while providing better scalability than other current block based cluster file systems. The placement policy in CRUSH can also take into account storage and server hierarchy, e.g. utilising redundant or spatially separated devices to enhance resilience.

Metadata daemons compute the data location and use the Paxos consensus protocol to arbitrate access, see `http://en.wikipedia.org/wiki/Paxos_algorithm`. This avoids any need to exchange location metadata. There is typically one MSD per 100 OSDs.

Clients use the FUSE mechanism for Posix like i/o and cache data locations returned from the MDS.

Ceph is being considered for use at Imperial College London.

3.5.3 iRODS

iRODS stands for integrated Rule Oriented Data Systems. It is a second generation data grid system providing a unified view and seamless access to distributed digital objects across a wide area network. It is an evolution of the first generation Storage Resource Broker (SRB) which provided a unified view based on logical naming concepts – users, resources, data objects and virtual directories were abstracted by logical names and mapped onto physical entities thus providing a physical-to-logical independence for client level applications. iRODS builds on this by abstracting the data management process itself – this is referred to as policy abstraction.

iRODS v1.0 provides user friendly installation tools, a modular environment for extensibility through micro-services, a Web based interface and support for Java, C and shell programming through libraries and utilities for application development.

The "integrated" part of iRODS comes from the fact that it provides a unified software environment for underlying services which interact in a complex fashion among themselves. This idea is different to a toolkit methodology where one is provided with a suite of modules to be integrated by the user or application to form a customised system. IRODS integration exposes a uniform interface to the client application hiding the underlying complexity. SRB also had an integrated envelope methodology with a single server installation hiding the details of third party authentication, authorisation, auditing, metadata managment, streaming access mechanism, resource (vendor level), etc.

iRODS currently has around 100 API functions and 80 command level utilities. These build on the integrated envelope adding more functionality and services. Functionalities include the following.

- Data Transport;
- Metadata catalogue for both system and user defined metadata;
- Rule engine for executing complex policies encoded as micro-services;
- Execution engine for execution of remote micro-services as workflows;
- Scheduling system for immediate, delayed and periodic queuing and execution;
- Messaging system for out-of-band communication among micro-services;
- Virtualisation system enabling the logical naming paradigm.

For an evaluation of iRODS in the JISC funded iREAD project see `http://www.wrg.york.ac.uk/iread`.

Note: SRB has been used in the NERC funded e-Minerals e-Science project, the JISC funded Cheshire-3 VRE project and on the Diamond synchrotron facility. iRODS is under evaluation at STFC.

3.5.4 Lustre

Lustre is probably the most pervasive parallel file system on large scale systems at the time of writing. It is open source and found on aroud 60 of the top 100 systems. Since Lustre was a Sun product, there is now some doubt about continued support from Oracle from Lustre-2 onwards. The Open Scaleable File System Consortium is addressing some of the concerns.

A Lustre file system has three major functional units as follows.

- A single metadata target (MDT) per file system that stores metadata, such as file names, directories, permissions and file layout, on the metadata server (MDS);

- One or more object storage servers (OSS) that store file data on one or more object storage targets (OST). Depending on the server hardware, an OSS typically serves between two and eight targets, each target being a local disk file system up to 8TB. The capacity of a Lustre file system is the sum of the capacities provided by the targets;
- Client(s) that access and use the data. Lustre presents all clients with standard Posix API and concurrent read and write access to the files in the file system.

The MDT, OST, and client can be on the same node or on different nodes. In typical installations these functions are on separate nodes with two to four OSTs per OSS node communicating over a network. Lustre supports several network types, including InfiniBand, TCP/IP on Ethernet, Myrinet, Quadrics and other proprietary technologies. Lustre can take advantage of remote direct memory access (RDMA) transfers, e.g. over IB, to improve throughput and reduce CPU usage.

The storage attached to the servers is partitioned, optionally organised with logical volume management (LVM) and/ or RAID and formatted as file systems. The Lustre OSS and MDS servers read, write, and modify data in the format imposed by these file systems.

An OST is a dedicated file system that exports an interface to byte ranges of objects for read and write operations. An MDT is a dedicated file system that controls file access and tells clients which object(s) make up a file. MDTs and OSTs currently use a modified version of ext3 to store data. In the future it was planned that Sun's ZFS or DMU would be used.

When a client accesses a file, it completes a filename lookup on the MDS. As a result, a file is created on behalf of the client or the layout of an existing file is returned to the client. For read or write operations, the client then passes the layout to a logical object volume (LOV), which maps the offset and size to one or more objects, each residing on a separate OST. The client then locks the file range being operated on and executes one or more parallel read or write operations directly to the OSTs. With this approach, bottlenecks for client-to-OST communications are eliminated, so the total bandwidth available for the clients to read and write data scales almost linearly with the number of OSTs in the file system.

Clients do not directly modify the objects on the OST file systems, but, instead, delegate this task to OSSes. This approach ensures scalability for large scale clusters and super-computers, as well as improved security and reliability.

In a typical Lustre installation on a Linux client, a Lustre file system driver module is loaded into the kernel and the file system is mounted like any other local or network file system. Client applications see a single, unified Posix like file system even though it may be composed of tens to thousands of individual servers and MDT or OST file systems.

On some HPC installations, computational nodes can access a Lustre file system by re-directing their i/o requests to a dedicated node configured as a Lustre client. This approach was for instance used in the LLNL BlueGene installation.

Another approach uses the liblustre library to provide user space applications with direct file system access. Liblustre is a user level library that allows nodes to mount and use the Lustre file system as a client. Using liblustre, the nodes can access the file system even if the service node on which the job was launched is not a Lustre client. Liblustre allows data movement directly between application space

and the Lustre OSSes without requiring an intervening data copy through the kernel, thus providing low latency, high bandwidth access from nodes directly to Lustre. Good performance characteristics and scalability make this approach the most suitable for using Lustre with HPC systems. Liblustre is the most significant design difference between Lustre implementations on systems such as Cray XT3 and Lustre implementations on conventional clustered workstations.

High availability features include a robust failover and recovery mechanism, making server failures and re-boots transparent. Version inter-operability between successive minor versions of the software enables a server to be upgraded by taking it off-line (or failing it over to a standby server), performing the upgrade, and re-starting it, while all active jobs continue to run. Users merely experience a delay.

Note: Lustre is being deployed for the Diamond synchrotron facility. It is used on the Jaguar system at ORNL which supports some 26,000 file system clients and 10PB of RAID6 storage. Lustre is also deployed on HECToR and will be part of the HPC Wales Grid with Fujitsu's Exa-byte File System (FEFS) as deployed in the K-computer in Riken. It is also used on WhamCloud supported by OpenSFS.

3.5.5 GPFS

GPFS is an IBM proprietary high performance file system. GPFS provides higher i/o performance by "striping" blocks of data from individual files over multiple disks and reading and writing them in parallel. Other features provided by GPFS include high availability, support for heterogeneous clusters, disaster recovery, security, DMAPI, HSM and ILM. See `http://www.ibm.com`.

A file that is written to GPFS is broken up into blocks of a configured size, typically less than 1MB each. These blocks are distributed across multiple nodes, so that a single file is fully distributed across the disk array. This results in high read and write speeds as the combined bandwidth of the many physical drives is high. This however makes the file system vulnerable to disk failures, so to prevent data loss, the file system nodes also have RAID controllers. It is also possible to replicate blocks on different file system nodes.

Other features of the file system include the following.

- Distributed metadata, including the directory tree. There is no single "directory controller" or "index server".
- Efficient indexing of directory entries for very large directories. Many file systems are limited to a small number of files in a single directory. GPFS does not have such limits.
- Distributed locking. This allows for full Posix file system semantics, including locking for exclusive file access.
- Partition Aware. The failure of the network may partition the file system into two or more groups of nodes that can only see the nodes in their group. This can be detected through a heartbeat protocol so that when a partition occurs, the file system remains live for the largest partition formed. This offers a graceful degradation of the fileystem.

- File System maintenance can be performed on-line. Most file system maintenance tasks including adding new disks and re-balancing data across disks can be performed while the file system is live.

For ILM, storage pools allow for the grouping of disks within a file system. Tiers of storage can be created by grouping disks based on criteria of performance, locality or reliability.

A file set is a sub-tree of the file system namespace and provides a way to partition the namespace into smaller, more manageable units. File sets provide an administrative boundary that can be used to set quotas and be specified in a policy to control initial data placement or data migration. Data in a single file set can reside in one or more storage pools. Where the file data resides and how it is migrated is based on a set of rules in a user defined policy.

There are two types of user defined policies in GPFS: file placement and file management. File placement policies direct file data as files are created to the appropriate storage pool. File placement rules are determined by attributes such as file name, the user name or the fileset. File management policies allow the file's data to be moved or replicated or files deleted. File management policies can be used to move data from one pool to another without changing the file's location in the directory structure. File management policies are determined by file attributes such as last access time, path name or size of the file.

The GPFS policy processing engine is scalable and can be run on many nodes at once. This allows management policies to be applied to a single file system with billions of files and complete in a few hours.

Note: GPFS is used at Daresbury Laboratory for HPCx, BlueGene, iDataPlex, POWER-7 and related services. It is also used at POL, the Proudman Oceanographic Laboratory and other UK academic data centres including many members of the UK HPC-SIG.

3.5.6 Panasas

PanFS has roots in common with Lustre as they are contemporary designs from Carnegie Mellon University. Garth Gibson, CTO of Panasas and former professor at CMU was also a co-author of RAID in 1988.

The Panasas system is a specialised storage cluster. It uses per-file, client driven RAID, has parallel RAID rebuild, treatment of different classes of metadata (block, file, system) and a commodity parts based blade hardware with integrated UPS. It also has many other NOW standard features such as object storage, fault tolerance, caching and cache consistency and a simple management model.

The storage cluster is divided into storage nodes and manager nodes at a ratio of typically about 10:1. The storage nodes implement an object store, and are accessed directly from PanFS clients. The manager nodes control the storage cluster, implement the distributed file system semantics, handle failure recovery and can export the Panasas file system via NFS and CIFS.

Each file is striped over two or more objects to provide redundancy and high bandwidth access. The file system semantics are implemented by metadata managers that mediate client access to objects

using the iSCSI/OSD protocol for read and write operations. I/O proceeds directly and in parallel to the storage nodes, bypassing the metadata managers. The clients interact with the metadata managers out-of-band via RPC to obtain access capabilities and location information for the objects that store files.

Object attributes are used to store file level attributes, and directories are implemented with objects that store name to object ID mappings. Thus the file system metadata is kept in the object store itself, rather than being kept in a separate database or some other form of storage on the metadata nodes.

Note: Panasas is used at Daresbury on several clusters and also at other NW-GRID sites. Panasas is expected to be used on the LANL Cielo system (Cray). It is used at other US government sites including Lawrence Berkeley, LLNL, NASA, ORNL and Sandia.

4 Case Studies, Technologies and Tools

In this section we illustrate some use cases with collections of technologies and tools in use in typical data centric research environments.

Typical requirements are to have high bandwidth for parallel i/o within an HPC system, HSM for data migration and backup, distributed replicas for resiliance, multi-access within a data centre for pre- and post-processing. Remote access may be required with search facilities using metadata. It is likely that such a system will be coupled with a dedicate large memory server or other data intensive system and have the capability to export compressed data streams for remote visualisation.

4.1 SciDAC Data Management Center

The overall architecture of the SciDAC Data Management Center has been described in Section 1. For further details about the project, see [25]. In practice the project has adopted the following software technologies.

- PVFS: Parallel Virtual File System;
- ROMIO: a high performance MPI-IO implementation;
- Parallel NetCDF: A high performance API for NetCDF;
- Kepler: scientific workflow application;
- ProRata: data analysis software for quantitative proteomics;
- FastBit: highly efficient indexing and searching software for scientific data;
- Sapphire: scientific data mining software;
- SRM-Lite: a light weight data mover tool;
- Active Storage: leveraging the computing capacity of the storage nodes.

4.2 IDIES, Johns Hopkins University

The Institute for Data Intensive Engineering and Science (IDIES) was founded in Apr'2009. It was based on the work of Alex Szalay and Jim Gray who provided a large scale database to allow astronomers to use SQL queries to extract data and execute user defined functions on 12TB data from the Sloan Digital Sky Survey. Applications from IDIES are explored further in [2].

This initial work has now been extended to a number of research domains including: turbulence, with a 27TB database; biology and environment, with 120M observations from forest sensor networks; data center monitoring using wireless sensors in collaboration with Microsoft; computer science research into data preservation and parallel query optimisation; data intensive architectures such as the GrayWulf and Amdahl Blade; 3D surface fitting such as in the Stanford Digital Michaelangelo Project and the LIDAR survey of New York City; neuro-science databases for statistical inference on EM and MR imaging data; Pan-STARRS asteroid database; Large Synoptic Survey Telescope data. Many of the services are publicly acessible via the Internet and have been used for "crowd sourcing" projects such as GalaxyZoo where users were asked to visually identify galaxy types.

Hardware available includes: GrayWulf, 50 Dell servers (500 CPU) and 1PB disk, Amdahl number to memory 1.0 and to disc 0.5; a 1,200 core cluster with 2TB memory connected using InfiniBand to database servers and the GrayWulf; 50 nodes with 100 nVidia GPUs to execute user defined DB functions out of process (SQLCLR); 36 node Amdahl Blade system, N330 dual core Atom, 4GB memory, 16 GPU cores total 76TB disk including SSD; visualisation facility producing 3D video streams from PB data sets with remote interaction; 10Gb/s dedicated connection to ORNL and UIC; proposed DataScope facility.

4.3 Data Intensive Computing at PNNL

The Pacific Northwest National Laboratory's approach to data intensive computing initially focused on three key research areas. From 2006-10 they developed and combined new technologies to create capabilities to test: (1) enabling scientific discovery and insight applied to remediating the environment; (2) decision support and control in securing cyber networks; and (3) situational awareness and response in preventing terrorism.

Software Architectures : Middleware for Data Intensive Computing (MeDICi), incorporates information integration capabilities, a virtualised data centre and a workflow engine to support the development of domain agnostic solutions. Support Architecture for Large Scale Sub-surface Analysis (SALSSA) is a framework used to study flow and transport problems. It uses CCA, the Common Component Architecture, to create a coupled model and an SPH, Smooth Particle Hydrodynamics, algorithm is built into this.

Hybrid Hardware Architectures : Research in hybrid computing evaluates the use of multi-threaded hardware architectures such as the following: (1) Cray XMT suitable for irregular memory access and fine grained synchronisation; (2) field programmable gate arrays (FPGA) for high throughput processing; and (3) multi-core processors that drive the analytics closer to the source.

Data Warehousing and Database: A Netezza TwinFin Warehouse Appliance is used for rapid

analysis of large data sets using a high performance parallel database for near real time feature extraction and mining. Algorithms can be compiled in R to run on the system. It can be accessed from the Cray XMT using JDBC or similar.

Analytic Algorithms and Visualisation : Advanced analytics use novel algorithms to provide real time analysis and visualisation capabilities for exploration and diagnostic discovery to facilitate human understanding.

Some propsed applications include: Social media analysis; Contingency analysis for the electric power grid; High-throughput video analysis; Understanding text documents; Architectural studies on multithreaded languages; Chapel language for hybrid systems methods; Dynamic network analysis; Social network analysis; Irregular database and runtime systems; Compiler and runtime system; Performance analysis and tools; Communication software for hybrid systems. The systems are also used for applications such as un-structured mesh generation and machine learning.

4.4 HPCx and NW-GRID

The National HPCx service ran from 2002-2008, and NW-GRID started in 2005. Technologies deployed include:

- distributed multi-core clusters with Grid middleware for access;
- Panasas: for high performance cluster file store;
- GPFS: for system wide file store across multiple clusters;
- MPI-2 and MPI-IO;
- parallel HDF-5;
- Tivoli: for backup and retrieval from tape store.

Note: HECToR is using Lustre.

4.5 CERN and STFC Infrastructures for Experimental Data

- ADS: the Atlas Data Store at RAL (similar large tape stores at CERN).
- CASTOR-2: overall hierarchical storage management;
- Oracle: for metadata and system information;
- LSF: for data transfer scheduling;
- Globus GridFTP: for remote file access;
- StorageD: drop box for upload and download, uses a separate database, cache and transfer queue;

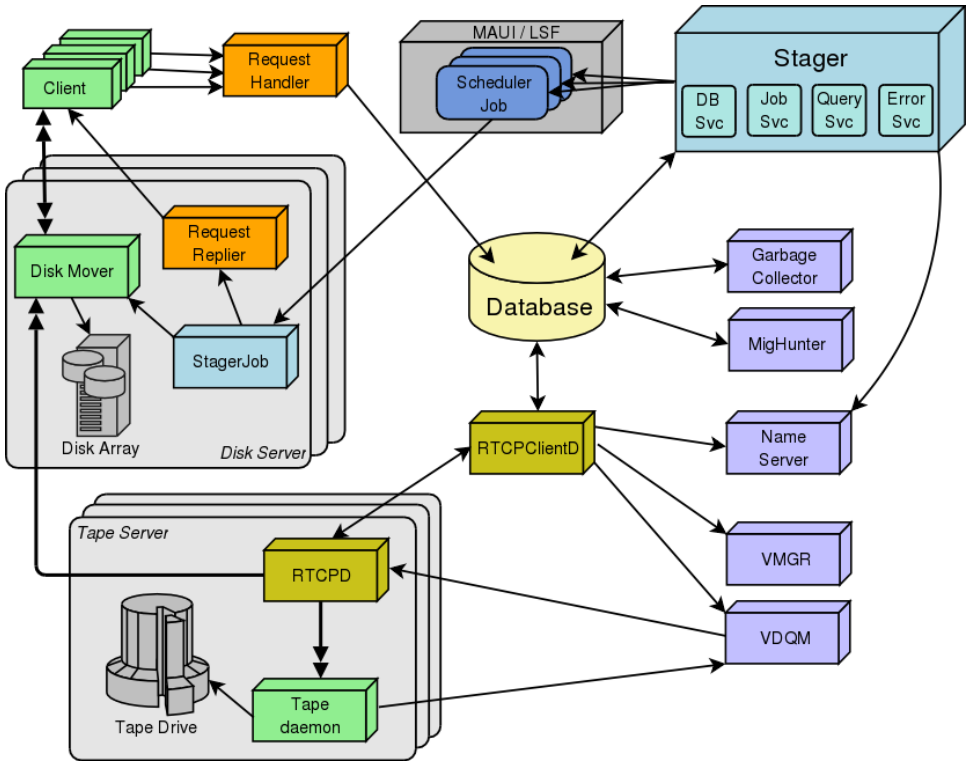

Figure 2: Figure 3: Architecture of CASTOR-2

- ICAT and TopCat, for metadata management and user access including search facilities;
- SRM-2: middleware for remote data access and management;
- Era: monitoring.

CERN currently uses CASTOR-2 to manage storage of LHC experimental data. The system design goals are to provide reliable central recording of the data, plus transparent access. Much of this access is intended to be from remote sites around the world via data Grid tools such as SRM. CASTOR itself is designed around a mass storage (tape) system and is therefore not suitable for deployment at sites without this facility.

CASTOR-2 is such a large installation it only exists at the largest sites in EGEE. These are CERN, plus three of the Tier-1 centres including RAL (UK), CNAF (Italy) and ASGC (Taiwan). The instance at CERN currently manages more than 50 million files and 5PB of data. The work at RAL [**?**] is standardising on CASTOR-2 plus some additional e-Science software components for management of data across all of the STFC experimental facilities. Variants are already in use on DLS, ISIS and CLF.

CASTOR-2 has been designed around a central Oracle database, which handles, as much as possible, the state of the system. The resilience of this is key to CASTOR's reliability. Around this database core all daemons are designed to be stateless, allowing for redundancy and daemon restarts without loss of service.

The Stager is a key component responsible for file migration. This manages the disk pools in front of the tape system, see Figure 2.

In order for the Stager to manage access to files on the disk pool it uses a scheduler plugin. This balances the load on the disk servers and can also provide policy such as "fair share" for file access. The scheduling problem for disk pools is similar to that faced in compute clusters, so CASTOR currently uses the commercial LSF batch scheduler from Platform Computing (recently acquired by IBM).

CASTOR has the ability to dynamically replicate "hot" files and even to switch access on an open file to a less busy replica.

The Name Server implements a hierarchical "directory" view of files stored in CASTOR. Files may actually be segmented, replicated and stored on various media in the system. The Name Server interface implements functionality required by the Posix standard.

Catalogue services are provided by ICAT which has a metadata database based on CSMD (the Core Scientific Metadata Model) [15]. ICAT also includes user session management. Functionality is provided via a web service so that a variety of client applications can be used. ICAT is available as open source from `http://code.google.com/p/icatproject/` and has active support and development. The TopCat client provides a browser based user interface for context aware search and data access via ICAT. TopCat is available as open source from `http://code.google.com/p/topcat`. TopCat has replaced DataPortal. Access control of metadata and data is supported. The releases of TopCat and ICAT are coordinated to ensure continuing compatibility. ICAT therefore responds to the need to make scientific data collected from experimental facilities available and re-usable, see Appendix B. The data itself is stored, curated and retained in other systems such as CASTOR. ICAT captures the context of the data including provenance and relevant experimental conditions.

Monitoring of the CERN system is carried out by integration with the LHC Era. Alarms are issued when any abnormal conditions are detected. Logging is done into the Oracle database, allowing the central gathering of information, plus the ability to cross query logs from different services.

5 Acknowledgements

This work was partly funded by EPSRC through the SLA with Daresbury Laboratory.

The author would like to thank the following people for input:

June Finch (NeISS Project and University of Manchester), Simon Hodson (JISC), Jason Lander (NGS and University of Leeds), Brian Matthews (RAL), Alistair Mills (RAL), Terry Hewitt (WTH Associates Ltd.),

References

[1] R.J. Allan *Virtual Research Environments: from Portals to Science Gateways* (Chandos Publishing, Oxford, 2009) 230pp in press `http://www.woodheadpublishing.com/en/book.aspx?`

bookID=1892&ChandosTitle=1

[2] R.J. Allan *Data Intensive Computing* DL Technical Report (2011) draft See http://www.grids.ac.uk/NWGrid/DataIntensive/

[3] R.J. Allan and K. Kleese *Data Management 2000* Proc. Intl. Workshop on Advanced Data Storage and Management for HPC. DL-CONF-00-001 (May 2000)

[4] C. Angeli, G.L. Bendazzoli, S. Borini, R. Cimiraglia, A. Emerson, S. Evangelisti, D. Maynau, A. Monari, E. Rossi, J. Sanchez-Marin, P.G. Szalay and A. Tajti. *The problem of interoperability: a common data format for quantum chemistry codes* COST D23 MetaChem working group draft http://abigrid.cineca.it/the-docs-archive/pubblications/Papero1.8_finalDraft.pdf

[5] J.V. Ashby, C. Greenough and R.J. Allan *Data management Tools for High Performance Applications* Technical Report (UKHEC, 2001) RAL-TR-2001-013 http://epubs.cclrc.ac.uk/work-details?w=29541

[6] G. Bell, A.J.G. Hey and A. Szalay *Beyond the Data Deluge* Science 323 (AAAS, 6/3/2009) 1297-8. DOI 10.112/science.1170411

[7] G. Brown, D. Corney, B. Matthews and A. Mills *Towards a Common Data Management Infrastructure for Large Scale Facilities* (STFC e-Science Centre, Nov'2011) unpublished

[8] J. Cope, M. Oberg, H.M. Tufo and M. Woitaszek *Shared Parallel Filesystems in Heterogeneous Linux Multi-Cluster Environments* (University of Colorado))

[9] Y. Gu, R.L. Grossman and J. Mambretti *A Peer-to-Peer Infrastructure for Distributing Large Scientific Data Sets over Wide Area High-Performance Networks: Experimental Studies Using Wide Area Layer 2 Services* http://www.rgrossman.com/dl/proc-106.pdf

[10] M. Grove *Necho – A System for Distributing and Managing Very Large Datasets* University of Reading (2007) http://acet.reading.ac.uk/projects/necho/index.php

[11] P. Halfpenny *et al. Draft Green Paper on SAC: Storage, Archiving and Curation* v2.2 (Manchester Informatics, Sep'2010) http://www.merc.ac.uk/sites/default/files/SACReportv2.2.pdf

[12] I. Kozin and M. Deegan *Evaluation of the eXludus Grid Optimiser* (DL, Nov'2007) http://www.cse.scitech.ac.uk/disco/publications/eXludus_GridOptimizer.pdf

[13] G. Mallinson *CFD Visualisation: Challenges of Complex 3D and 4D Data Fields* Int. J. Com. Fluid Dynamics 22:1-2 (Jan'2008) 49-59

[14] B. Mann, R. Williams, M. Atkinson, K. Brodlie, A. Storkey and C. Williams *Scientific Data Mining, Integration, and Visualization* Report of the workshop held at the e-Science Institute, Edinburgh, 24-25/10/2002. http://umbriel.dcs.gla.ac.uk/NeSC/general/talks/sdmiv/report.pdf

[15] B. Matthews, S. Sufi, D. Flannery, L. Lerusse, T. Griffin, M.T. Gleaves and K. Kleese *Using a Core Scientific Metadata Model in Large Scale Facilities* Int. J. of Digital Curation 5:1 (2010) 106-18

[16] C. Mountford *EXludus Evaluation for the ETF* (University of York, 2007) Technical Report UKeS-2007-04 http://www.wrgrid.org.uk/eXludus.pdf

[17] A. Osuna, G. Miller, B. Poston and J. Auvenshine *Introducing the IBM Grid Access Manager* (IBM Redbooks, 2008, SG24-7612-00) 90pp ISBN 0738485012 http://www.redbooks.ibm.com/abstracts/sg247612.html?Open

[18] P. Simmonds, J. Stroyan, N. Brown and L. Parker-Rhodes *Data Centres: the Use, Value and Impact* (RIN and JISC, Sep'2011) http://www.rin.ac.uk/data-centres

[19] S.C. Simms, G.G. Pike and D. Balog *Wide Area Filesystem Performance using Lustre on the TeraGrid* Proc. TeraGrid Conference (2007) http://datacapacitor.researchtechnologies.uits.iu.edu/lustre_wan_tg07.pdf

[20] R.R. Sinha, A. Termehchy, M. Winslett, S. Mitra and J. Norris *Maitri: A Format-Independent Framework for Managing Large Scale Scientific Data* http://dais.cs.uiuc.edu/~termehch/cidr-2007.pdf (2007)

[21] G.A. Stewart, D. Cameron, G.A. Cowan and G. McCance *Storage Management in EGEE* Proc. 5th Australasian symposium on ACSW frontiers 68 (2007) 69-77 http://epp.ph.unimelb.edu.au/twiki/pub/EPP/WebHome/storage_and_dm.pdf

[22] M. Valle *Scientific Data Management – an Introduction* CSCS (2008) http://personal.cscs.ch/~mvalle/sdm/scientific-data-management.html

[23] J. Vetter and K. Schwam *Techniques for High Performance Computational Steering* IEEE Concurrency 7:4 (1999) 63-74 doi:10.1109/4434.806980

[24] S.A. Weil, S.A. Brandt, E.L.Miller and C. Maltzahn *CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data* (UC Santa Cruz, 2006)

[25] SciDAC Scientific Data management Center. Web sites: https://sdm.lbl.gov/sdmcenter and http://www.scidac.gov/ASCR/ASCR_SDM.html

[26] *Scientific Data Format FAQ* http://www.cv.nrao.edu/fits/traffic/scidataformats/faq.html

[27] *NetCDF FAQ* http://www.unidata.ucar.edu/software/netcdf/docs/faq.html

[28] *Digital Curation Centre Glossary* http://www.dcc.ac.uk/resource/glossary

[29] *DCC Guide: How to Create a Data Plan* DCC (Sep'2011) http://www.dcc.ac.uk/news/new-dcc-guide-how-develop-data-plan

[30] *DCC Curation Reference Manual* (DCC) http://www.dcc.ac.uk/resources/curation-reference-manual.

[31] *Managing Research Data* Collection of papers (DCC) http://www.facetpublishing.co.uk/title.php?id=7562&category_code=810

[32] *Digital Preservation Coalition Definitions* http://www.dpconline.org/graphics/intro/definitions.html

[33] http://www.epsrc.ac.uk/about/standards/researchdata/Pages/expectations.aspx

[34] *Principles and Guidelines for Access to Research Data from Public Funding* (OECD Publications, 2007) http://www.oecd.org/dataoecd/9/61/38500813.pdf

[35] *Research governance framework for health and social care* (NHS, 2005, 2006) http://www.dh.gov.uk/en/Publicationsandstatistics/Publications/PublicationsPolicyAndGuidance/DH_4108962

[36] I2S2 Project *Research Activity Lifecycle Model* (UKOLN, 2007) .

[37] (JISC, 2010) http://www.jisc.ac.uk/publications/programmerelated/2010/foiresearchdata.aspx.

A RCUK Principles on Data Management and Sharing

http://www.rcuk.ac.uk/research/Pages/DataPolicy.aspx

- Publicly funded research data are a public good, produced in the public interest, which should be made openly available with as few restrictions as possible in a timely and responsible manner that does not harm intellectual property.
- Institutional and project specific data management policies and plans should be in accordance with relevant standards and community best practice. Data with acknowledged long-term value should be preserved and remain accessible and usable for future research.
- To enable research data to be discoverable and effectively re-used by others, sufficient metadata should be recorded and made openly available to enable other researchers to understand the research and re-use potential of the data. Published results should always include information on how to access the supporting data.
- RCUK recognises that there are legal, ethical and commercial constraints on release of research data. To ensure that the research process is not damaged by inappropriate release of data, research organisation policies and practices should ensure that these are considered at all stages in the research process.
- To ensure that research teams get appropriate recognition for the effort involved in collecting and analysing data, those who undertake Research Council funded work may be entitled to a limited period of privileged use of the data they have collected to enable them to publish the results of their research. The length of this period varies by research discipline and, where appropriate, is discussed further in the published policies of individual Research Councils.
- In order to recognise the intellectual contributions of researchers who generate, preserve and share key research datasets, all users of research data should acknowledge the sources of their data and abide by the terms and conditions under which they are accessed.
- It is appropriate to use public funds to support the management and sharing of publicly-funded research data. To maximise the research benefit which can be gained from limited budgets, the mechanisms for these activities should be both efficient and cost-effective in the use of public funds.

B STFC Scientific Data Policy

STFC, through the facilities it operates and subscribes to and the grants it funds, is one of the main UK producers of scientific data. This data is one of the major outputs of STFC and a major source of its economic impact. STFC, as a publicly funded organisation, has a responsibility to ensure that this data is carefully managed and optimally exploited, both in the short and the long term.

B.1 Scope

This policy applies to all scientific data produced as a result of STFC funding:

- Through grants to universities in particle physics, astronomy and nuclear physics.
- Through access to beam time at STFC supported facilities, e.g. ISIS, CLF, Diamond, ILL, ESRF.
- Through STFC subscriptions to other organisations, e.g. CERN, ESO.

This includes data produced as a result of past funding from STFC or its predecessor organisations (e.g. PPARC, CCLRC) which has already been curated.

This policy does not apply to:

- Data resulting from or relating to work carried out by STFC under contract/ service level agreement with other organisations, e.g. data resulting from work on HECToR (EPSRC) or the National Grid Service (JISC, EPSRC), data curated at the BADC (NERC), or data arising from commercial use of facility beam time. Policy regarding such data is the responsibility of the contracting organisation.
- Research outputs arising from STFC funding, such as publications.
- Software as a form of data in its own right (as distinct from software required to make use of data).
- Physical collections of items, which may be considered as a form of database. However, it would be expected that similar considerations concerning curation and exploitation would also apply in such cases.
- Data which are purely administrative in nature.

B.2 General principles

1. STFC policy incorporates the joint RCUK principles on data management and sharing (see Appendix A). Those principles are therefore not repeated here.
2. Both policy and practice must be consistent with relevant UK and international legislation.

3. For the purposes of this policy, the term "data" refers to: (a) "raw" scientific data directly arising as a result of experiment/ measurement/ observation; (b) "derived" data which has been subject to some form of standard or automated data reduction procedure, e.g. to reduce the data volume or to transform to a physically meaningful coordinate system; (c) "published" data, i.e. that data which is displayed or otherwise referred to in a publication and based on which the scientific conclusions are derived.
4. STFC is not responsible for the use made of data, except that made by its own employees.
5. Data management plans should exist for all data within the scope of the policy. These should be prepared in consultation with relevant stake holders and should aim to streamline activities utilising existing skills and capabilities, in particular for smaller projects.
6. Proposals for grant funding, for those projects which result in the production or collection of scientific data, should include a data management plan. This should be considered and approved within the normal assessment procedure.
7. Each STFC operated facility should have an ongoing data management plan. This should be approved by the relevant facility board and, as far as possible, be consistent with the data management plans of the other facilities.
8. Where STFC is a subscribing partner to an external organisation, e.g. as a member of CERN, STFC will seek to ensure that the organisation has a data management policy and that it is compatible with the STFC policy.
9. Data management plans should follow relevant national and international recommendations for best practice.
10. Data resulting from publicly funded research should be made publicly available after a limited period, unless there are specific reasons (e.g. legislation, ethical, privacy, security) why this should not happen. The length of any proprietary period should be specified in the data management plan and justified, for example, by the reasonable needs of the research team to have a first opportunity to exploit the results of their research, including any IP arising. Where there are accepted norms within a scientific field or for a specific archive (e.g. the one year norm of ESO) they should generally be followed.
11. "Published" data should generally be made available within six months of the date of the relevant publication.
12. "Publicly available" means available to anyone. However, there may a requirement for registration to enable tracking of data use and to provide notification of terms and conditions of use where they apply.
13. STFC will seek to ensure the integrity of any data and related metadata that it manages. Any deliberate attempt to compromise that integrity, e.g. by the modification of data or the provision of incorrect metadata, will be considered as a serious breach of this policy.

B.3 Recommendations for good practice

1. STFC recommends that data management plans be formulated following the guidance provided by the Digital Curation Centre `http://www.dcc.ac.uk/resources/data-management-plans`. STFC (e-Science department) can provide advice upon request.

2. STFC would normally expect data to be managed through an institutional repository, e.g. as operated by a research organisation (such as STFC), a university, a laboratory or an independently managed subject specific database. The repository(ies) should be chosen so as to maximise the scientific value obtained from aggregation of related data. It may be appropriate to use different repositories for data from different stages of a study, e.g. raw data from a crystallographic study might be deposited in a facility repository while the resulting published crystal structure might be deposited in an International Union of Crystallography database.

3. Plans should provide suitable quality assurance concerning the extent to which data can be or have been modified. Where "raw" data are not to be retained, the processes for obtaining "derived" data should be specified and conform to the standard accepted procedures within the scientific field at that time.

4. Plans may reference the general policy(ies) for the chosen repository(ies) and only include further details related to the specific project. It is the responsibility of the person preparing the data management plan to ensure that the repository policy is appropriate. Where data are not to be managed through an established repository, the data management plan will need to be more extensive and to provide reassurance on the likely stability and longevity of any repository proposed.

5. Plans should cover all data expected to be produced as a result of a project or activity, from "raw" to "published".

6. Plans should specify which data are to be deposited in a repository, where and for how long, with appropriate justification. The good practice criteria assume that this data is accompanied by sufficient metadata to enable re-use. It is recognised that a balance may be required between the cost of data curation (e.g. for very large data sets) and the potential long term value of that data. Wherever possible STFC would expect the original data (i.e. from which other related data can in principle be derived) to be retained for the longest possible period, with ten years after the end of the project being a reasonable minimum. For data that by their nature cannot be re-measured (e.g. earth observations), effort should be made to retain them "in perpetuity".