EUROPEAN FUSION DEVELOPMENT AGREEMENT

EFDA–JET–CP(01)04-05

R. Layne, M. Wheatley and JET EFDA Contributors

New Data Storage and Retrieval Systems for JET Data

New Data Storage and Retrieval Systems for JET Data

R. Layne, M. Wheatley
and JET EFDA Contributors*

EURATOM-UKAEA Fusion Association, Culham Science Centre, Abingdon, OX14 4XB, United Kingdom.
**See Annex of J. Pamela et al., "Overview of Recent JET Results and Future Perspectives", Fusion Energy 2000 (Proc. 18th Int. Conf. Sorrento, 2000), IAEA, Vienna (2001).*

Preprint of Paper to be submitted for publication in Proceedings of the
3rd IAEA TCM on Control, Data Acquisition and Remote Participation,
(Padova, 16-19 July 2001)

ABSTRACT

Since the start of JET, an IBM mainframe has been the main platform for data analysis and storage[1]. The mainframe was removed in June 2001 and Solaris and Linux are now the main data storage and analysis platforms. New data storage and retrieval systems have therefore been developed: the Data Warehouse, the JET Pulse File (JPF) server, and the Processed Pule File (PPF) system. These support access to the data from codes running on many distributed clients on platforms including Linux, Windows NT and Solaris at JET, and at other fusion laboratories using JET Remote Data Access (RDA) software[2] and MDSplus[3]. A requirement for all these systems was to insulate users from the change in architecture and operating system by supporting a programming interface identical to the old IBM systems. A further requirement was to improve the existing systems by using current technology to make the systems more maintainable, and allow extra functionality to be added in the future. The new systems had to be developed while JET operations continued in parallel. The new systems will be described, and the design decisions that led to the final systems will be outlined.

1. THE DATA WAREHOUSE

1.1 THE HARDWARE

The replacement hardware for the IBM mainframe consists of a Sun Microsystems ES4500 with 4 400Mhz CPUs , 4 Gbytes of system memory, over 1.5 Tbytes of FCAL attached disks (connected via two independent FCAL interfaces) and a StorageTek 9310 robotic tape silo with 5760 slots in 6 model 9840 tape drives (connected to the ES4500 via six dedicated SCSI interfaces), each tape cartridge being able to hold around 40 Gbytes of data.

The 42 FCAL disks are located in three Sun StorEdge A5100 shelves with each disk being mirrored to a disk in a different shelf. Together with the two FCAL interfaces this arrangement provides multiple redundant paths to each drive and allows the system to continue operating in various failure scenarios.

1.2 DATA NAMING

Each item of data has two names associated with it -the **data source name** and the **warehouse item name**.

The data source name can be likened to the Unix filesystem path on the source computer. Examples are: /jet/PF/GAP/JPF/on/1234567 is the JET machine Poloidal Field (PF) sub-system component of the synchronous (online) pulse file for shot number 1234567.

/jet/PPF/user/1234567 89-chain1 is the JET processed pulse file (PPF) for pulse 1234567, analysis sequence number 89 for user “chain1”.

The warehouse item name is generated by applying a substitution algorithm to the data source name.

The **warehouse index** is a simple UNIX directory structure and, without this translation of the

name, many of the index directories would have hundreds of thousands of entries, which would result in significantly degraded file-system performance. The warehouse item names for the above examples would be:

/jet/PF/GAP/JPF/on/1234___/1234567

and

/jet/PPF/user/chain1/1234___/1234567/89

1.3 THE SOFTWARE

JET has around 8 Tbytes of data and new data is currently being generated at around 1 Gbyte per pulse. With our current storage, it is not practical to have this amount of data on-line. To overcome this problem we use a data migration/recall strategy. There are three basic warehouse disk areas - **Control**, **Index** and **Spool**.

The **Control** area has definitions for each type of data that the warehouse will accept. This includes the data names, which host the data can be sent from, whether the data can be overwritten, and how the warehouse item name is generated from the data source name.

The **Index** area holds status information about each data item in the warehouse, such as how many copies exist on tape and the date and time that each copy was made.

The **Spool** area holds transient copies of the data items (such as JPFs and PPFs). User requests for data are handled by specific server tasks (such as the JPF and PPF servers). The spool area is the data source for these servers and the servers can request data to be recalled from tape if it is not currently available in the spool.

Data is removed from the spool on a Least Recently Used basis to ensure a suitable amount of free space is available for newly created or recalled data. Typical data recall times from tape are around 45 seconds (30 seconds to find and mount the tape, up to 12 seconds to position to the required file and the remainder to transfer the data to disk).

The spool is split into three distinct zones. Each new data item is built in the **incoming** zone and only when complete is it 'moved' into the **active** zone where it is available for user access. Similarly, an **outgoing** zone is used by a cleanup process to remove items from the active zone. This is important when the data item is composite (e.g. a directory tree).

When new data arrives in the spool area, three tape copies are made within an hour. Two of these copies are removed from the tape silo each night and stored in separate locations. The third copy remains in the tape silo to repopulate the spool area as required to satisfy user data requests.

Data in the spool area is not deletable until the three tape copies *and* an overnight backup of the warehouse index have been made. This means that, barring catastrophic failure, the security of the data is ensured.

Locally developed software is used to move data from the systems where the data is created, to manage the warehouse disk areas and to request the tape copies. Veritas NetBackup software is used to migrate data between the warehouse spool area and the tape silo under the direction of the locally developed software.

2. THE JPF DATA SERVER

The JPF server provides access to JPF data for client applications on Linux, Windows NT and Solaris. JPFs contain raw data from JET experiments. They are generated on big-endian computers and the data files are accessed via a subroutine library. This library was only available on Solaris Sparc systems and on the IBM mainframe. It was not possible, or desirable, to port this library to other platforms.

Instead, a thin-client version of the library was produced with the same calling parameter sequences as the real version of the library. The client calls are forwarded to the server, where they are executed and the results returned to the caller. Byte swapping, where needed, is done in the client library.

This approach has distinct advantages: the endian problem is eliminated; the client library is very small; bug fixes in the native access library do not have to be propagated to all client machines.

Clients are completely de-coupled from the underlying data storage. This, in itself, has advantages. It is not necessary to have all the data on-line: data can be recalled rapidly on demand, with 'popular' data being more likely to be on-line.

Logging of data access is possible, and data access restrictions and quotas can be implemented if required. Rogue clients can be easily identified. The raw JPF data could be re-organised/stored in another format, such as a relational database or an MDSplus tree, with no impact on the data clients.

The JPF server has been in use for nearly 12 months and has proved to be very reliable, with no reported performance problems.

3. THE PPF SYSTEM

The PPF system has been in use since the start of the JET project for the storage and management of processed data. Around 2.5 million PPF files, containing about 1TByte of data in total, are available. The system allows access to multiple versions of data for each JET pulse, with the most recent version being the default.

3.1 COMPONENTS OF THE PPF SYSTEM

The old PPF system consisted of a number of components, which were designed and developed at JET. At an early stage, it was decided that, where possible, commonly used third-party solutions should be used as components of the new PPF system, rather than custom-developed software.

PPF files

These contain the bulk of the processed data. Each contains header information and data, x and time vectors for a number of signals. In the old system, these were direct access binary files. The files were stored on the IBM mainframe.

In the new system, the direct-access binary files were replaced by a third party file format with its own low-level file access routines. After consideration of various formats, netCDF, developed by UCAR/ Unidata[4], was chosen. The netCDF format matched closely the requirements for the

PPF files, and expertise in this format was available at JET. The files are stored in and managed by the JET Data Warehouse.

PPF Directory

All user access (read and write) to the PPF files is done via the PPF directory, which forms an index to the data. The directory maps requests for data made by pulse number and signal name to the physical location of the data. Also, directory services can list information such as signals that exist for a particular shot. In the old system, the Directory consisted of a set of direct access binary files. For the new system, the PPF Directory was replaced by a commercial RDBMS. Mimer [5] was chosen, as it was already widely used and supported at JET and provided all the required functionality.

User-callable routines

Around 50 functions are called from user-written codes to read data from and write data to the PPF system and interrogate the Directory. In the old system, this library was written in FORTRAN. For the new system, new user-callable routines were written in ANSI C. The API exhibits an identical interface to the old PPF library.

Low-level routines

Low-level routines form the interface between the user-callable routines and the PPF Directory and PPF files. In the old system, these routines were written in IBM Assembler. In the new system, this access is performed using embedded SQL to access the directory, and netCDF library calls to access the PPF files.

Client/server system

The PPF system allows users to read and write PPFs from Linux, Windows NT and Solaris. The old PPF server ran on the IBM mainframe, with FORTRAN clients running on other platforms. Data was transferred in IBM format, and converted to the local format by the client. The new PPF server runs on Solaris, and accesses the JET Data Warehouse. A client has been developed in C, which runs on Linux, Windows NT and Solaris. Platform independent data transfer is performed using XDR (External Data Representation) [6].

3.2 INTERACTION BETWEEN COMPONENTS OF THE PPF SYSTEM

The interaction between the components of the PPF system is illustrated in Fig 1.

4. DEVELOPMENT OF THE NEW PPF SYSTEM

4.1 DATABASE DESIGN

The structure and use of the old PPF Directory was reverse-engineered into a form suitable for implementation via an RDBMS.

The resulting database allows queries to be executed which are equivalent to the old directory software. For example, list signals exist for a particular JET pulse, or list PPFs for a given pulse range created by a given user.

4.2 FILE CONVERSION

As part of the migration to the new system, every PPF file had to be converted to a new format. Following design of the format of the netCDF file corresponding to a PPF file, a program to convert each PPF to a netCDF file was written. A second code was written independently by a different developer to read each PPF file and each netCDF file to verify the data was identical.

As part of the conversion process, data to be stored in the Mimer database was written to ASCII files for later loading. This allowed the bulk data transfer to be started while the database was still under development.

4.3 CLIENT/SERVER SOFTWARE

The PPF server was designed to be easily scalable and to keep different user sessions independent. A master server process runs, and spawns a new child process for each new client. The child process either times out or is terminated by a request from the client.

All state information is stored in a command block, which is passed between the server and the client. This means that a client process can cope with the termination of the server process, or a reboot of the data server. This is necessary since some user programs may run for days at a time.

Communication between the PPF server and the clients is implemented using XDR Record Streams. External Data Representation (XDR) is a standard that describes arbitrary data structures in a machine-independent fashion. XDR Record Streams implement bi-directional transmission of an arbitrarily long sequence of XDR records via a TCP/IP socket.

4.4 USER-CALLABLE ROUTINES

The fundamental requirement was that the new PPF system exhibit the same interface to users as the old system. Although the new routines are in C, they export a FORTRAN interface (e.g. hidden length arguments for strings).

5. TYPICAL USE OF THE PPF SYSTEM

Users access the PPF system either from codes they have written themselves, or from general-purpose utilities such as JETDSP, the JET Display Program. Typical transactions to read and write data are illustrated in Fig.2.

6. CURRENT STATUS AND FUTURE PLANS

The new PPF system went live on 20th June 2001. Most users were able to run their existing codes against the new system without relinking.

The current system is viewed as a basis for future development, and we aim to add new functionality as dictated by the needs of the physicists.

Possible requirements identified so far are: support for different data formats (the old system only supported integer and floating point data), and support for functions of more than two independent variables.

CONCLUSIONS

The IBM mainframe was removed on 20 th June 2001, and the new systems described in this paper are now in daily use. No statistics are yet available for the performance of the new systems during a JET experimental campaign.

The design and development of these new systems was completed within 18 months. This was done at the same time as supporting all of the existing systems on site, and supporting the needs of physicists during operations.

The pragmatic approach of using a combination of third-party and in-house solutions allowed us to complete the project on schedule while still supporting the computing needs of JET.

ACKNOWLEDGEMENTS

For their contributions to the design and development of these systems:

PPF System: Nicholas Cook, Mark Edwards, Anvita Matilal, David Robson and Ian Sall (EURATOM/UKAEA Fusion Association, Culham, UK) and Annedore Buhler (MPI für Plasmaphysik, EURATOM Association, D-85748 Garching, Germany).

JPF Server and Data Warehouse: Adrian Capel, Keith Norman, Mark Rainford, and Jonathan Todd (EURATOM/UKAEA Fusion Association, Culham, UK).

Other members of the CODAS Unit and the Database Management and Software Development Group of EURATOM/UKAEA Fusion Association, Culham, and contributors to the EFDA-JET workprogramme.1

REFERENCES

[1]. J.P.Christiansen, “Integrated Analysis of Data from JET”, Journal of Computational Physics **73** 85 (1987)

[2]. K.Blackler, “Remote Access to JET Data and Computers”, 2nd IAEA Technical Control Meeting on Control, Data Acquisition and Remote Participation on Fusion Research, Lisbon, July 1999.

[3]. MDSplus: http://www.psfc.mit.edu/mdsplus

[4]. NetCDF: http://www.unidata.ucar.edu/packages/netcdf/

[5]. Mimer: http://www.mimer.com

[6]. XDR: http://rfc1014.x42.com

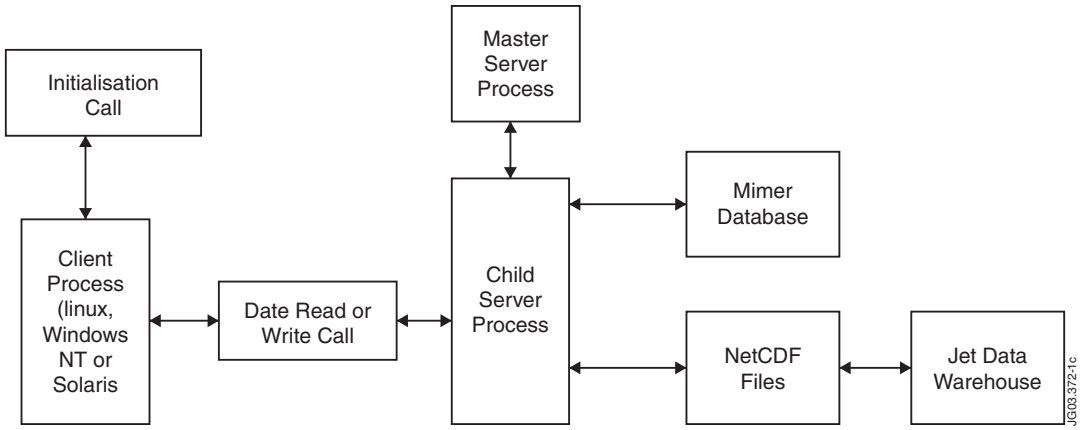

Figure 1: Interaction between components of the PPF system.

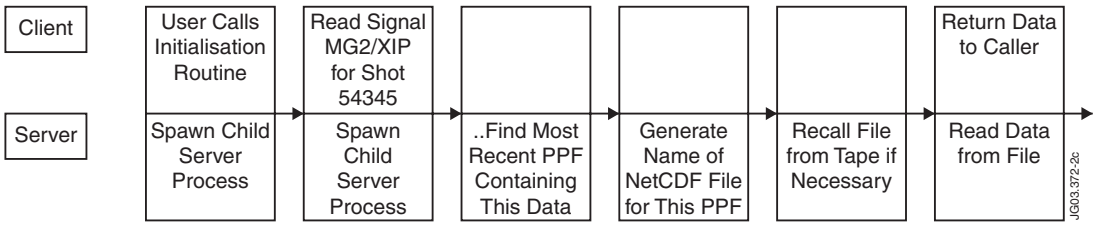

Figure 2: Typical read transaction

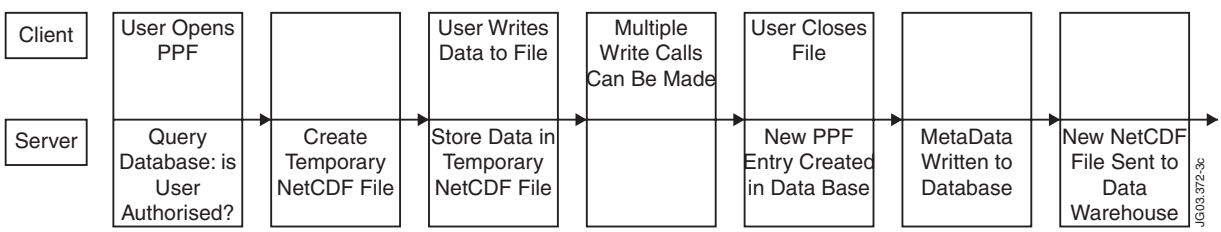

Figure 3: Typical write transaction