Nonstop Lose-Less Data Acquisition and Storing Method for Plasma Motion Images

journal or publication title	Plasma and Fusion Research
volume	Vol.2
number	Special Issue
page range	pp.S1117-1 - S1117-4
year	2007-01-01
URL	http://hdl.handle.net/10655/334

doi: 10.1585/pfr.2.S1117

Nonstop Lose-Less Data Acquisition and Storing Method for Plasma Motion Images

Hideya NAKANISHI, Masaki OHSUNA, Mamoru KOJIMA, Setsuo IMAZU[1)], Miki NONOMURA, Haruhiko OKUMURA[2)], Yoshio NAGAYAMA and Kazuo KAWAHATA

National Institute for Fusion Science, Toki 509-5292, Japan
[1)]*Pretech Corp., 1-19-13 Kanayama-cho, Atsuta-ku, Nagoya 456-0002, Japan*
[2)]*Mie Univ., 1577 Kurimamachiya-cho, Tsu, Mie 514-8507, Japan*

(Received 12 December 2006 / Accepted 20 February 2007)

Plasma diagnostic data analysis often requires the original raw data as they are, in other words, at the same frame rate and resolution of the CCD camera sensor. As a non-interlace VGA camera typically generates over 70 MB/s video stream, usual frame grabber cards apply the lossy compression encoder, such as mpeg-1/-2 or mpeg-4, to drastically lessen the bit rate. In this study, a new approach, which makes it possible to acquire and store such the wideband video stream without any quality reduction, has been successfully achieved. Simultaneously, the real-time video streaming is even possible at the original frame rate. For minimising the exclusive access time in every data storing, it has adopted the directory structure to hold every frame files separately, instead of one long consecutive file. The popular 'zip' archive method improves the portability of data files, however, the JPEG-LS image compression is applied inside by replacing its intrinsic deflate/inflate algorithm that has less performances for image data.

Keywords: plasma diagnostics, loss-less compression, MPEG, real-time, streaming, JPEG-LS, zip, LABCOM/X

DOI: 10.1585/pfr.2.S1117

1. Introduction

As a popular VGA (640 × 480 pixels) full-color camera constantly generates over 70 MB/s non-interlaced video stream, ordinary frame grabber cards usually apply the lossy compression encoder, such as mpeg-1/-2 or mpeg-4, to lessen the bit rate drastically. Diagnostic analysis of fusion plasma, however, basically requires to process the raw data for extracting some analytical view or information. For 2-D diagnostics, whole the picture frames produced by camera sensors must be preserved at the same frame rate and resolution in the data acquisition (DAQ) and storage system.

As a result of it, we often encounter the "data explosion" problem, in which image data occupies the greater part of storage volume. Fig. 1 (top) shows the transition of LHD's total data amount acquired by every shot. In the recent few years, we could observe the deterioration of data compression efficiency that is obvious as the difference between raw and compressed data sizes. Detailed values are also listed in Table 1.

As camera image acquisition has started simultaneously with introducing the CompactPCI (cPCI) digitizers in LABCOM system, total amount in 7-th cycle has no camera data while 8-th cycle includes some. The compression ratio of 8-th cycle becomes much worse than previous. It has already been found that "zlib" compression algorithm, which is currently adopted as embedded scheme in LABCOM system, becomes less compressive for 2-D image data and even slower [1].

Table 1 Deterioration of compression ratio from 2003 to 2004 [1].

	7c	8c	increased %
raw (MB)	868.3	2111.8	243.2
comp. (MB)	164.5	640.7	389.6
comp./raw (%)	18.9	**30.3**	–

In this study, we have revamped some parts of DAQ mechanisms in LABCOM system to improve the efficiency of handling many 2-D camera frames or long-time video streams. We certainly have paid attention to the physics requirements that all the data processing must be "loss-less", in other words, keep the complete reversibility in case of data compression.

2. DAQ Structure

LHD data acquisition system, i.e. "*LABCOM system*" has been designed to adopt the "distributed structure" in DAQ and storage servers. Especially, an individual DAQ PC is distributed for each plasma measurement. It is very advantageous when expanding or installing a new additional measurement. The good and continuous increase of the number of diagnostics in LHD, as shown in Fig. 1 (bot-

author's e-mail: nakanisi@nifs.ac.jp

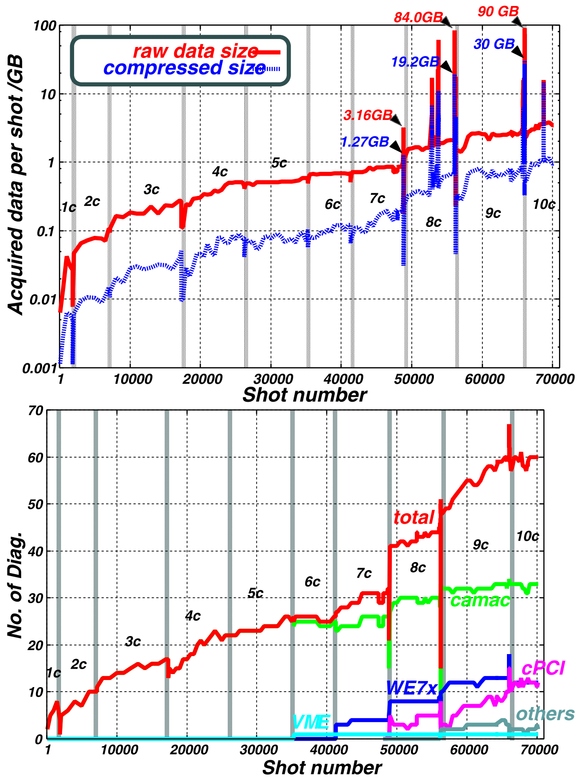

Fig. 1 Transition of shot-by-shot data amount in LHD (top), and the number of diagnostics by each digitizer kind (bottom): Some prominent peaks of data amount represent quasi-steady-state experimental campaigns.

tom), is a very good proof of it.

As our prior works already reported [2, 3], LABCOM system has been fully functional for steady-state operations with maximum acquisition throughput of 80 MB/s from every digitizer front-end to DAQ computers. Such the ultra-wideband performance has been brought by using modern CompactPCI (PXI) digitizers. A PXI frame grabber module has first enabled us to acquire full-color and full-rate VGA frames in real time. Fig. 2 shows a schematic view of LABCOM system.

3. Real-Time Streaming and Storing

The necessity of real-time (RT) data streaming is explained simply in left-hand side of Fig. 2. Not only the RT display for monitoring the status of plasma measurements but also the RT analysis by using massive-size diagnostic data would be considered to be indispensable especially in steady-state experiments.

This system, therefore, has been designed to be ready for such the ultra-wideband data streaming as they are generated, i.e. "loss-less". Although maximum 80 MB/s streaming is even possible, 1/N thinning is also available by client request. It is because client PCs often have less performance to receive the full-bandwidth stream continuously. Streams are, therefore, sent by light-weight UDP/IP,

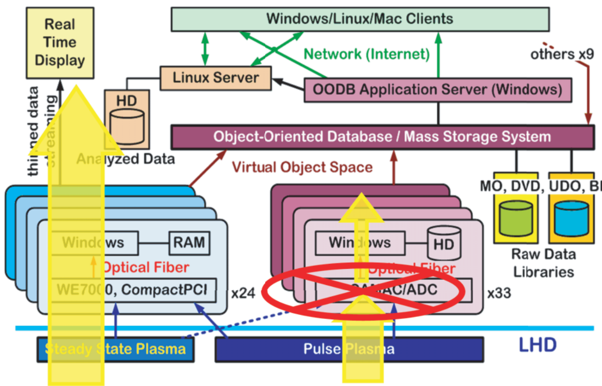

Fig. 2 Schematic view of LABCOM real-time and batch-processing data acquisition structure: Left-hand part can provides real-time data acquisition and retrieval, but right one is *not* capable for it. Legacy CAMAC digitizers works only on batch processing.

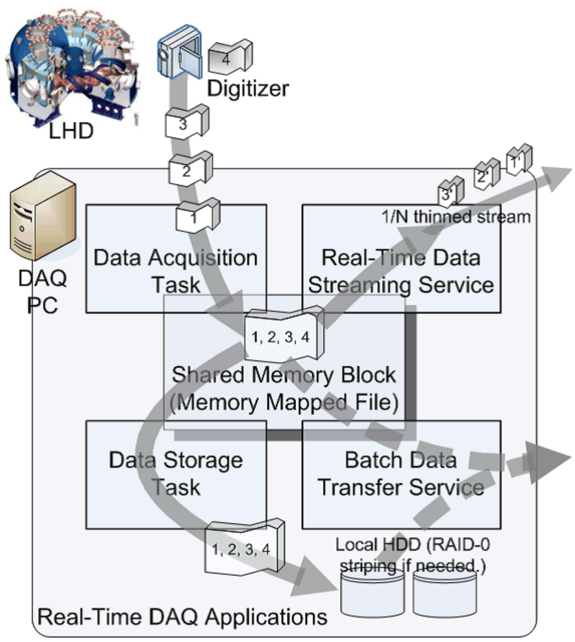

Fig. 3 Mechanism of inter-process data delivery for making ultra-wideband real-time data streaming and storing in parallel.

not heavy TCP/IP. By using a script-interpreting sample viewer in PV-WAVE data visualisation platform, we have easily confirmed 3~5 fps throughput on non-powerful Pentium III client PC. Fig. 3 explains the inside scheme of the RT DAQ server.

Another difficulty exists in continuous data storing. As it is impossible to store non-stop data inflow into an endless file, the continuous stream must be divided into multiple time chunks to be stored as individual files. As a conceptual extension from LHD's normal short-pulse experiments, we have decided to divide every RT signals into consecutive 10 second chunks and number them with incremental sub-shot integer,

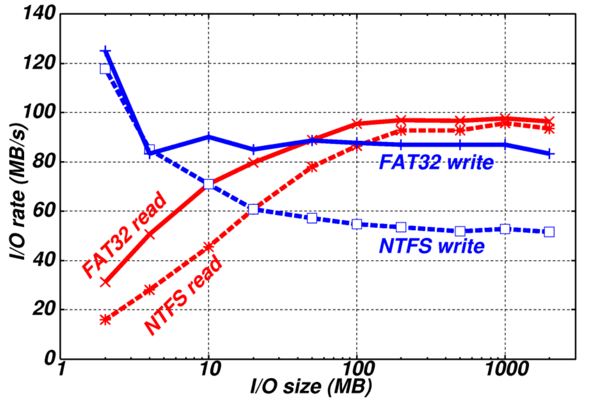

Fig. 4 Performance of 2*HDD RAID-0 striping disk array: As every HDD has ~1 MB RAM cache internally, writing throughputs for 2 MB and less block sizes never represent proper values.

such as `#654321.1, #654321.2, #654321.3, ..., #654321.NN`. This mechanism also enables us to read RT data quasi-parallelly even in the middle of non-stop data writing.

Even though larger I/O throughput generally requires bigger I/O block size, one UDP/IP message size is limited under 65535 bytes. So, data storing blocks must be much bigger than network streaming ones. In order to manage both of them in RT DAQ computer, we have applied the shared memory (SHM) mechanism to make one inflow to diverge into two ways.

HDD also has a limit for internal data transfer, which is commonly 40 ~ 60 MB/s. Therefore, RAID-0 (striping) by writing in parallel to multiple HDDs is necessary when data inflow is over 40 MB/s. Fig. 4 shows a test result of two HDD's RAID-0 performance. Here we can see that read-out performance depends on its block size proportionally, and that writing one is almost independent. It is rather much sensitive to the difference of filesystem types. It is quite natural that more secure "journaling" filesystem (NTFS) provides much less speed than non-journaling one (FAT32).

Anyway, we have confirmed that usual cameras, from 8-bit greyscale interlaced VGA (~ 8.8 MB/s) to 32-bit full-color progressive one (~ 70.3 MB/s), can be managed by our system.

4. Portable Archive File

As a camera signal contains temporal series of 2-D frames, the whole data become much bigger only by one channel output. For instance, a color VGA camera acquires about 70 GB video data for a thousand second long-pulse discharge. In such case, loading the whole data on main memory of data analyzing PCs will be almost impossible. Then the fast data retrieval of specified frames, not the whole data, will be indispensable.

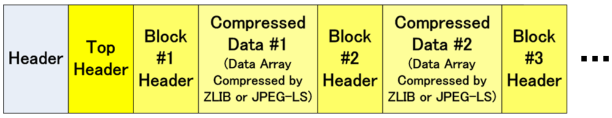

Fig. 5 "titz" archive file format: It has the inside structure to store multiple blocks, however, it has no indexing table for faster seek of every data blocks.

LABCOM system has been utilizing the (de)flate compression algorithm which is very popular as famous "zlib", "gzip", and "zip" packages. From once compressed block, it is very difficult to retrieve some part of a whole block without decompression. So, in order to make it possible to pick out a part of data, they must be first divided into smaller pieces and then compressed separately. For image data, frame-by-frame compression is considered to be most convenient.

To realize the structured compressed blocks, we first tried to introduce "titz" library which outputs a sequentially attached compressed blocks with some informative headers [4]. However, the seek speed to jump to the specified block becomes slower with longer titz archive. It is because in titz archive we always have to seek the whole file from the beginning until reaching the specific block. (Fig. 5)

For enabling direct jump to the head of specific block, archive file should have so-called an "index" table of contents inside. Therefore, we decided to apply standard "`.ZIP`" archive format because it has a fast seeking index at the end, namely "central directory" [5].

```
[local file header 1]
[file data 1]
[data descriptor 1]
.
.
.
[local file header n]
[file data n]
[data descriptor n]
[archive decryption header]
[archive extra data record]
[central directory]
  [file header 1]
  .
  .
  [file header n]
  [digital signature]
[end of central directory record]
```

This inside structure of `.ZIP` also enables us to have "Shot/Channel/(Frame)" sub-container structures with data block(s) and meta-informative text files in each. Therefore, all waveform chunks and image frames are compressed separately and stored having independent filenames in archive.

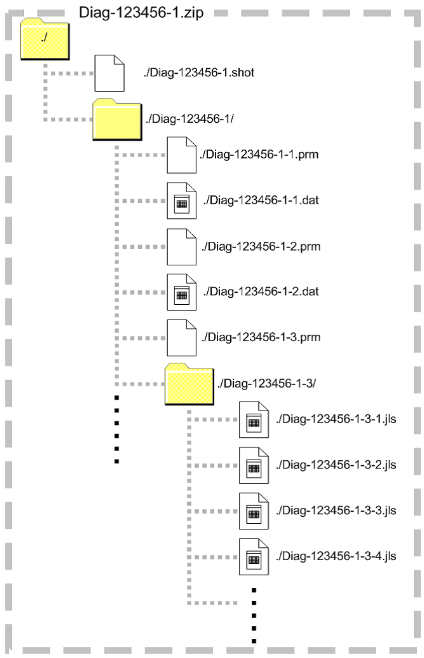

Fig. 6 Contents' structure of new LABCOM/X `.ZIP` archive file: "`.jls`" filename extension means JPEG-LS file.

As zlib (de)flate method is embedded in `.ZIP` utility, 1-D waveform data can be compressed and archived uninterruptedly into `.ZIP` file. On the other hand, 2-D image data have to be compressed by the external method before making structured archive file. We have examined both the compression speed and efficiency of plural algorithms, and decided to adopt JPEG-LS [1, 6]. This new file archiving has been named as "LABCOM/X" format in which a series of 1-D and 2-D data files can be mixed, as shown in Fig. 6.

5. Conclusion

In this study, a new approach, which makes it possible to acquire and store such the wideband video stream without any quality reduction nor time gap, has been successfully achieved. Simultaneously, the real-time video streaming is even possible at the original frame rate.

For minimising the exclusive access time in every data storing, it has adopted the directory structure to hold every frame files separately, instead of one long consecutive file. The popular "zip" archive method improves the portability of data files, however, the JPEG-LS image compression is applied inside by replacing its intrinsic deflate/inflate algorithm that has less performances for image data.

By using a standard and popular `.ZIP` archive format, data portability has been quite improved than the previous private one. We could expect that this advantage will lead to a better inter-connectivity in multi-site data systems.

Acknowledgements

This study is supported by NIFS under NIFS06ULHH503 and NIFS05KCHH004, and also by Softpia Japan 2006 Collaborative Research Project.

[1] M. Ohsuna, H. Nakanishi, S. Imazu, M. Kojima, M. Nonomura, M. Emoto, Y. Nagayama, H. Okumura, Fusion Eng. Des. **81** (15-17), 1753 (2006).

[2] H. Nakanishi, M. Ohsuna, M. Kojima, S. Imazu, M. Nonomura, M. Emoto, H. Okumura, Y. Nagayama, K. Kawahata, LHD exp. group, J. Plasma Fusion Res. **82**, 171 (2006), (in Japanese).

[3] H. Nakanishi, M. Kojima, M. Ohsuna, S. Komada, M. Nonomura, M. Yoshida, S. Imazu and S. Sudo, Fusion Eng. Des. **66-68**, 827 (2003).

[4] Tokyo Inst. Tech., TITech Data Compression Library for Large Scale Data, http://www.gsic.titech.ac.jp/˜titz/docs/en/ (2003).

[5] PKWARE Inc., .ZIP File Format Specification, http://www.pkware.com/business_and_developers/developer/appnote/ (2006).

[6] HP Labs, LOCO-I/JPEG-LS, http://www.hpl.hp.com/loco/ (2005).