ArchiveDB – scientific and technical data archive for Wendelstein 7-X

Christine Hennig [1], Josef Maier[2], Martin Grün[3], Jon Krom[1], Torsten Blum[1], Michael Grahl[1], Peter Heimann[2], Heike Riemann[1], Heike Laqua[1], Marc Lewerentz[1], Anett Spring[1], Andreas Werner[1]

[1]Max Planck Institute for Plasma Physics, Wendelsteinstraße 1, 17491 Greifswald, Germany
Christine.Hennig@ipp.mpg.de, Jon.Krom@ipp.mpg.de, Torsten.Bluhm@ipp.mpg.de, Michael.Grahl@ipp.mpg.dem, Heike.Riemann@ipp.mpg.de, Heike.Laqua@ipp.mpg.de, Marc.Lewerentz@ipp.mpg.de, Anett.Spring@ipp.mpg.de, Andreas.Werner@ipp.mpg.de
[2]Max Planck Institute for Plasma Physics, Boltzmannstraße 2, 85748 Garching, Germany
Josef.Maier@ipp.mpg.de, Peter.Heimann@ipp.mpg.de
[3]FERCHAU Engineering GmbH, Steinmüllerallee 2, 51643 Gummersbach, Germany
Martin.Gruen@ipp.mpg.de

ArchiveDB is the data archive for all scientific and technical data collected at the Wendelstein 7-X project. It is a distributed system allowing continuous data archival. ArchiveDB has demanding requirements regarding performance efficiency (storage performance of 30 GB/s during experiments, expected storage amount of 1.4 PB/year), reliability (availability of 364 days/year), maintainability (testability) and portability (including change of hardware and software).

Data acquisition with continuous operation and high time resolutions (up to nanoseconds scale) for physics data is supported as well as long-term recording up to 24hour/7days for operational data (~1Hz rate). Moreover, all results of data analysis are stored in the archive. Another challenge, uniform retrieval of measured and analyzed data, allowing time and structure information as selection criteria, is mastered as well.

The key concepts of data storage and retrieval are: (1) partitioning of incoming data in groups and stream, (2) chunking of data in boxes of manageable size covering a finite time period, and (3) indexing of data using absolute time as ordering and indexing criteria.

Continuous operation of the ArchiveDB software and hardware for various systems and components relevant to Wendelstein 7-X has been done successfully for several years, thus, showing that the key requirements are satisfied. The overall data amount so far has reached 7 Terabyte over 9 years of data taking. Round-the-clock operation of the archive is in place since 5 years. Initial plasma operation OP1.1 of Wendelstein 7-X has been supported with no downtime during the whole experimental campaign.

The paper describes the software engineering concepts that have been used, consolidated and refined over the years of continuing productive ArchiveDB use and development. Changes in the underlying techniques, e.g. a change of the data store, have been encapsulated via an Application Programming Interface (API). This API unifies different implementations and is also suitable for data migration.

Keywords: Continuous Operation; Data Archive; Data Stream Management System; Big Data; Lambda Architecture; CoDaC; Java

1. Introduction

1.1 The Wendelstein 7-X fusion device and the CoDaC system

Wendelstein 7-X (W7-X) is a large fusion experiment of the stellarator type. It is constructed at Max Planck Institute for Plasma Physics in Greifswald, Germany [1]. The construction phase has lasted more than a decade. Finally, the machine has been commissioned and started its initial operation phase in 2015 [2].

W7-X has planned an extensive set of physics diagnostics including actuators and sensors positioned around the machine at nearly 300 locations.

Simultaneously with the machine construction, the software for control, data acquisition and communication (CoDaC) was developed.

Experimentation of W7-X is expected to produce huge amounts of data. These are mainly measured data from physics diagnostics but also results of data analysis, operational data of the machine parameters and the experiment control and execution parameters.

In order to ensure comparability of data from different sources, a dedicated high resolution timing network providing central time has been developed [3]. During data acquisition all sensor data are combined with the corresponding time data, thus having time stamped sensor and process data. All data is archived with timestamps in the W7-X ArchiveDB data store.

Data must be preserved over the lifetime of W7-X. An in depth description of the challenges of W7-X CoDaC can be found in [4].

1.2 Key requirements for continuous operation and its implications

The Wendelstein 7-X experiment is capable of continuous operation which implies a demanding novel requirement for the data acquisition of a fusion device. All CoDaC software is fully capable of continuous operation, i.e. for the intended 30 minutes of operation. During operation, a data amount of 30 GB/sec is expected. Moreover, a wide range of end systems is running on a round the clock basis. Expected overall data rates are in the range of 1.4 PB/year. These data centric challenges make ArchiveDB a "big data" installation.

It is a paradigm change from "shot and collect" (traditional shot based fusion experiments) towards "view and react". This implies a novel requirement of "read immediately after write" to the data archive as well as the overall CoDaC system. For an early proposal of the data acquisition see [5] and for more details regarding the archival architecture have a look at [6] and [7].

Figure 1 shows how the participating components interact with each other: CoDaStation is responsible for data acquisition and may perform simple online analysis during acquisition [5]. ArchiveDB is the software described within this paper. Offline data analysis is able to retrieve data from the archive and write results back to the archive. A data import mechanism allows storage of analyzed data. Online Data Monitoring [6] is used to visualize reduced data during experiments independent of archival using strip chart plots. Data Browser [7] allows to display archived data by user selection.

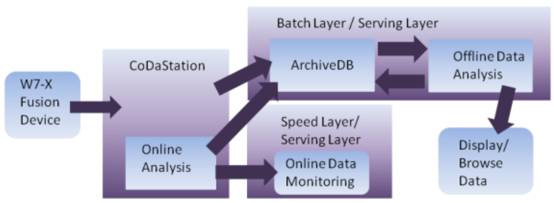

Fig. 1: The lambda architecture pattern applied to W7-X CoDaC software. Arrows show data flow (a) during archival (from and to CoDaStation and the Speed Layer) resp. (b) at arbitrary time after archival (Offline Analysis and Data Browser)

Manifold quality requirements regarding performance efficiency (e.g. read performance and latency), reliability (e.g. high availability), maintainability (e.g. testability) and portability (e.g. adaptability) round up the requirements elicitation.

2. Related work

There is a huge amount of development in this area. Gray et al tried to predict scientific data management in the coming decade [8]. The Computing Research Association (CRA) summarized challenges and opportunities for big data [9]. Aggarwal wrote a complete book about managing and mining sensor data [10]. PlanetData defined a big data curriculum [11].

Technical papers with interesting techniques targeting similar requirements are e.g.

Sector [12] and UDT (a high performance data transport protocol which is used in PowerFolder) [13] and try to overcome existing limitations and bottlenecks of established technologies. On the opposite end there are systems that completely use established technologies in order to provide a system for storing and retrieving huge amounts of distributed data e.g. dcache [14].

Some papers focus on query questions e.g. Nehme et al suggest a real time stream tagging framework that allows for querying data [15]. The druid framework targets similar problems [16]. Kraemer works in the field of queries over data streams [17].

2.1 Existing fusion archives and CoDaC systems

Large fusion devices usually come with their own control, data acquisition and archival software. Examples are JET [18] that has the ability to store continuous data in pulse files and "long pulse" machines, LHD [19], ToreSupra [20] and its upgrade WEST, and ITER.

Many of the smaller fusion devices use mdsplus [21] as software for control and data acquisition.

Promising future proof archival concepts can be found at ITER [22]. KIT has a Large Scale Data Facility [23], [24], [25] currently based on iRODS datagrid [26] and Hadoop [27] and is no longer restricted to the fusion community.

2.2 Other research dealing with sensor and mass data

Taking a broader look at storage of mass sensor data there are more systems in high energy physics, mainly accelerators (e.g. LHC), astrophysics, climate research (e.g. World Data Center Climate) and other areas. Advanced data management systems have in common an explicit and standard data access layer with precise Meta data, see the Open Archival Information System standard for a description [28].

The Extremely Large Data Base Community [29] tries to identify trends and commonalities in extreme scale data sets. The community covers a wide range of domains: astronomy, biology, finances, geosciences, healthcare, oceanography, oil & gas, physics, web analytics, seismology and is no longer limited to science and research.

CERN conducted a databases future workshop in 2011 [30]. The HDF group [31] is providing technologies and supporting services that make the management of large and complex data collections possible. SciDB [32], [33] [34] started as open source collaboration for science data a couple of years ago and is currently supported by paradigm4 [35].

2.3 Big Data in industry

At the moment there is hype about "Big Data" stating it to be the "new oil". But how can big data be defined and categorized? Big Data is characterized by the "Big Five": Volume, Velocity, Variety, Veracity, and Value. Some Modern Big Data Systems tend to use the lambda architecture pattern first introduced by Nathan Marz [36], see also Figure 1. Many technologies are emerging in this field: Data Stream Management Systems with Complex Event Processing, the world of NoSQL, MapReduce and Hadoop, niche techniques as SciDB and SciQL, HDF5, and many more, see e.g. [36] for a good executive summary and overview.

2.4. Why another development?

W7-X ArchiveDB has a history of nearly 15 years of development and thus could not use the recently emerged big data technologies from the very beginning. As shown in this chapter, no single one-solution-fits-all-purposes Big Data system is available out of the box.

The key concepts used for ArchiveDB have been proven robust, stable and suitable for the specialized purpose of storing W7-X relevant data. Thus, we decided not to change to any of the emerging technologies at the present state.

3. Key Concepts

Storage of continuous W7-X data is done using the following key concepts:

3.1. Partitioning of data in groups and streams

Data sources are usually sensors sampled using Control Station software [37], [38], [39]. The data sampled by an ADC is stored continuously in a single Archive Stream in the archive.

The stream is associated with another Archive Stream containing the parameter data that is describing both the sensors and the ADC characteristics and is stored whenever a parameter changes. This allows for interpretation of the stored ADC data.

Both Archive Streams are enriched with information about the Control Station itself which is stored in another couple of parameter-type Archive Stream. This information also contains when a station is started or stopped and in which status the station itself is. For a single Control Station this easily sums up to 20 to 1000 Archive Streams in the archive and is grouped together into an Archive Stream Group (see figure 2).

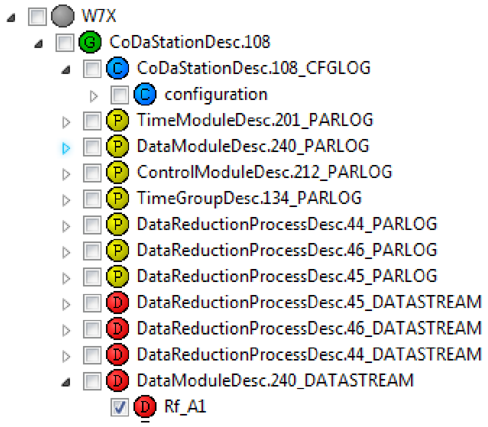

Fig. 2: Screenshot of archive data structures in a browser showing acquired data (D, red) plus related parameter and configuration (P, yellow and C, blue) grouped together in a group (G, green) for a Control Station (CoDaStation). The example shows data acquired for ECRH Gyrotron A1 operation: radio frequency Rf_A1 and related data.

3.2. Storage details of data including chunking and serialization

All archive data is stored as plain old java objects (POJOs) called boxes. This was introduced due to the fact that the initial archival system solution was based on an object database. The data is contained in POJOs and the POJOs are serialized for storage, see [40] for a discussion about java serialization.

Boxes files per stream and day contain the archive payload, i.e. the serialized POJOs. The boxes files are stored as binary random access files in the underlying file system. They are accompanied with an index file per boxes file.

The two different kinds of data to be stored are: (1) Mass data is usually arising continuously. (2) Structured data is usually arising at certain points in time. Structured data may be parameters, configurations and experiment execution information e.g. archived segment control operation [41].

3.2.1. Mass Data (data boxes) and Chunking

Mass data consist of a single primitive type. Sampled data, video type data, calculated analysis data is typical mass data and is usually acquired continuously. Continuous data is then partitioned into chunks of well defined time intervals. Storage is done in boxes of the respective primitive data type together with all timestamps (see figure 3). For every data type a special data boxes type is available. Section 7 gives some numbers regarding chunk size.

3.2.2. Structured Data (parameter boxes)

Structured data comes with plenty of different data types (e.g. Strings for names and descriptions, numeric types for settings). Parameter data is stored sporadically. Parameter boxes allow for storage of any structured data in a tree structure.

Parameter data must be furnished with the appropriate timestamps. Every parameter tree is stored in a single box. The connection between mass data and its related parameter data is guaranteed by a naming schema and the timing information. Parameters are stored at the beginning of a measurement and every time the content changes.

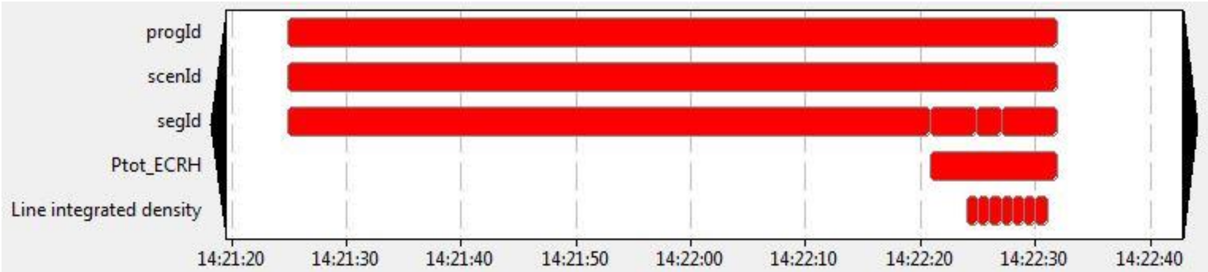

Fig. 3: screenshot of index data showing the arbitrary archive boxes indices and how to match between multiple archive streams (rows) using the absolute time (displayed at the x axis) acquired during the first hydrogen experiment. “progId”,” ScenId” and “SegId” show the archived time of program, segment and scenario execution whereas” Ptot_ECRH” shows availability of ECRH total power in the archive and “line integrated density” shows availability of density data.

3.3 Indexing of the data using absolute timestamps

As mentioned above, all data (mass data and parameter data) must be stored together with absolute W7-X timestamps. This allows for matching related data by time and makes the content of the archive comparable, see section 4.

Thus, the timestamps are heavily used as index criterion. Therefore it is reasonable to store the timestamps as index information (see figure 3).

This is done in a separated index file per Archive Stream and day. Every index file contains the first and last timestamp of all boxes together with the byte offset of the box in the boxes file. This allows fast read access to the data in question.

4. APIs and Queries

4.1 Application Programmer Interfaces (API)

The Stream Access API (low level, internal, data centric) has preserved the first ten years of operation of the archive development. It is a tightened API on top of the basic software allowing access to both the outdated object database and the file based archive, thus also allowing for easy data migration.

The Signal Access user API [42] (high level, object centric) eases user access to archived data. Users do not need to know details about the boxes concept. All data is represented as a Signal. The approach can be used both for mass data of arbitrary dimension and for parameter data (a parameter tree structure’s leaf is treated as a separate signal).

4.2 Reading archive data: API calls and Queries

All queries need two primary keys: one key is the selection of the Archive Stream (figure 2) whereas the other key is the absolute time (figure 3).

For comparison a SQL like query is given in listing 1 that can be applied e.g. to a Cassandra NoSQL database [43].

```
SELECT temperature
FROM temperate
WHERE weatherstation_id='123ABC'
AND event_Time > '2015-04-03 07:01:00'
AND event_Time < '2015-04-03 07:04:00';
```

Listing 1: A typical SQL-like time series query example in NoSQL cassandra

The equivalent query via the Stream Access API is shown in listing 2. It is a data centric, native archive operation. The example shows the time of first hydrogen plasma with ECRH power signal of Gyrotron A1. Note that in the low level query the data has CoDaC internal naming, see also Fig. 2 for understanding the example.

```
StreamAccess streamAccessObject =
StreamAccess.of(“W7X”, “CoDaStationDesc.108”,
“DataModuleDesc.240” );
final long t0 = W7Xtime( '2016-02-03 14:20:00' );
final long t1 = W7Xtime( '2016-02-03 14:22:00' );
streamAccessObject.getBox( TimeInterval.with
(t0, t1) );
```

Listing 2: The corresponding query in the low level ArchiveDB API in java-like pseudo code

The equivalent query via the Signal Access user API is shown in listing 3. Note that the signal is aliased to a better human understandable name. The resulting “Signal” object provides object oriented access methods to the data.

```
SignalAddress sigaddr = selectFromList( “W7X”,
“ECRH”,  “Total Power”, “Rf_A1”);
sigreader  =
SignalToolsFactory.makeSignalReader( sigaddr );
final TimeInterval ti = TimeInterval
      .with(W7Xtime ( ‘2016-02-03 14:20:00’ ),
            W7Xtime ( ‘2016-02-03 14:22:00’ ) );
 final Signal sig = sigreader.readSignal( ti,
ReadOptions.firstNSamples( 510 );
```

Listing 3: The corresponding query in the high level API Signal Access for users

4.3 Reading archive data: How to find relevant time periods

Data acquired outside experiments (machine observation, tests of ECRH etc.) is usually gathered by knowing relevant time intervals ahead or by listening/browsing when relevant data is available.

During Wendelstein 7-X experiment operation (pulse) it is also archived which programs, scenarios and segments are executed (see [44]) and an what time, thus, providing the users with relevant pulse timing and scheduling information, see Fig. 3 for an example screenshot. This is especially valuable for visitor scientists looking for some specific plasma data.

5. ArchiveDB: a decade of operation and a major change of underlying technology

This chapter describes the major changes undergone in both hardware and software while continuously providing ArchiveDB data read and write service.

5.1. Data history: a decade of successful operation

The ArchiveDB development started in 1998 and has been ongoing ever since. There is long experience using the streams and boxes concept. The first productive use was ECRH data from W7-X Gyrotron commissioning. Later the WEGA Stellarator had been operated for some years as fusion device and had been used as CoDaC test bed [45] including data storage in the archive.

Moreover, the Mistral Stray Radiation Test Facility has been completely operated by CoDaC Software and stored its data into ArchiveDB. Data from W7-X coil tests carried out at CEA Cadarache site had been imported into ArchiveDB and must be preserved as well.

5.2. Underlying technique: Major changes

5.2.1 Transition from object data base to plain files

The initial version used the object database ObjectivityDB [46] as object store for both data and parameter boxes plus time index data. The solution was productive from 2005 to 2014.

ObjectivityDB stores data in proprietary encoded files of arbitrary size in the file system. Access to the data is only possible using ObjectivityDB and its software layer, thus the data access needs a runtime license and is not freely available.

To overcome this situation the data store has been changed to plain binary files using the java serialization mechanism as explained in section 3.

5.2.2 Transition from schema based to schema less

ObjectivityDB maintains the class schema of boxes and index data as built-in feature. Schema migration is supported. However, the schema is rather rigid and needs 1:1 mapping of our business classes. This resulted in an inflexible mapping of the wide variety of parameter classes.

When dropping ObjectivityDB also the storage of the structured parameter data has been revised. The storage is now schema-less, i.e. each structured parameter item cover both its values and its self-contained schema (piggy-back).

5.2.3 Use of a distributed file system for horizontal scaling

For a file based archive solution the file system plays a major role in fulfilling the scalability requirements. In use is the distributed scalable file system General Parallel File System (GPFS) [47]. This allows distribution of data between multiple archive servers both for reading and writing.

For every Archive Stream a single writer instance writes on a dedicated server. Multiple readers can read from multiple servers. The data is distributed as files using the GPFS file system features and thus is available to all potential readers very shortly after storage.

The ArchiveDB solution must be capable of horizontal scaling (scale out: add more servers). This feature is guaranteed by using GPFS.

Connecting GPFS to the High Performance Storage System (HPSS) [48] via GHI [49] is in place. The use of GPFS and HPSS was decided due to the fact that it is already in place and should be used for data archival as well.

5.2.4 Data access during renewal and access to existing legacy data in Objectivity

All data in the archive has been accessible over the whole period of renewal and there was no outage period, thus allowing machine observation data to be archived without interruption. The Stream Access API is able to access both data archives. Migration software has been developed based on this API. The migration of existing data from ObjectivityDB into the new ArchiveDB system is ongoing. Once finished, the Objectivity based archive will be abandoned.

6. A note on data import

In order to import data that does not come with time stamps in-build (such as the import of external instrument test data) synthetic time stamps must be generated. For those diagnostics a trigger system has been provided. All triggers are measured and archived, thus allowing retrieval of the timestamps when the trigger occurred. These timestamps can be used as starting point to generate timestamps for the acquired data and store both data and time in the archive.

7. Data throughput during OP1.1

Wendelstein 7-X operation phase OP 1.1 with a limiter installed lasted from 10th of Dec

2015 to 10th of Mar 2016. Plasma temperatures reached 8 keV (electrons) and 2 keV (ions). Pulse lengths reached more than 5 seconds. Data acquisition sources reached 59 Control Stations, 4 archive write servers, and 12 read/write servers. About 300 reading clients accessed archived data randomly. Data upload from 120 diagnostic end systems has been managed successfully.

Typical data chunks are in the range between 40.000 bytes all 2 seconds (typical machine observation, 24*7) to 27.200.000 Bytes all 100 ms (typical high resolution multi channel ADC during experiment pulses only). Finding the proper data chunk size follows from experience and is subject to individual requirements and boundary conditions of the diagnostic end system. For small data amounts data is expected to reach the archive early minimizing possible data loss on the way to the archive, thus leading to small chunks. For large data amounts the acquisition machine and hardware must not be overloaded (large chunks) and the data in a chunk must fit into the physical memory (small chunks), see [37] and [5] for an in depth discussion. As a consequence, the optimum chunk size is determined by experimentation for the various categories of data sources.

Fig. 4 shows the total amount of data acquired during OP 1.1 with a visible holiday break during the Christmas season.

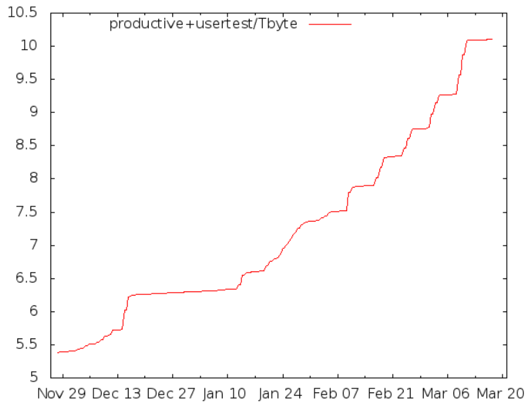

Fig. 4: Increase of data in ArchiveDB during Wendelstein7-X OP1.1

8. Lessons learned

The technology of storing data as objects together with timestamps has been used successfully throughout the years and proven to be a sustainable solution. It is indispensable to provide W7X time stamps for all data that need to be archived.

ArchiveDB stores all measured data together with describing parameter information. Moreover, all experiment control and execution parameters are archived as well. Parameter information is subject to change over time. ArchiveDB supports this diversification by allowing for changes and evolution of the archived configuration schema. Archival of flexible tree structures of any structured data makes the archive schema less.

The archival access pattern follows the Single Writer/Multiple Reader pattern which is also implemented at ITERDB [22]. ArchiveDB is scalable by utilizing the scale out technique of the underlying file system GPFS. Writers can write to and readers can read from different servers transparently without explicitly noticing.

The paradigm change from shot and collect to view and react is a major challenge for some fusion community users. Time based queries make data retrieval of arbitrary time intervals feasible. Users that are used to use shot numbers to identify their data expect a single shot to fit in a single transaction and in local memory. This is no longer guaranteed. For larger data sets users have to split large queries into smaller ones by looping over smaller time intervals.

9. Conclusions

The paper describes continuous archival of time stamped sensor data and all related parameter data for a scientific fusion experiment. Its main technique consists of chunking the data in boxes with timestamps. Timestamps are used as primary index over the data. Query examples are provided.

The ArchiveDB solution proved to be powerful, robust and long term stable. Changes of underlying storage techniques have already been mastered once (invisible to users) and thus become established and integral part of the solution.

Nearly a decade of operation with multiple and diverse data sources shows the success of the data archival. The system can well compete as niche technique with emerging general purpose big data technologies for the special use case.

Continuous data archival round the clock is in place with W7-X machine data accompanied by archival of commissioning data.

Acknowledgements

The authors like to thank Georg Kühner for providing the architectural concepts used within this work.

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[1] H.-S. Bosch, V. Erckmann, R. W. T. König, F. Schauer, R. J. Stadler, A. Werner
Construction of Wendelstein 7-X – Engineering a Steady-State Stellarator
IEEE Transactions on Plasma Science 38 (3) 265-273 (2010)

[2] H.-S. Bosch, R. Brakel, M. Gasparotto, H. Grote, D. A. Hartmann, R. Herrmann, M. Nagel, D. Naujoks, M. Otte, K. Risse, T. Rummel, A. Werner
Transition from Construction to Operation Phase of the Wendelstein 7-X Stellarator
IEEE Transactions on Plasma Science 42 (3) 432-438 (2014)

[3] J. Schacht, H. Niedermeyer, H. Laqua, A. Spring, I. Müller, S. Pingel, et al.
Tasks and structure of the WENDELSTEIN 7-X control system
Fusion Engineering and Design, 81 (July (15–17)) (2006), pp. 1799–1806

[4] Werner, A. ; Bluhm, T. ; Grahl, M. ; Hennig, C. ; Kuhner, G. ; Krom, J.G. ; Laqua, H. ; Lewerentz, M. ; Maier, J. ; Riemann, H. ; Schacht, J. ; Spring, A. ; Svensson, J. ; Zilker, M.
Cutting edge concepts for control and data acquisition for Wendelstein 7-X
10.1109/SOFE.2013.6635430

[5] P. Heimann et al.
Status report on the development of the data acquisition system of Wendelstein7-X
Fusion Engineering and Design 71 (2004) 219–224

[6] Ch. Hennig, P Heimann, S Heinzel, H Kroiss, G Kühner, H. Kühntopf, J Maier, J Reetz, M Zilker
A concept of online monitoring for the Wendelstein 7-X experiment
Fusion Engineering and Design Volume 71, Issues 1–4, June 2004, Pages 107–110

[7] T. Bluhm, P. Heimann, Ch. Hennig, H. Kroiss, G. K¨uhner, J. Maier, H. Riemann, M. Zilker
Data access and its implementation at Wendelstein 7-X
Fusion Engineering and Design 83 (2008) 387–392

[8] http://arxiv.org/pdf/cs/0502008

[9] http://www.cra.org/ccc/files/docs/init/bigdatawhitepaper.pdf

[10] http://www.charuaggarwal.net/sensortoc.pdf

[11] http://planet-data.org/emc-data-science-and-big-data-analytics-curriculum

[12] http://sector.sf.net

[13] http://udt.sourceforge.net/

[14] http://www.dcache.org/manuals/dcache-whitepaper-light.pdf

[15] http://www.vldb.org/pvldb/2/vldb09-24.pdf

[16] http://static.druid.io/docs/druid.pdf

[17] http://archiv.ub.uni-marburg.de/diss/z2007/0671/

[18] R. Layne, A. Capel, N. Cook, M. Wheatley
Long term preservation of scientific data: Lessons from jet and other domains
Fusion Engineering and Design, Volume 87, Issue 12, December 2012, Pages 2209–2212

[19] H. Nakanishi, M. Ohsuna, M. Kojima, S. Imazu, M. Nonomura, M. Hasegawa, K. Nakamura, A. Higashijima, M. Yoshikawa, M. Emoto, T. Yamamoto, Y. Nagayama, K. Kawahata, LHD Experiment Group
Data Acquisition and Management System of LHD
Fusion Science and Technology, Volume 58, Number 1, July/August 2010, Pages 445-457

[20] B Guillerminet, M Le Luyer, B Rothan
Tore supra data acquisition: A system for long duration discharges
Fusion Engineering and Design, Volume 43, Issues 3–4, January 1999, Pages 259–264

[21] http://www.mdsplus.org/

[22] Abla, Heber, Schissel, Robinson, Abadie, Wallander, Flanagan
yITERDB-The Data Archiving system for ITER
Fusion Engineering and Design 89 (2014)

[23] http://www.scc.kit.edu/forschung/lsdf.php

[24] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5739022&tag=1

[25]
http://www.ipe.fzk.de/~stotzka/publications/publications/sutter2010.2.pdf

[26] http://irods.org/

[27] https://hadoop.apache.org/

[28]
http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=57284

[29] http://www.xldb.org/

[30] Databases future workshop
https://indico.cern.ch/event/130874/abstract-book.pdf

[31] http://www.hdfgroup.org/

[32] http://scidb.org/

[33] http://oai.cwi.nl/oai/asset/18678/18678A.pdf

[34] http://www.scidb.org/Documents/SciDB-CIDR2009.pdf

[35] http://www.paradigm4.com/

[36] Leitfaden Big-Data-Technologien - Wissen für Entscheider
http://www.bitkom.org/files/documents/BITKOM_Leitfaden_Big-Data-Technologien-Wissen_fuer_Entscheider_Febr_2014.pdf

[37] T. Bluhm, et al.
Wendelstein 7-X's CoDaStation: A modular application for scientific data acquisition
Fusion Engineering and Design, Volume 89, Issue 5, May 2014, Pages 658–662

[38] Laqua, H.; Niedermeyer H.; Schacht J Spring A.
Real Time Software for the Fusion Experiment WENDELSTEIN 7-X.
Fusion Engineering and Design 81 (2006), Issues 15-17, S. 1807-1811. ISSN 0920-3796

[39] Hennig, C.; Heimann, P.; Kroiss, H.; Kühner, G.; Kühntopf, H.; Maier, J.; Zilker, M.
Using the Network as Bus System for Long Discharge Data Acquisition and Processing
Fusion Engineering and Design, Volume 81, Issues 15–17, July 2006, Pages 1735–1740

[40] Joshua Bloch
Effective Java 2nd edition
2008

[41] A. Spring, H. Laqua, J. Schacht
User control interface for W7-X plasma operation
Fusion Engineering and Design, Volume 82, Issues 5–14, October 2007, Pages 1002–1007

[42] J.G. Krom
Some Ideas about a Storage-Technology-Independent data-access API for the W7X data-archive: The “Signal Based Access” API
lab report

[43] http://de.slideshare.net/patrickmcfadin/cassandra-20-and-timeseries

[44] Anett Spring, Heike Laqua, Helmut Niedermeyer
User interaction concept for plasma discharge control on WENDELSTEIN 7-X
Fusion Engineering and Design, Volume 81, Issues 15–17, July 2006, Pages 1957–1961

[45] J. Schacht, et al.
Stellarator WEGA as a test-bed for the WENDELSTEIN 7-X control system concepts
Fusion Engineering and Design, Volume 83 (2008), Pages 228-235.

[46] http://www.objectivity.com/

[47] Frank Schmuck and Roger Haskin
GPFS: a shared-disk File System for Large Computing Clusters
Proceedings of the 2002 Conference on File and Storage Technologies (FAST)

[48] http://hpss-collaboration.org/

[49] http://www.hpss-collaboration.org/hpss_for_gpfs.shtml