Real-time data compression for data acquisition systems applied to the ITER Radial Neutron Camera

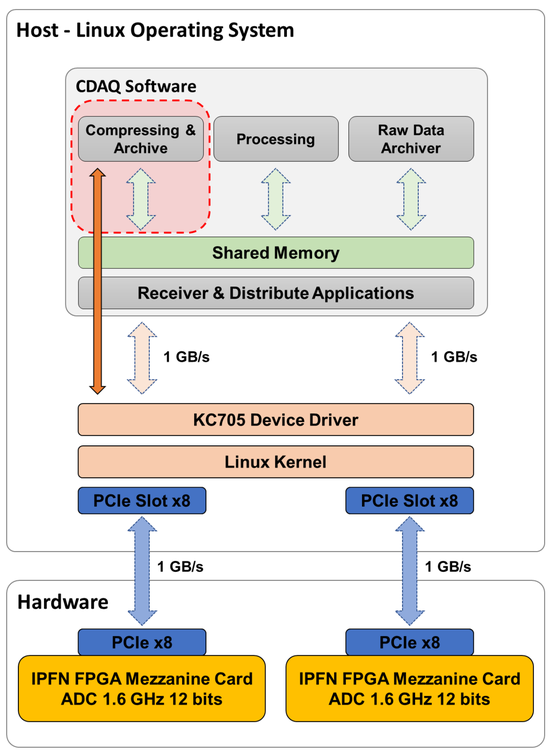
B. Santos[1], N. Cruz[1], A. Fernandes[1], P.F. Carvalho[1], J. Sousa[1], B. Gonçalves[1], M. Riva[2], F. Pollastrone[2], C. Centioli[2], D. Marocco[2], B. Esposito[2], C.M.B. Correia[3], B. Brichard[4] and R.C. Pereira[1]

[1] Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
[2] ENEA C. R. Frascati, Dipartimento FSN, via E. Fermi 45, 00044 Frascati (Roma), Italy.
[3] LIBPhys-UC, Department of Physics, University of Coimbra, P-3004 516 Coimbra, Portugal
[4] Fusion for Energy, Barcelona, Spain.

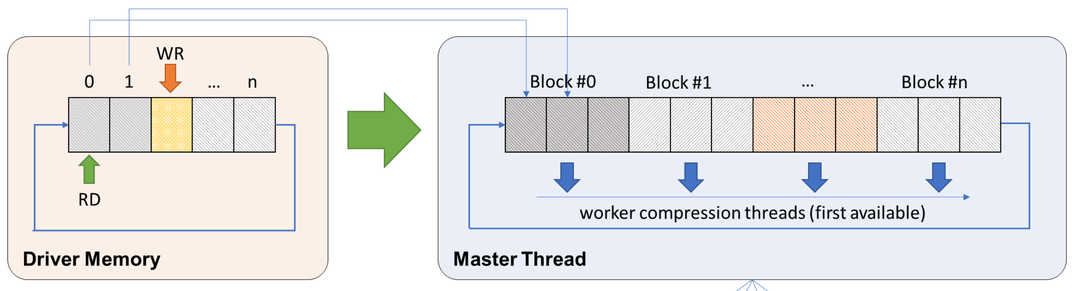
Introduction

- To achieve the aim of the **ITER Radial Neutron Camera Diagnostic**, the data acquisition prototype must be compliant with a **sustained 2 MHz peak event** for each channel with 128 samples of 16 bits per event. The data is acquired and processed using an IPFN FPGA Mezzanine Card (FMC-AD2-1600) with 2 digitizer channels of 12-bit resolution **sampling up to 1.6 GSamples/s** in a PCIe evaluation board from Xilinx (KC705) installed in the host PC.
- The acquired data in the event-based **data-path is streamed to the host through the PCIe** x8 Direct Memory Access (DMA) and the maximum data **throughput per channel is ~0.5 GB/s of raw data (event base), ~1 GB/s per digitizer** and up to **~1.6 GB/s in continuous mode**.
- The prototype architecture comprises **one host PC with two KC705** modules and **four channels**, **producing up to ~2 GB/s in event mode** and up to **~3.2 GB/s in continuous mode**.
- **Real-time data compression to reduce data throughput** from the host to ITER databases was evaluated using the **LZ4 lossless compression algorithm**, which provides **compression speed up to 400 MB/s per core**.
- This contribution presents the **architecture, implementation and test** of the **parallel real-time data compression** system running in **multiple isolated CPU cores**. The average space saving and the performance results for long pulse acquisitions up to 30 minutes, using different data block sizes and different number of CPUs, are also presented.

System Architecture



- The Host Computer runs **Scientific Linux 7** as Operating System with **kernel 3.10-rt** and **LZ4 version 1.7.5**
  - Intel® Core™ i7-5930K@3.50 GHz (12 cores)
  - 256 GB SSD and 64 GB of RAM
- To **test** the **compression** branch **maximum performance**, in the presented tests, the **compression application** is directly **connected to the device driver**.
- The device driver implements an internal **circular buffer** to store temporally the data transferred from the hardware, until it is read from the consumer applications.
- The **shared memory layer** is only used when several clients need the data at same time.
- Introducing the shared memory layer, the compression **performance must not be compromised**.



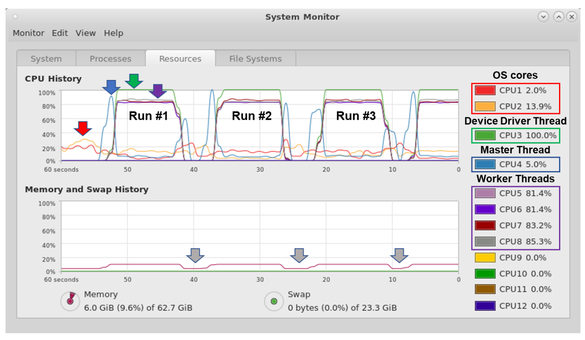
- The master thread and the worker threads runs in **isolated cores**.
- The **master thread** reads the data from device driver, **packs it in data blocks** with configurable data size and **distributes** it across the configurated number of **worker threads**.
- Each **block** is **tagged with an id**.
- The **id** is used to **store the compressed data in the right position** of a shared buffer between worker threads
- The **worker threads** can **store the compressed** data in a memory buffer or **deliver** it **through the network** directly to the data archiver.

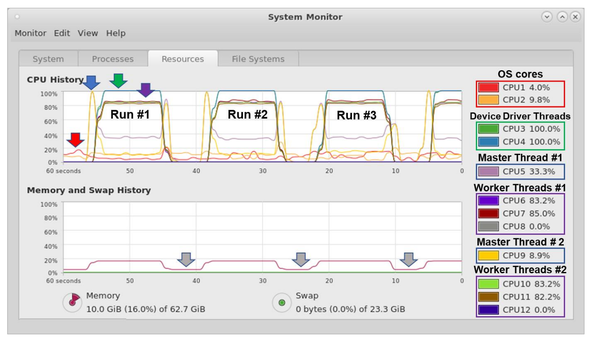
Results

- The **minimum number of needed cores** per acquisition rate (with **no data loss**) is presented in the table below.
- The **device driver** and the **master thread** implement **algorithms to check the data loss**
- The input signal was a pulse type signal (gamma distribution) from a waveform generator with different pulse width using a 10 MB block size during 60 minutes.

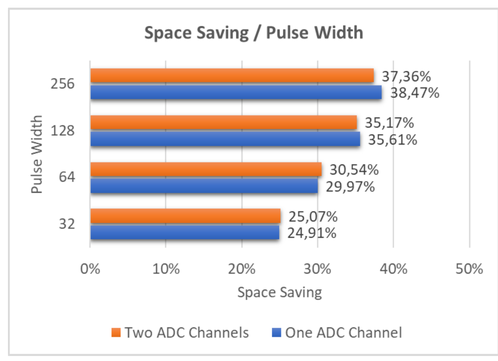
Number of Channels	Pulse Width	Data Rate (MB/s)	Data Loss / Number of CPUS						CPU Load (avg/core)		Compression				
			1	2	3	4	5	6	Worker Threads	Master Thread	Speed (MB/s) Avg	Speed (MB/s) Min	Speed (MB/s) Max	Ratio	Space Saving
1	32	128	0,00%	-	-	-	-	-	37,00%	0,54%	353,73	342,20	355,63	1,33	24,91%
1	64	256	0,00%	-	-	-	-	-	81,00%	0,59%	317,43	307,80	319,10	1,43	29,97%
1	128	512	40,68%	0,00%	-	-	-	-	84,00%	0,57%	302,76	294,87	304,23	1,55	35,61%
1	256	768	61,06%	22,33%	0,00%	-	-	-	86,00%	0,56%	297,74	280,02	299,04	1,63	38,47%
2	32	256	0,00%	-	-	-	-	-	73,00%	0,71%	352,35	340,90	353,41	1,33	25,07%
2	64	512	36,94%	0,00%	-	-	-	-	80,00%	0,57%	321,14	312,86	322,40	1,44	30,54%
2	128	1024	69,38%	38,83%	8,80%	0,00%	-	-	82,00%	2,75%	310,64	212,45	313,25	1,54	35,17%
2	256	1536	79,58%	59,27%	39,09%	19,12%	4,32%	0,00%	88,00%	25,64%	284,58	196,20	313,33	1,60	37,36%

- The figures below present the **CPU and memory usage** during 3 runs of 10 seconds duration with 1024 MB/s of data acquisition rate. The **left** snapshot presents the state of the system during an acquisition from **one board** and the **right** from **two boards simultaneously** (512 MB/s per board).





- The chart in the right presents the **relation between space saving and pulse width** for signals using one and two ADCs.
- The table below presents the test of **different data block sizes** with same input signal and configurations (1 Channel, Pulse Width 128, Data Rate 512 MB/s).



Block Size (MB)	CPU Load (avg/core)		Compression				
	Worker Threads	Master Thread	Speed (MB/s) Avg	Speed (MB/s) Min	Speed (MB/s) Max	Ratio	Space Saving
1	85,00%	0,86%	296,86	269,57	298,93	1,52	34,14%
10	84,00%	0,57%	302,76	294,87	304,23	1,55	35,61%
100	83,00%	0,63%	305,43	304,82	306,06	1,54	35,06%
200	82,00%	0,86%	306,36	305,34	306,77	1,52	34,29%

Conclusions

- The presented **architecture is scalable and adjustable**. The number of worker threads can be configured to comply with different algorithms and data throughput.
- The stress test shows a **stable solution during 60 minutes acquisitions** with acquisition rates at 1.5 GB/s, using 6 worker threads in parallel.
- The system was also **tested** in Fedora Linux 27, **kernel 4.16** with similar results.
- It was identified a relation between space saving and pulse width.
- The data block size did not improve the space saving, however the standard deviation of the compression speed during the pulse is reduced.
- To compress **1 GB/s from one board**, a minimum of **5 cores** is needed (1 master and 4 worker threads)
- Based on the tests with individual boards:
  - The system needs **14 cores** to compress **1 GB/s from two boards** in real-time (2 cores for operating system, 2 cores for the device driver, 2 cores for master thread and 8 cores for the worker threads).
- The preliminary tests with two boards simultaneously showed a performance decreasing in the minimum number of CPUs (With 512 MB/s of data rate acquisition the system needs 3 cores, instead of 2 with a single board). Tests with two hardware modules acquiring simultaneously are scheduled to a future task.

Main References:

[1] N. Cruz et al., Real-time software tools for the performance analysis of the ITER Radial Neutron Camera, Fusion Eng. Des. 123 (2017) 1001-1005.

[2] R.C. Pereira et al., Real-Time data acquisition Prototype proposal of the ITER radial neutron camera and gamma-ray spectrometer , Fusion Eng. Des. 123 (2017) 901-905.

[3] LZ4 website, https://lz4.github.io/lz4/

Corresponding author: bsantos@ipfn.tecnico.ulisboa.pt

The work leading to this publication has been funded partially by Fusion for Energy under the Contract F4E-FPA-327. IST activities also received financial support from “Fundação para a Ciência e Tecnologia” through project UID/FIS/50010/2013. This publication reflects the views only of the author, and Fusion for Energy cannot be held responsible for any use which may be made of the information contained therein.