Real-time Data Streaming and Storing Structure for the LHD's Fusion Plasma Experiments

Nakanishi Hideya, Ohsuna Masaki, Kojima Mamoru, Imazu Setsuo, Nonomura Miki, Emoto Masahiko, Yoshida Masanobu, Iwata Chie, and Ida Katsumi

***Abstract*—The LHD data acquisition and archiving system, i.e., LABCOM system, has been fully equipped with high-speed real-time acquisition, streaming, and storage capabilities. To deal with more than 100 MB/s continuously generated data at each data acquisition (DAQ) node, DAQ tasks have been implemented as multitasking and multithreaded ones in which the shared memory plays the most important role for inter-process fast and massive data handling. By introducing a 10-second time chunk named "subshot," endless data streams can be stored into a consecutive series of fixed length data blocks so that they will soon become readable by other processes even while the write process is continuing. Real-time device and environmental monitoring are also implemented in the same way with further sparse resampling. The central data storage has been separated into two layers to be capable of receiving multiple 100 MB/s inflows in parallel. For the frontend layer, high-speed SSD arrays are used as the GlusterFS distributed filesystem which can provide max. 2 GB/s throughput. Those design optimizations would be informative for implementing the next-generation data archiving system in big physics, such as ITER.**

***Index Terms*—LHD, steady state, LABCOM system, real-time streaming, GlusterFS.**

I. Introduction

The LHD (Large Helical Device) is a superconducting fusion experimental device which enables steady state plasma sustainment [1]. It is known in fusion research as the first pioneer that started genuine steady state fusion plasma experiments. Its data system, called *"LABCOM system,"* has been fully equipped with high-throughput real-time (RT) data acquisition (DAQ), streaming, and archiving capabilities from many digitizer frontends to the data consumers on the network [2], [3].

In the 17th annual campaign, the LHD has renewed the world record of acquired data amount in fusion experiments from 328.5 GB (2012) to 891.6 GB (2013) by a 48 min. long steady state plasma sustainment. In the short pulse operation whose duration is less than 10 s, it produces about 23 GB/pulse raw data in every 3 min. cyclic operation. It consequently has about 180 pulses per day in the short pulse operation.

Figure 1 shows the tendency of the LHD's increasing number of DAQs and also the raw data growth for each pulse. It clearly shows that the number of DAQ nodes continues increasing linearly. However, the data amount continues growing exponentially. The data growth observed in the LHD fits Moore's law, doubled in 18 months [4], very well. As the LHD started operation more than 15 years ago, we have many legacy CAMAC digitizers which are not capable of fast RT operation. We continue using them up to the present, however, RT-operable modern digitizers such as NI PXI and Yokogawa WE7x have become very popular and recently occupy more than 2/3 of all the DAQ nodes. The data growth in the past 10 years has been made mostly by them.

As easily found in Fig. 1, the introduction of RT-capable fast digitizers and the beginning of the LHD's steady state trials were well synchronized. The real steady state plasma operation has been started since the 7th annual campaign in 2003. Prior to the real operation, we started the research investigation, system design, and code development for a new RT DAQ system about three years before.

In the following sections of this paper, we will report the technical details that enable us to realize high performance real time data acquisition, storing, and redistribution system in LHD.

Manuscript received June 16, 2014; revised November 26, 2015.

Nakanishi H. and co-authors are with the National Institute for Fusion Science (NIFS), Toki 509-5292, Japan (http://www.nifs.ac.jp/).

Nakanishi H. and Ida K. are also with SOKENDAI (The Graduate University for Advanced Studies), Japan (http://www.soken.ac.jp/).

II. Steady State Operation in LHD

Helical systems like LHD can sustain current-free stable plasmas, and therefore never require any fast feedback control for plasma stabilization. As the LHD is also fully equipped with superconducting magnetic field coils, there is no need of transient pulse operation on them. This is why most of the actuators of LHD are slowly controlled in the order of 1 s^{-1}. High speed real time processing is only applied for the physics measurements, analyses and data visualizations especially during the long pulse, i.e., steady state plasma sustainment. One-hour long plasma sustainment was first achieved in the 9th annual campaign (2005), two years after the real trial started. At that time, 90 GB raw data were successfully acquired by our RT DAQs.

On the other hand, tokamak systems are known as they require fast feedback controls for sustaining the fusion plasmas. That is why most of the tokamaks have adopted the real-time processing systems based on the VMEbus hardware and the real-time operating system, such as VxWorks, LynxOS, or Linux/RT [5], [6]. Since the closed feedback loops typically require the latency time as about 10^{-3} s or less, the performance optimization had been primarily made not on their high data throughput but on their low latency ability. Therefore, the real-time feedback controls were ordinarily made on a limited number of measured signals, never on full size of raw data [7].

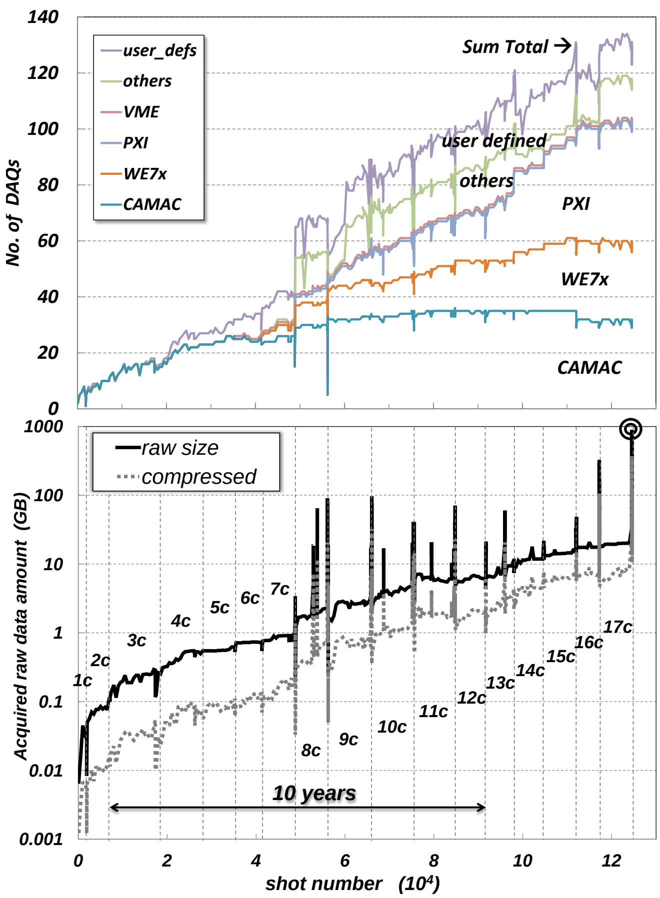


Fig. 1. DAQ trends in LHD: (Top) The total number of DAQ nodes grows very linearly in which real-time operable PXI and WE7x digitizers are increasing and the legacy camac never increase except in the very early stage. (Bottom) Acquired raw data for each shot continues growing exponentially. Each upper spike corresponds to a long-pulse trial held in LHD. A double circle shows the 891.6 GB/pulse world record. The growth rate fits Moore's law very well.

Being free from the fast feedback controls, the LHD system has been able to concentrate the performance optimization upon the highest data throughput. Actually, sufficiently big block sizes have been adopted to achieve almost ideal data transfer speed on each input/output (I/O) port, even though they may inevitably cause rather big time delays. They are in the range between 10 ms and 0.5 s so that in general they are too long for fast feedback controls. Of course, fast feedback controls will be possible if we choose the smallest block size corresponding to 10 ms or less latency. The intention for high throughputs is one of the most distinguished characteristics of the LHD's streaming DAQ and archiving system.

III. Fundamental System Design and Requirements

The LABCOM system is a distributed concurrent system in which every plasma measurement is controlled by its own acquisition computer simultaneously in parallel. It has been designed to be capable of scaling out the total performance linearly by increasing the number of DAQ nodes. It adopts a massively parallel processing (MPP) model which consists of a number of distributed DAQ nodes, data consumers, storage servers, and an index database which manages all the distributed data locations spread in many DAQ and storage nodes. The MPP architecture is significant for providing scalable I/O performance for large-scale DAQ systems. Actually, in the last annual campaign (2013), LHD operated about 110 distributed DAQ nodes for both short pulse and long pulse steady state experiments.

Generally in fusion plasma experiments, the primary requirements for modern RT DAQ and storage can be considered as follows:

1) Continuous DAQ at the same sampling rate of burst operation, typ. 1 MS/s/ch for fluctuations or 1 GS/s/ch for kinetic effects
2) Several tens or 100 channels for every 2-D/3-D measurement → 100–200 MB/s generated by each DAQ
3) 1 Gbps cameras, e.g., GigE Vision of VGA×100 fps full color, or cyclic operations of high-speed cameras, typ. 4 k×1 M fps, by using a buffer for several GBs
4) Performance scalability and topological extendability, for 100–1000 distributed nodes
5) High speed data receiving and storing at the data storage, accepting more than 100 incoming streams
6) RT data streaming service at the same rate of the data generation to the end consumers
7) RT redundant preservation or rapid replication for data safety.

In order to realize more than 100 MB/s RT DAQ, which was not necessarily easy at that time, we started an investigation of the RT-capable DAQ in 2000 and made a first decision to adopt the CompactPCI (cPCI) standard for the digitizer platform [8].

At present, 1 M fps high-speed cameras and 1 GS/s high-speed ADCs generate about 20% of overall physics data in LHD. However, the amount of those data has been growing very steadily in the recent several years. In the coming several years, it will become necessary to start the technical investigation for 1 GB/s RT DAQs.

Since modern PC has a good parallel processing capability, we could design each DAQ node to be "intelligent" and multifunctional. The expected functionalities can be categorized into seven independent processes: i) digitizer controller, ii) sequence executor, iii) local archiver, iv) data migrator, v) RT data streaming server, vi) bulk data server, and vii) commanding/monitoring agent. Table I and Fig. 2 show the schematic process list and view in a DAQ node. These service processes are implemented as multitasking and multithreaded programs to fully utilize the modern multicore cpu performance. However, the inter-process data passing mechanism would dominate the I/O performance in the case of dealing with massive data streams. The practical design adopted for the LABCOM system will be explained in the following subsection.

For the user interface of storing and retrieving their own data, the LABCOM system provides the common application programming interface (API) library named "*dbStore and Retrieve,*" respectively. They can work not only for various programming languages such as C/C++, Fortran, Python, Ruby, and Java but also for interactive data manipulation platforms such as IDL, PV-WAVE, LabVIEW, and MATLAB

TABLE I
I/O MANAGER PROCESSES IN DAQ NODE.

	functionality	I/O port	RT/batch	destination
i.	DAQ	digitizer link	RT	digitizers
ii.	(Main program)	(Shared RAM)	RT	(inter-process)
iii.	Local archiving	disk i/o	batch	local disk
iv.	Batch migration	network	batch	main storage
v.	RT streaming	network	RT	client PCs
vi.	Data transferring	network	batch	client PCs
vii.	DAQ agent	network	RT	manager

[9]. Although both raw and analyzed data can be handled through the APIs, some highly sophisticated results would be better handled in the relational database which is called as "*Kaiseki (data analysis) Server*" in the LHD system [10]. These APIs are commonly used for both short and long pulse data, but in the RT operation the dedicated interface will be necessary for sending and receiving the endless streams. The details will be explained in the next section.

Agent-oriented method will be another key technology to keep watching the healthiness of every distributed node's behavior. Commanding so many nodes may also need a sophisticated mechanism to handle various orders. The commanding/monitoring agent program is also implemented in DAQ node for those roles under the *Manager–Agents* distributed system model.

IV. Real Time Data Handling

A. RT Data Acquisition

When we started the fundamental system design for the RT data system, VME-based digitizers were very popular and often used in fast feedback controls for plasma stabilization purposes. Because they need a small latency time for closed-loop feedback controls, they usually implemented a small size of fifo buffers to send out the digitized data samples. However, the effective bandwidth of VMEbus was about 30 MB/s, which was not sufficient for our design requirements. Thus, we had to adopt another digitizer platform to realize a 100 MB/s capable new DAQ frontend.

After some technical investigations, we decided to adopt NI PXI digitizers first and later PXI-Express (PXIe), which are fully compatible with cPCI and Compact-PCI Express (cPCIe) standards, respectively. They are capable of streaming out at maximum 110 MB/s and 3.2 GB/s consecutive data blocks continuously [11].

Table I shows the list of I/O ports through which acquired data streams come and go inside DAQ PC. As every peripheral device has the most appropriate I/O manner, such as data block size or timing intervals of iteration, the port controlling application should be optimized correspondingly to each one for providing the best I/O performance. Because of this reason, we have implemented the multitasking structure where each I/O port is managed by its own dedicated program and there is an interchange memory area shared by related processes. As previously mentioned, all the manager processes are commanded and monitored through the agent process running on each DAQ node. Fig. 2 shows the schematic diagram.

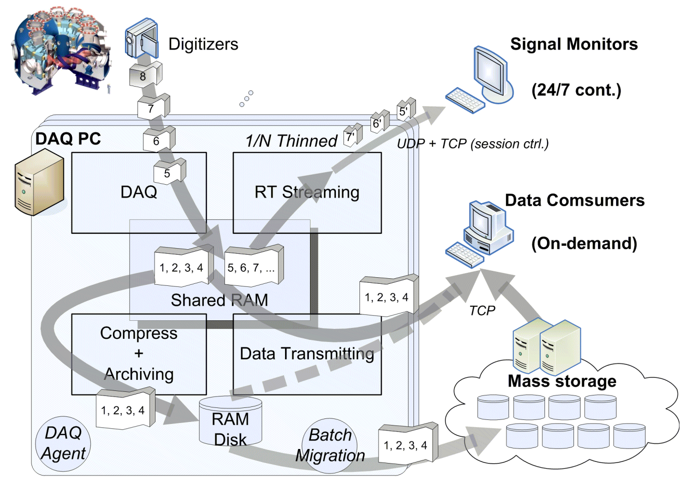


Fig. 2. Inside diagram of real-time DAQ: Consecutive incoming blocks from the digitizer are once stored on the shared volatile memory (RAM), and when the whole subshot is accumulated it will be written into a persistent subshot file on local HDD. Another independent task will migrate the subshot file into the outer mass-storage area. On-demand data clients may possibly retrieve the last subshot data from RAM area when its local storing is not completed [12], [13]. In archiving and transferring the data blocks are handled in 10 s 'subshot' size.

Generally speaking, processing latency time (second) from data input to output is an opposite property to data throughput (byte/second) on many RT processing systems. For making maximum use of physical bandwidth of I/O link, larger blocking size is better for reducing the process overheads. However, this makes the latency longer and loses the real time property.

As the LABCOM system strongly intended to achieve the best I/O performance for the growing amount of massive data, we prefer to adopt a big block size large enough to obtain an optimal speed on every I/O link. Of course, different block sizes must be chosen for the respective device ports.

Raw data blocks will be sent into the dedicated shared RAM space inside DAQ PC. This acquisition block size is usually given by the digitizer's driver software. The incoming data stream will be delivered further into different routes for local storing, RT streaming, and batch transferring. The size of each segmented time chunk is defined as the same as the maximum duration time of short pulse plasma experiments, i.e., 10 seconds. For the fail-proof capability against any unusual delays of data storing or network transferring, the intermediate shared memory is doubly spared so that the system can survive for some while even in the state of delayed processing.

B. 24/7 Continuous Slow Acquisition

In case of fusion experiments, we can practically assume that every plasma discharge has the end. However, environmental radiation and device status monitorings need 24/7 continuous operation which should never be stopped at any moment. On the other hand, being different from physics measurements, they never need faster sampling rates than 1 kS/s in most cases. Such the slow sampling may generate small and numerous time chunks if 10 second block size for physics measurements were applied to them.

TABLE II
NONSTOP 24/7 MONITORING FOR DEVICE AND ENVIRONMENT IN LHD [14].

	rate	samples/period	/day	channels
CDP	10^3 s^{-1}	600 000/10 min.	144	512
RMSAFE	0.2 s^{-1}	120/10 min.	144	16

Fine sampling raw data of 1 kS/s/ch are important for the case that some meaningful events can be found in their signals. However, slow monitoring data are rather useful in exploring long-term trend graphs over days, weeks, or months. Thus, we have modified the DAQ system to cope with the variable chunk size so that 10 minutes of 1 day long time chunks can be generated by 1 S/s/ch resampling with an arithmetic integral filter. Table II shows the list of slow monitorings running in LHD [14].

Due to the sampling theorem, the raw data must be filtered below the half frequency of the resampling frequency beforehand [15]. Environmental radiation monitoring, however, may need peak hold signals rather than averaged out signals. To be flexible for a variety of resampling algorithms, the computation is independently made on another PC receiving the data stream from the DAQ PC.

C. RT Data Streaming

DAQ PCs must accept any client connections and send the data stream continuously. For reducing the server burden for sending massive data, RT streams are sent on light weight UDP protocol. A TCP connection is used in parallel for controlling the network session, i.e., negotiation commanding [3], [13]. Such the combination use of UDP and TCP is popular among RT-streaming protocols such as RTSP with RTP [16]. This protocol optimization enables us to continuously send the whole 100 MB/s raw data stream with a reasonably light load on the server. Like other streaming protocols, however, it is not implemented with any packet loss detection and recovery mechanizm so that the RT applications must be coded with taking the potential packet loss into consideration.

UDP-based massive data sending does not place a heavy strain on server. However, receiving massive data is not necessarily a light task for some client PCs of low performance. In order to guarantee that any client PCs can receive the streams without any loss, sending 1 sample out of N (1/N) data thinning has been implemented on the server-side RT-streaming program (see Fig. 2). The thinning ratio can be set by the client request at the negotiation phase in establishing each streaming session. Different from the slow continuous monitoring, this communication does not support the pre-filter before the thinning due to avoiding any heavy load in the server-side processing.

In the case of 2-D image data, of course, thinning is done based not on samples but on frames. All the low-level data management are performed inside the dedicated client API communicating with "rt-transd" server. In the same way as the batch data handling, we provide the multilingual client "rt-retrieve" API and also the wrapper for the integrated data manipulation platforms, all of which can be used on Linux, MS Windows, and MacOS/X operating systems.

D. Continuous Data Archiving

To store endless data streams having indefinite duration time, we have introduced the idea of "subshot" that cuts the whole length into a series of 10 second time chunks [12], [17]. The writer process only keeps the latest chunk opened for writing data and closes all the prior ones. Hence, any other processes can read them even while the writing process is proceeding. We give a sequential extension number to the shot number with a manner of **"shot.subshot"** such as #100000.1, #100000.2, #100000.3, ... for every 10-second chunk of long pulse data. Since the "zip" archiving format is used to store the data in this system, their filenames will be something like "Diagname-100000-1.zip," "Diagname-100000-2.zip," ..., respectively. All the channel data of the same time segment are archived in one zip file.

We first decided the 10-second "subshot" rule to be appropriate for the duration of short pulse experiments in LHD. As mentioned in Section IV-B, however, the fixed length subshot is not necessarily suitable for slow monitoring. In addition, very high speed digitizers such as the 12.5 GS/s/ch oscilloscope and the 1 Mfps digital camera have rapidly become widespread even in fusion plasma diagnostics. They output several GB raw data for a few second pulse plasma, so that the users often prefer much smaller time chunks for analyzing computation on their own PC memory. Therefore, we have modified the original design of fixed length subshot to be a variable length fit for any cases.

Zlib and *JPEG-LS* lossless compression algorithms already have been embedded in our batch migration process. There are many high speed RT-encoding hardware, such as *MP3* and *MPEG*. However, they are mostly "lossy" algorithms that could not be applied for physics diagnostic data. We have already verified that *zlib* and *JPEG-LS* routines provide better compression ratios on 1-D and 2-D data, respectively, compared to other methods [13]. However, they require some definite periods of processing time to achieve good ratios, and hence are not applicable to RT processing of massive sized data.

Light weight arithmetical calculations such as *Run-length encoding (RLE)* can perform RT compression [18]. However, they usually do not provide better compression ratio than roughly 1/3 of *zlib* and *JPEG-LS* cases. Further investigation is desired on this problem in the near future.

E. RT Monitoring and Commanding Console

Especially in steady state operation, RT progress of all DAQ nodes must be monitored for avoiding any unnoticed long stop. The status monitoring "agent" is running on each DAQ node to report the internal progress through the IP multicast protocol to the network. The central "manager" gathers all the reports from the distributed agents by listening to the IP multicast, and displays them on the Web server in real time. In the LABCOM system, we show all the DAQ channel statuses and refresh them every second. As Fig. 3 shows, one DAQ

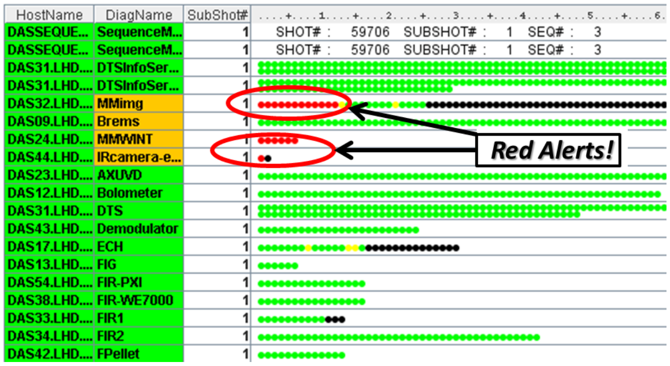


Fig. 3. DAQ monitoring web console: Each channel status of many DAQs are shown as the color of the line formed bullets. When some trouble happened, the suffered nodes would be automatically placed at the top of this list and also indicates the "red alert" of the fault channels by color. Since the refresh interval is 1 s, it can help the DAQ monitoring operator notice very rapidly some abnormal conditions in acquiring the channel data or on the DAQ node. When the fault situation dissolved, the red alert will turn to be green colored like the lower normal nodes [12].

status corresponds to one line in which each channel status is shown as one colored bullet. The difference of color shows the channel status as Black = Ready, Green = Completed normally, Yellow = Acquisition under way, and Red = Some error happened on that channel.

In addition, control commands, such as start, stop, shutdown, reboot, change operation mode, etc., can also be sent from the Web console to the DAQ agent, individually or simultaneously to all. This web monitoring/commanding console helps us find any DAQ or archiving trouble almost within a second or two, and then enables us to execute the recovery commands within about 10 seconds.

V. Optimization of Storage Structure

As mentioned in the last section, the number of 100 MB/s or higher bandwidth RT DAQs has been gradually increasing in recent few years. The severest requirements in the whole DAQ and archiving system will confront not the distributed DAQ but the data importing part of the central primary storage because many massive streams rush in during a long plasma discharge.

Until the last experimental campaign in 2012, we had used a cloud-based storage in LHD. It was able to perform load balancing among member nodes, however, the I/O throughput is basically limited by the unit RAID performance of a few 100 MB/s. It must be upgraded at least one digit higher to receive multiple 100 MB/s streams, thus we have decided to introduce a new SSD-based fast frontend storage on top of the bulk archive. A "distributed replicated" GlusterFS volume can provide data redundancy with a simple structure for scale-out ability [19]. In case of disk trouble, this will maintain the normal performance and never cause a long and heavy I/O rush for rebuilding the RAID volume.

Considering the cost difference between SSD and HDD, we newly designed double storage layers as follows:

1) Fast frontend storage on "distributed replicated" GlusterFS volume consisting of multiple SSDs

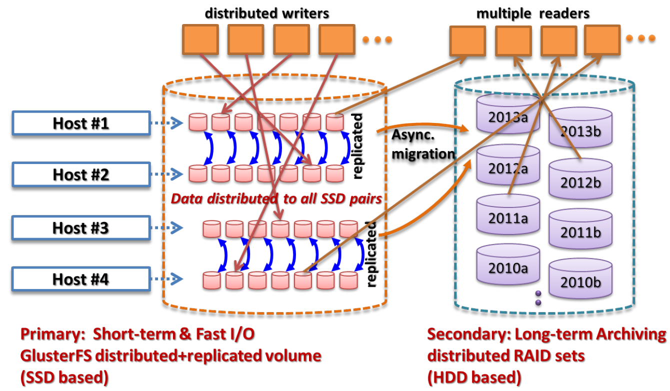


Fig. 4. Double storage layer of SSD-based GlusterFS volume and usual HDD-based RAID pairs: The frontend GlusterFS volume can be expected to have a high bandwidth up to a few GB/s owing to the parallel use of SSDs. Written data on the former will be asynchronously migrated to the latter volumes by another application process [20].

2) Long-term massive data archive on multiple replicated pairs of normal RAIDs of HDDs.

Figure 4 shows the practical design of the double layer storage structure. The frontend storage consists of 4×7 SSD for a distributed replicated GlusterFS volume to provide faster throughputs for I/O rush. The backside storage is a cluster of mirrored sets of HDD RAIDs providing a long-term huge storage space for data services to the clients. Their typical read and write performances are 2 and 0.9 GB/s for SSD arrays, and 0.9 and 0.45 GB/s for HDD arrays, respectively. We have verified that the combination use of fast SSD and large HDD arrays can provide us both high I/O performance and huge capacity together.

VI. Conclusion

The LHD is a pioneer coping with a real steady state operation of the fusion device. The LABCOM data system has developed the real-time operability for LHD steady state experiments, and is fully equipped with high-speed real-time data acquisition and streaming functionalities. To deal with more than 100 MB/s nonstop inflow, acquisition tasks have been implemented as multitasking and multithreaded processes with a shared memory area for inter-process fast data handling. The shared memory area also assists in mediating the different block sizes and speeds of individual I/O devices.

In steady state operation, endless data streams will be cut into a series of 10 s time chunks, named "subshot" in the LHD experiments, and transferred through the network finally to be archived into the storage one after another. This data chopping enables the data users to read the most recent data even while the write process continues. However, slow monitoring and more than 100 MB/s fast acquisition need longer and shorter segment sizes, so that the segment size has been modified to be able to have a variable length other than 10 s.

RT streaming service adopts a combination use of TCP and UDP protocols such as RTSP. 1/N down-resampling is also applicable on the client request.

Also coping with an increasing number of >100 MB/s fast DAQs, the central storage structure has been refined to have double layers for high speed and huge capacity. SSD and HDD arrays are used, respectively. The frontend GlusterFS distributed replicated volume is verified to be able to receive 2 GB/s inflow having the redundancy for data safety.

Surrounding utilities, such as synchronous time distribution system and web operation/monitoring console, have been also furnished for steady state operation in the LHD. We can conclude that all the developed systems and utilities have realized the successful steady state operations of the LHD data system.

These achievements will also be quite helpful in implementing the next-generation big physics experimental data system, such as ITER and JT-60SA. However, there still remain some problems on lossless data compression and comfortable data retrieval for several GB massive data. Further investigations on them are desirable in the near future.

Acknowledgment

This work was performed with the support and under the auspices of the NIFS collaboration research programs; NIFS14KUGM096, NIFS14KUTR107, NIFS14ULHH006, and NIFS14ULII001. It is also partly supported by the Graduate University for Advanced Studies (SOKENDAI).

References

[1] O. Kaneko, H. Yamada, S. Inagaki, M. Jakubowski, S. Kajita, S. Kitajima, M. Kobayashi, K. Koga, T. Morisaki, S. Morita, T. Muto, S. Sakakibara, Y. Suzuki, H. Takahashi, K. Tanaka, K. Toi, Y. Yoshimura, T. Akiyama, Y. Asahi, N. Ashikawa, H. Chikaraishi, W. Cooper, D. Darrow, E. Drapiko, P. Drewelow, X. Du, A. Ejiri, M. Emoto, T. Evans, N. Ezumi, K. Fujii, T. Fukuda, H. Funaba, M. Furukawa, D. Gates, M. Goto, T. Goto, W. Guttenfelder, S. Hamaguchi, M. Hasuo, T. Hino, Y. Hirooka, K. Ichiguchi, K. Ida, H. Idei, T. Ido, H. Igami, K. Ikeda, S. Imagawa, T. Imai, M. Isobe, M. Itagaki, T. Ito, K. Itoh, S. Itoh, A. Iwamoto, K. Kamiya, T. Kariya, H. Kasahara, N. Kasuya, D. Kato, T. Kato, K. Kawahata, F. Koike, S. Kubo, R. Kumazawa, D. Kuwahara, S. Lazerson, H. Lee, S. Masuzaki, S. Matsuoka, H. Matsuura, A. Matsuyama, C. Michael, D. Mikkelsen, O. Mitarai, T. Mito, J. Miyazawa, G. Motojima, K. Mukai, A. Murakami, I. Murakami, S. Murakami, T. Muroga, S. Muto, K. Nagaoka, K. Nagasaki, Y. Nagayama, N. Nakajima, H. Nakamura, Y. Nakamura, H. Nakanishi, H. Nakano, T. Nakano, K. Narihara, Y. Narushima, K. Nishimura, S. Nishimura, M. Nishiura, M. Nunami, T. Obana, K. Ogawa, S. Ohdachi, N. Ohno, N. Ohyabu, T. Oishi, M. Okamoto, A. Okamoto, M. Osakabe, Y. Oya, T. Ozaki, N. Pablant, B. Peterson, A. Sagara, K. Saito, R. Sakamoto, H. Sakaue, M. Sasao, K. Sato, M. Sato, K. Sawada, R. Seki, T. Seki, V. Sergeev, S. Sharapov, I. Sharov, A. Shimizu, T. Shimozuma, M. Shiratani, M. Shoji, S. Sudo, H. Sugama, C. Suzuki, K. Takahata, Y. Takeiri, Y. Takemura, M. Takeuchi, H. Tamura, N. Tamura, H. Tanaka, T. Tanaka, M. Tingfeng, Y. Todo, M. Tokitani, K. Tokunaga, T. Tokuzawa, H. Tsuchiya, K. Tsumori, Y. Ueda, L. Vyacheslavov, K. Watanabe, T. Watanabe, T. Watanabe, B. Wieland, I. Yamada, S. Yamada, S. Yamamoto, N. Yanagi, R. Yasuhara, M. Yokoyama, N. Yoshida, S. Yoshimura, T. Yoshinaga, M. Yoshinuma, and A. Komori, "Extension of operation regimes and investigation of three-dimensional currentless plasmas in the large helical device," *Nuclear Fusion*, vol. 53, no. 10, p. 104015, 2013.

[2] 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, and LHD Experiment Group, "Data Acquisition and Management System of LHD," *Fusion Sci. Technol.*, vol. 58, no. 1, pp. 445–457, 7 2010.

[3] H. Nakanishi, M. Ohsuna, M. Kojima, S. Imazu, M. Nonomura, T. Yamamoto, M. Emoto, M. Yoshida, C. Iwata, M. Shoji, Y. Nagayama, K. Kawahata, M. Hasegawa, A. Higashijima, K. Nakamura, Y. Ono, M. Yoshikawa, and S. Urushidani, "Data acquisition system for steady-state experiments at multiple sites," *Nuclear Fusion*, vol. 51, no. 11, p. 113014, 2011.

[4] "Moore's law," Wikipedia, Sep. 2015. [Online]. Available: http://en.wikipedia.org/wiki/Moore's_law

[5] I. Yonekawa, Y. Kawamata, T. Totsuka, H. Akasaka, M. Sueoka, K. Kurihara, T. Kimura, and the JT-60U Team, "JT-60U CONTROL SYSTEM," *Fusion Sci. Tech.*, vol. 42, no. 9-11, pp. 525–529, 2002.

[6] D. Alves, A. C. Neto, D. F. Valcarcel, R. Felton, J. M. Lopez, A. Barbalace, L. Boncagni, P. Card, G. De Tommasi, A. Goodyear, S. Jachmich, P. J. Lomas, F. Maviglia, P. McCullen, A. Murari, M. Rainford, C. Reux, F. Rimini, F. Sartori, A. V. Stephen, J. Vega, R. Vitelli, L. Zabeo, K.-D. Zastrow, and JET EFDA contributors, "A New Generation of Real-Time Systems in the JET Tokamak," *IEEE Trans. on Nucl. Sci.*, vol. 61, no. 2, pp. 711–719, Apr. 2014.

[7] S. Balme, Y. Buravand, P. Fejoz, F. Leroux, P. Pastor, N. Ravenel, and P. Spuig, "Real-Time Data Acquisition System for Control and Long-Pulse Operation in Tore Supra," *Fusion Sci. Technol.*, vol. 56, no. 3, pp. 1273–1283, Oct. 2009.

[8] H. Nakanishi, M. Kojima, and LABCOM group, "Design for real-time data acquisition based on streaming technology," *Fusion Eng. Des.*, vol. 56-57, no. 1-4, pp. 1011–1016, 2001.

[9] H. Nakanishi, M. Ohsuna, M. Kojima, S. Imazu, M. Nonomura, Y. Nagayama, and K. Kawahata, "Portability improvement of LABCOM data acquisition system for the next-generation fusion experiments," *Fusion Eng. Des.*, vol. 82, no. 5-14, pp. 1203–1209, Oct. 2007.

[10] M. Emoto, S. Ohdachi, K. Watanabe, S. Sudo, and Y. Nagayama, "Server for experimental data from LHD," *Fusion Eng. Des.*, vol. 81, no. 15-17, pp. 2019–2023, Jul. 2006.

[11] National Instruments, "PXI and CompactPCI," http://www.ni.com/pxi/, 2005.

[12] H. Nakanishi, M. Ohsuna, M. Kojima, S. Imazu, M. Nonomura, M. Emoto, H. Okumura, Y. Nagayama, K. Kawahata, and LHD exp. group, "Ultra-Wideband Real-Time Data Acquisition in Steady-State Experiments," *J. Plasma Fusion Res.*, vol. 82, no. 3, pp. 171–177, Mar. 2006, [in Japanese].

[13] M. Ohsuna, H. Nakanishi, S. Imazu, M. Kojima, M. Nonomura, M. Emoto, Y. Nagayama, and H. Okumura, "Unification of ultra-wideband data acquisition and real-time monitoring in LHD steady-state experiments," *Fusion Eng. Des.*, vol. 81, no. 15-17, pp. 1753–1757, Jul. 2006.

[14] H. Nakanishi, S. Imazu, M. Ohsuna, M. Kojima, M. Nonomura, M. Shoji, M. Emoto, M. Yoshida, C. Iwata, H. Miyake, Y. Nagayama, and K. Kawahata, "Improved Data Acquisition Methods for Uninterrupted Signal Monitoring and Ultra-Fast Plasma Diagnostics in LHD," *Plasma Fusion Res.*, vol. 7, p. 2405007, 2012.

[15] "Nyquist–Shannon sampling theorem," Wikipedia, Aug. 2015. [Online]. Available: http://en.wikipedia.org/wiki/Sampling_theorem

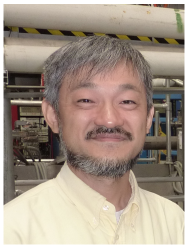
[16] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326, Apr. 1998. [Online]. Available: https://tools.ietf.org/html/rfc2326

[17] H. Nakanishi, M. Ohsuna, M. Kojima, S. Imazu, M. Nonomura, H. Okumura, Y. Nagayama, and K. Kawahata, "Nonstop Lose-less Data Acquisition and Storing Method for Plasma Motion Images," *Plasma Fusion Res.*, vol. 2, p. S1117, 2007.

[18] "Run-length encoding (RLE)," Rosetta Code, Aug. 2015. [Online]. Available: http://rosettacode.org/wiki/Run-length_encoding

[19] "Getting started with GlusterFS – Architecture," Gluster community, Jul. 2015. [Online]. Available: http://gluster.readthedocs.org/en/latest/Quick-Start-Guide/Architecture/

[20] H. Nakanishi, M. Ohsuna, M. Kojima, S. Imazu, M. Nonomura, M. Emoto, T. Yamamoto, Y. Nagayama, T. Ozeki, N. Nakajima, K. Ida, and O. Kaneko, "Revised cloud storage structure for light-weight data archiving in LHD," *Fusion Engineering and Design*, vol. 89, no. 5, pp. 707–711, 5 2014.



Nakanishi Hideya received the B.E. and M.E. degrees in nuclear engineering from the University of Tokyo, Tokyo, Japan in 1990 and 1992, respectively, and the Dr. Eng. degree from the Graduate University of Advanced Studies (SOKENDAI), Hayama, Japan in 2003.

From 1995 to now, he is with the National Institute for Fusion Science (NIFS), Toki, Japan and is currently an Associate Professor of NIFS and also of SOKENDAI. His research has been concerned with ICT applied systems for fusion plasma experiments.

Dr. Nakanishi is a member of the Japan Society of Plasma Science and Nuclear Fusion Research, the Physical Society of Japan, and the Information Processing Society of Japan.