Research and Implementation of Efficient Parallel Processing of Big Data at TELBE User Facility

Mohammed Bawatna
Institute of Radiation Physics
Helmholtz Zentrum Dresden
Rossendorf
Dresden, Germany
ORCID: 0000-0002-2382-3363

Bertram Green
Institute of Radiation Physics
Helmholtz Zentrum Dresden
Rossendorf
Dresden, Germany
ORCID: 0000-0002-7184-5620

Sergey Kovalev
Institute of Radiation Physics
Helmholtz Zentrum Dresden
Rossendorf
Dresden, Germany
ORCID: 0000-0002-2290-1016

Jan-Christoph Deinert
Institute of Radiation Physics
Helmholtz Zentrum Dresden
Rossendorf
Dresden, Germany
ORCID: 0000-0001-6211-0158

Oliver Knodel
Department of Computational Science
Helmholtz Zentrum Dresden
Rossendorf
Dresden, Germany
ORCID: 0000-0001-8174-7795

Rainer G. Spallek
Institut für Technische Informatik
Technische Universität Dresden
Dresden, Germany
rainer.spallek@tu-dresden.de

Abstract— **In recent years, improvements in high-speed Analog-to-Digital Converters (ADC) and sensor technology has encouraged researchers to improve the performance of Data Acquisition (DAQ) systems for scientific experiments which require high speed and continuous data measurements — in particular, measuring the electronic and magnetic properties of materials using pump-probe experiments at high repetition rates. Experiments at TELBE are capable of acquiring almost 100 Gigabytes of raw data every ten minutes. The DAQ system used at TELBE partitions the raw data into various subdirectories for further parallel processing utilizing the multicore structure of modern CPUs.**

Furthermore, several other types of processors that accelerate data processing like the GPU and FPGA have emerged to solve the challenges of processing the massive amount of raw data. However, the memory and network bottlenecks become a significant challenge in big data processing, and new scalable programming techniques are needed to solve these challenges. In this contribution, we will outline the design and implementation of our practical software approach for efficient parallel processing of our large data sets at the TELBE user facility.

Keywords— Big Data, Data Processing Pipeline, Data Acquisition Systems, Signal Processing, Data analytics

I. INTRODUCTION

As data acquisition and sensor technologies advance, data is acquired at an increasing rate and in vast quantities. In the scientific field, a lot of signal processing needs to be done to extract the value out of the raw data. This is usually done manually by researchers on every piece of measured data, and the results are placed together to know the final results of the measurement. The major challenge of integrating the modern ultra-fast ADC in the DAQ system is to handle the high rate of data measured from different sources and to be able to incorporate these data.

If the generated raw data is a too large or too long time to be processed in a simple processor, then several processors can work in parallel to handle this massive data. However, many complex algorithms are needed to divide the raw data into small chunks that can be processed in parallel as in [1-3]. Moreover, the synchronization and subtasks allocation between the working parallel processors is a significant challenge. There are two main ways for managing the task allocations between the parallel processors: shared memory, and message passing as in [4][5]. In the shared memory technique, all the processors have the same privileges to access the data. However, several algorithms are used to avoid possible consistency and memory writing conflicts. In the message passing technique, which is more suitable to the distributed processors that communicate through the network, each processor can send the messages with the processed results to other processors. The latter technique is more suitable for heterogeneous working processors. In this paper, the above method is implemented and tested on 40 identical CPU of 2.39 GHz Intel Xeon Processor with 150 GB of installed RAM [6].

At the THz source at the Electron Linac for beams with high Brilliance and low Emittance (TELBE) user facility [7], where the pump-probe experiments as in [8-11] can be done by exiting the samples with by THz pulses at 100 kHz and probing the results by laser pulses at 200 kHz as shown in Fig. 1. The data volume and acquisition rate problems cause by the massive increase in the amount of measured data is a real challenge because making sense of the data requires a massive amount of data processing. On the other hand, an emerging technology for super high parallel computing like GPUs and FPGAs need much work in combining the data and ensuring maximum efficiency of data processing can be achieved. Effective software design is essential to decrease the processing bottlenecks that not only come from the memory, disk, and network, but also from the part of raw data that cannot be parallel processed. The workflow for big data analysis presented in this paper is for studying the beam profile based on the characteristics of THz sources at the accelerator-based photon sources as in [10][11].

The outline of the paper is organized as follows: first; we will present the current Data Acquisition (DAQ) system at TELBE, along with the structure of the experimental data, challenges, and limitations. Second; we show the design for a workflow approach for parallel processing of our big data, and the factors limiting the speed of the process. Finally, we will discuss our plan for extending the efficiency of parallel processing at TELBE.

II. Related Work

There are several approaches for data acquisition and parallel processing at several THz sources machines around the world such as Karlsruhe Research Accelerator (KARA), Deutsches Elektronen Synchrotron (DESY), and THz source at ELectron source with high Brilliance and low Emittance (TELBE). The data structure in each approach depends on the beamline status, activates, the data throughput, the variety of experiments, and the need for fast feedback. Fig. 1 shows the typical chain of information for photon science experiments.

The current and upcoming advanced detectors produce very high data throughput, which motivates the research in the field of data flow management and parallel processing. However, the variety of the data structure at each light source requires a different evaluation of the parallel portion of the code. In this section, we will discuss the used hardware architecture to perform the parallel processing of the recorded raw data at each light source.

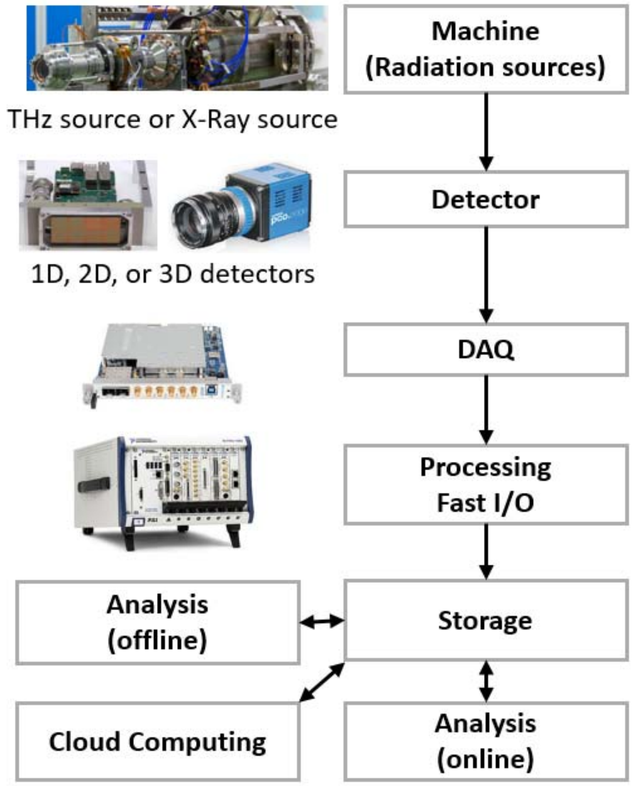

Fig. 1. The typical chain of information at light sources.

The proposed Karlsruhe Linear array detector for MHz repetition rate SpectrOscopy (KALYPSO) at Karlsruhe Research Accelerator (KIT) (in short KARA), as in [12] and [13] allows recording high data-rates up to 30 Gb/s over long time scales, this approach is a linear detector array with a data acquisition system (DAQ) at 2.7 MHz repletion rate of THz pulses.

The THz pulses for pump-and-probe experiments with the fs pulses from the Free-electron LASer (FLASH) at DESY in Hamburg, as in [14], has unique data acquisition system which was developed to record all relevant data from about 900 ADC channels with 800 bunches per cycle. The control and data acquisition systems for FLASH are based on DOOCS, as in [15] and [16], a Distributed Object-Oriented Control System, and they run on Solaris and Linux machines with an interface to Windows. This DAQ system is formed of a multiprocessor SUN Fire E2900 hosting 32 GB of memory (the DAQ server) and a SUN Fire X4500 providing 22 TB RAID file space using SUNs Zeta byte File System (ZFS). The primary computing requirements at most of the light sources are fast feedback which is to reduce the time to complete the experiment and improve data quality, the need for very short startup time, sufficient throughput between storage and processing, and the need to tune data analysis during experiments

III. Characteristics of DAQ System at TELBE

In order to overcome the significant limitations of the 4th generation light sources in terms of their use in time-resolved ultrafast experiments, we developed 100 kHz high repetition rate pulse-resolved arrival time monitor [9] with high dynamic range and temporal resolution ~10 fs as shown in Fig. 2.

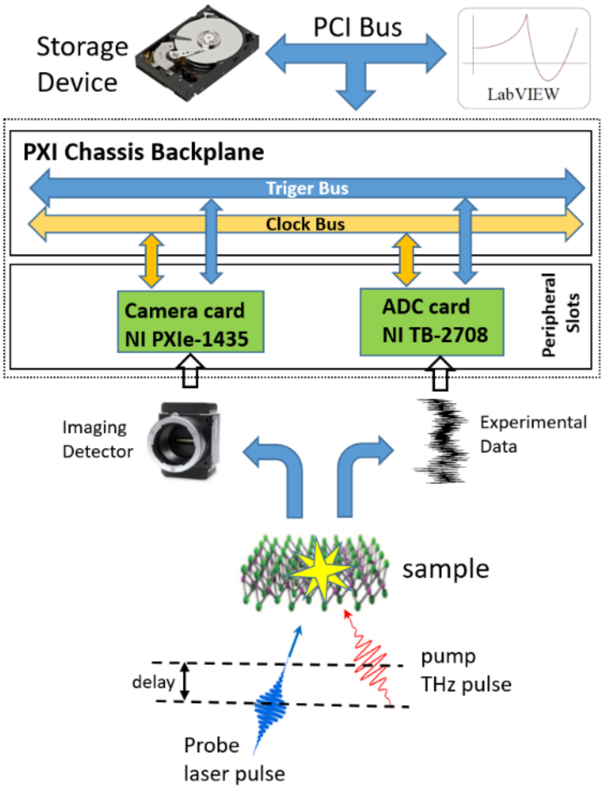

Fig. 2. Block diagram for the current DAQ system hardware at TELBE.

The DAQ system consists of two major parts: High-speed imaging detector which monitor the arrival time of the THz pulses, and two photodetectors connected to the Analog-to-Digital Converter (ADC) card to measure the experimental data.

The generated raw data volume depends on the speed of the used imaging detector and ADC card. At TELBE, 100 kHz frame rate, Basler SPL2048 spectrometer camera is used as a linear array imaging detector which generates a 358 Megabytes/Second of raw data diagnostic. For each THz pulse, the imaging detector capture 2048 pixels as in Fig. 3

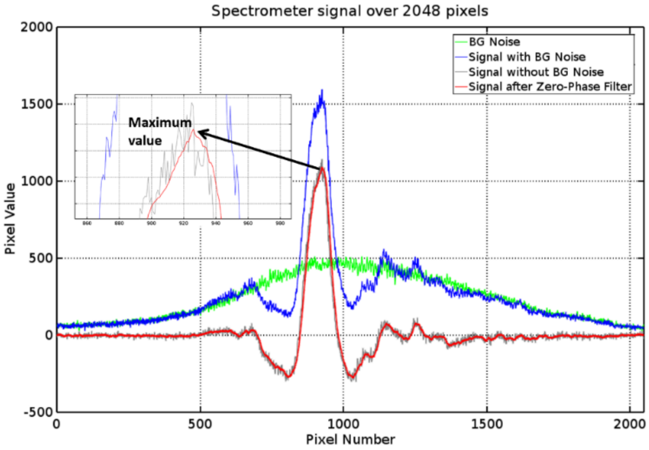

Fig. 3. The signal measured by the spL2048 camera for single-shot measurement of THz pulse at TELBE. The frame rate in these measurements was 100 kHz

The pixels generated by the imaging detector are stored as ten images of TIFF format at a size of 2048x10000 for each.

The speed of the ADC card determines the Signal-to-Noise Ratio (SNR) of the measured experimental data. However, using ultra-fast ADC will add more challenges to the volume of raw data. Currently, we use theTB-2708 ADC card that measures the experimental data at a speed of 10 Mega Sample Per Second (MSPS).

A large volume of raw data is generated every day at TELBE, and further off-line analysis is required to produce a signal that is readable by the scientists as in Fig. 4.

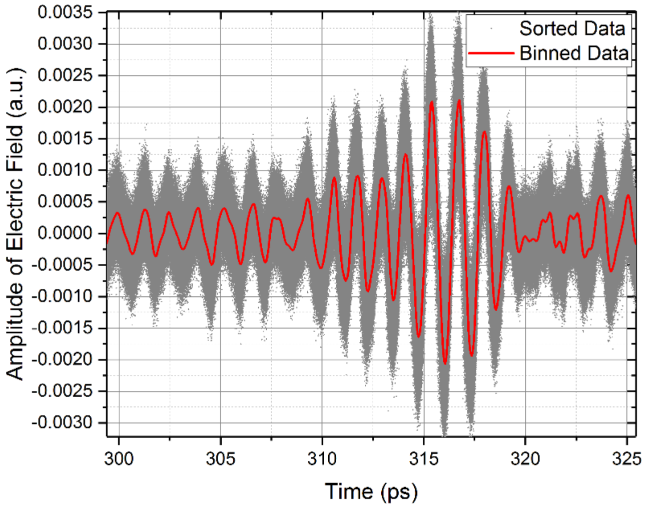

Fig. 4. Electro-Optic Signal (EOS) of 0.8 THz pulses: (grey dot) Raw, sorted data with added arrival time data from pulse-resolved diagnostics, (red line) Averaged data in 50 fs range bins.

IV. Workflow Approach for Big Data Parallel Processing at TELBE

For each pump-probe experiment, several loops are measured, and every loop contains M number of steps that can be between 50 and 200 as in Fig. 5.

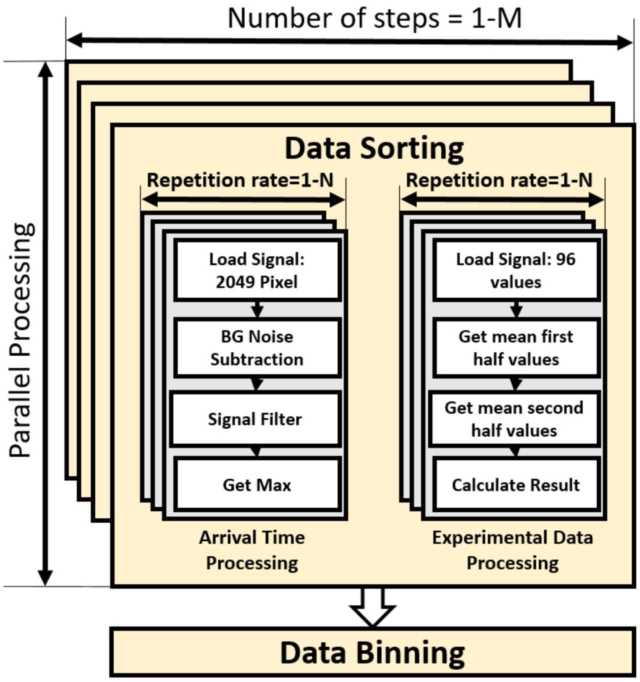

Fig. 5. Block diagram of the parallel processing workflow at TELBE.

The data acquisition software was written in C++ using the PThreads library. Each thread has its stack, and executes a given piece of code, and shares its memory with other threads, and thus can access the same global variables, same heap memory, and the same set of file descriptors. All these threads execute in parallel.

The first step in the data sorting process is to take the raw line image and convert it into a time value. This is done by first converting each line in the TIFF image to an array of integer values, then filtering each array to remove high-frequency noise which can interfere with peak finding. Then, a peak finding algorithm finds the pixel where the peak occurs. Each pixel is assigned a time of 10 f, and this is the arrival time of this particular pulse.

Next, the pump-probe experiment output is processed. This data consists of an array of single-precision points taken at a rate of 10 MHz. Each point represents the voltage reading from the detector at 100 ns intervals. The first step is to calculate a signal strength based on this array of data points. The described system uses a mean of the points, to obtain a single numeric value to represent signal strength. Simply using the highest data point value introduces high-frequency noise into the signal, as well as never returning a zero value due to imperfect balancing in the balanced detector. Once both an arrival-time value and a signal value have been calculated from the raw data, they are matched up. Several steps can be taken sequentially, each time slightly moving the linear delay stage that is part of the probe laser beamline. The software will add a delay to all the points of each step which correspond to the delay introduced by the movement of the delay stage. Based on the number of steps in each loop, the data sorting process can be done in parallel, so for each step, a thread can be used to perform the calculation of both the arrival-time value and the signal value.

The next step is data binning. After the arrival-time data is added to the individual points to provide a much more accurate picture of the pulse, the points collected into 50 fs wide bins and averaged to reduce the intensity noise. This process cannot be done in parallel.

```
**Initialize** *Time_START*, *Time_END*, *Repetition_Rate*,
*Time_Counter_Array*, *Sum_Array*;
**FOR** each Step % usually 150 steps
 **FOR** Time from *Time_Start* to *Time_End* % increment 50 fs
   **FOR** *Repetition_Rate* from 1 to 100000 % increment 1
       **IF** *Signal_Value* in the time range
          increment the  *Time_Counter_Array* by 1
          Add *Signal_Value* to *Sum_Array*;
      **ENDIF**
   **ENDFOR**
 **ENDFOR**
**ENDFOR**
% calculating the Average values
**Initialize** *Binned_Data_Array*
**FOR** Time from *Time_Start* to *Time_End* % increment 50 fs
   *Binned_Data_Array* = *Sum_Array* / *Time_Counter_Array*;
**ENDFOR**
```

Fig. 6. Pseudo code for the process of data binning.

The generated experimental data at 100 kHz repetition rate is partitioned and stored in text files and TIFF images. The text files at the size of 38 Megabytes are for the experimental raw data; the number of these text files determine the number of total steps inside the measured loop. The TIFF images are for the arrival time data measured by the linear array detector. For each 100000 THz pulse, ten TIFF images consist of a matrix of 2048 x 10000 pixels for each are generated and stored.

V. Experiments and Discussion

In order to assess the performance of parallel processing of our big data at TELBE, real tests were performed with a variable number of identical 2.4 GHz Intel Xeon Processor (Skylake, IBRS), and 150 GB of installed DDR memory are used as in TABLE I.

TABLE I. Specifications of the used Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz in this experiment

Specification	***The result from the lscpu command***
Architecture	x86_64
CPU(s)	80
CPU max MHz	3700
CPU min MHz	1000
NUMA node(s)	2
Vendor ID	GenuineIntel
CPU family	6
Model	85
BogoMIPS	4800
L1d cache	32K
L1i cache	32K
L2 cache	1024K
L3 cache	28160K

In this experiment, the workflow application was written in C++ code, and Intel Hyperthreading technology was implemented as in [12][13] for evaluating the parallel processing functions listed in Fig. 5. The data set used in this experiment consists of ten loops at the size of 69,257 GB for each loop that represent 150 steps per loop. Each loop consist of 1500 TIFF images at the size of 409 MB for each image, and 150 text files with of 38.4 MB for each text file.

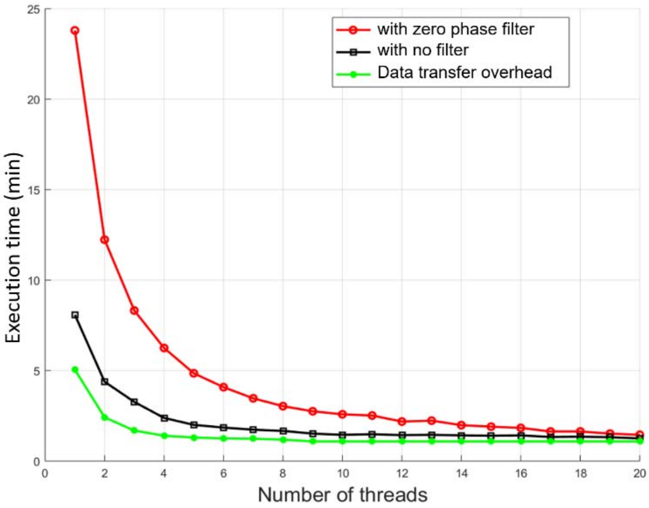

Fig. 7. Execution time [min] for one loop versus the number of threads.

The data set is stored in a Network Attached Storage (NAS) device which has multiple hard drives in a RAID configuration.

Based on the measured results, as shown in Fig. 7, the comparison of execution times for the data set in three cases determines the factor limiting the speed up of the workflow application at TELBE. The drop in performance comes from bottlenecks in memory and disk access. Furthermore, the execution time of the zero phase filter computational calculations as in [19-21] is affected by the size of caches at the CPU and the installed dynamic RAM.

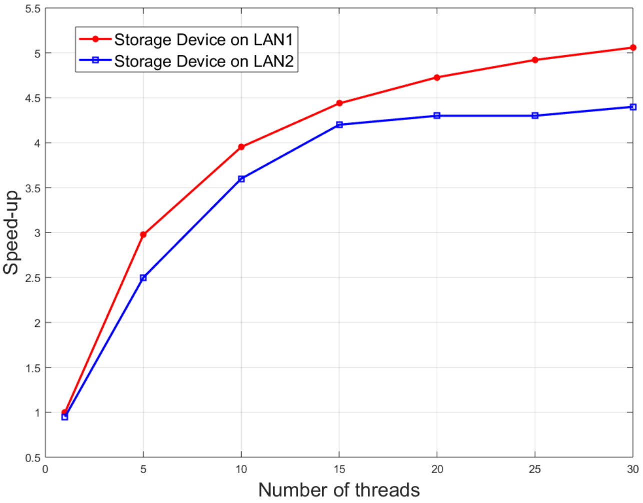

Fig. 8. Speed-up versus the number of threads

The practical speed-up of parallel processing can be affected by three factors: the parallel portion of the code, the speed of the processor, and the speed of the computer resources such as the main memory, network throughput, and the storage device. Fig. 8 shows the speed-up versus the number of used threads when loading the data set of 67395.57 GB, 3010 files, through two different Local Area Networks: LAN1 in which the storage devices are connected with the data processing devices through the same IEEE 802.3 standard Gigabit router that is dedicated to this workflow application, and LAN2 in which the storage devices are connected through virtual LAN that contains both of fast Ethernet and Gigabit Ethernet network speeds. However, as the number of processing or storage nodes grow significantly, the congestion control and access methods in LAN2 plays an essential role in the architecture of High-Performance Computing (HPC) systems and data centers.

VI. Future Data-Intensive FPGA Processing

The rapid development in the reconfigurable processing units, namely the Field Programable Gate Array (FPGA) that contains thousands of embedded Digital Signal Processing (DSP) units and have the capability to perform the required parallel processing to process the data stream from an ADC which can generate vast amounts of data, make the online data processing more attractive for the developers of high speed DAQ systems.

The current imaging detector used at TELBE supports a line rate up to 140 kHz. Due to the limitations in the number of pixels supported by high frame rates imaging detectors available today, we decided to use the KALYPSO linear array detector due to its unprecedented MHz line rate and 256 pixels. KALYPSO is a front-end ASIC that measure photon energy at the range of visible light, as in [12] and [13]. As future work, we will improve the repetition rate of our DAQ system using KALYPSO detector version 2.1 with interface implemented in High-Flex FPGA board. The specifications of the KALYPSO version 2 imaging detector are presented in TABLE II.

TABLE II. Comparison of the characteristics of image sensors used at TELBE

Specification	**Basler SPL2048**	**KALYPSO v2**
Number of pixels	2048 pixels	256 pixels
Pixel width	10 µm	50 µm
Pixel height	10 µm	3 mm
Sensor type	CMOS	CCD
Wavelength	400nm - 1µm	300nm - 1µm
Color	Mono	Mono
Line rate	Up to 140 kHz	Up to 2.7 MHz
Sensor width	20.5 mm	12.8 mm
Pixel bit depth	12 bit (ADC)	12 bit (ADC)

The High-Flex board, as in [13] has Virtex XC7VX330T FPGA on it, with almost 2 million of Configurable Logic Blocks (CLBs), 68 Mb of RAM Blocks (BRAM), 4 GigaByte of DDR3 with 64 lanes at 1866 Mb/s, two High Pin Count FMC connectors, and a PCIe Gen 3 x 16 lanes that enable a full duplex data throughput up to 130 Gb/s. The Printed Circuit Board (PCB) has 16 layer metals stack, and support a picosecond time controlled transmission lines.

Moreover, we plan to enhance the signal to noise ratio and the time resolution by integrating a new purpose-built CMOS line array detector, as in [12] and [13] that is capable of measuring at MHz repetition rates, which dramatically increase the volume of raw data, and hence the required resource allocation.

VII. Integration Into an Overall Scientific Data Management Workflow

The next generation on-line data processing update at the TELBE user facility as described in the previous section requires regarding the FAIR Guiding Principles for scientific data, as in [24], the integration of the experiment into a reproducible data management workflow. The goal is that the research data is Findable, Accessible, Interoperable, and also Reusable. By starting with the experiment itself, most of the raw (experiment) and post-processed (publication) data has to be stored for long-term archiving into our data center.

,In terms of workflow-integration, we plan to establish a workflow management system based on workflow templates with the opportunity to define a custom workflow similar to [26]. We are beginning with the generation of the data directly on the experiment. The workflow lifecycle describes the whole scientific process ranging up to publication and long-term archiving to meet the requirements of a state-of-the-art publication process. The workflow itself includes the storage of the raw experiment data, different kinds of associated metadata and first automated post-processing steps, e.g. the work described in Section III. The described post-processing can then be executed on multiple nodes in a computing cluster of heterogeneous processor cores (such as FPGAs and GPUs). Individual post-processing steps can be described in a supplementary workflow. Therefore, instances are created and mapped to available resources for execution. In the end, every processing step on the data can be traced and stored for a later publication.

The planned underlying data management system is similar to the system in [25] and provides the opportunity to organize, manage and access the data inside the data center using namespaces and replication rules. The datasets generated by the experiment are managed in containers and the researcher can study the data using specialized hardware for analysis. Therefore the use of objects is essential to work with the data efficiently. The idea is that every researcher can define his container connected with different datasets. Also, storage and file transfers to other locations and data centers using modern transport protocols are considered.

For the data center managing universal workflows, the use of special hardware for data processing is essential to meet timing, real time, energy, and resource restrictions. The executables –required for the special hardware in the workflows– are provided by component libraries. A crucial challenge is the specific executables: optimized software using different parallelization paradigms, FPGA bitstreams, and GPU kernel. In the first step, they will be implemented using the hardware specific approaches VHDL for FPGAs and OpenCL-Code for the GPU implementations. To optimize usability and reduce design time in the future, a universal system generating hardware specific executables from single-source post-modern C++ code, as introduced in [27], will be used. With such an approach, the dynamic execution on the best suitable hardware is tangible.

Finally, an automated long-term data archiving (and publication) in our data publication platform Rodare [28] concludes every workflow cycle. The publication system allows researchers to distribute and to cite their research data with a unique DOI (Digital Object Identifier). The assigned DOI can be used to cite the research data in publications and improve the reproducibility of in our modern research landscape. Combined with the metadata generated and attached at every workflow step, the search in a huge amount of scientific data is as easy as never before.

The TELBE user facility and especially the efficient processing of the related workflows are a first step in building groundbreaking data management systems for big data scientific workflows at HZDR.

VIII. Conclusions

In this paper, we evaluated our workflow application for parallel processing of big data generated at TELBE.

The rapid evolution of high-speed processors has allowed scientists to reduce the time needed to process data from several hours to a few minutes. However, the improvement in the speed of ADC and linear array detectors as in [17][18] dramatically increased the amount of data, and rise the needs for more scalable and distributed solutions that able to process the big data in a reasonable time.

Several factors affect the total execution time of the code which is the amount of raw data, the CPU performance such as its speed, and cashes, the number, speed and caches capacities of the CPU, the speed of the main memory and the storage devices, the network throughput, and the parallel portion of the code.

Developing to a new architecture requires rewriting the whole data acquisition system, although the planned changes in performance would require this even if the hardware were to remain the same. It is changing from post processing system to a real-time acquisition and processing system. Some of the work currently done offline, in which the raw data is processed to extract the arrival time and experimental signal, is being done instead at the time of acquisition before the data is saved. This dramatically reduces the needed network bandwidth for moving the data to storage but also reduces the amount of storage needed for even a few hours of measurements.

REFERENCES

[1] Li, W., (2019). EXTREME DATA Big Data Applications Using Workflows for Data-Parallel Computing. [online] Citeseerx.ist.psu.edu. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.707.2166.

[2] K. Asanovic et al., The Landscape of Parallel Computing Research: A Viewfrom Berkeley, Tech. Report UCB/EECS-2006-183, (2006).

[3] Van der Steen, A. (2008). Overview of recent supercomputers. [online] Computing.llnl.gov. Available at: https://computing.llnl.gov/tutorials/parallel_comp/OverviewRecentSupercomputers.2008.pdf.

[4] Álvarez-Rodríguez, J., Alor-Hernández, G. and Mejía-Miranda, J. (2018). Survey of Scientific Programming Techniques for the Management of Data-Intensive Engineering Environments. Scientific Programming, 2018, pp.1-21.

[5] Czarnul, P. (2018). *Parallel programming for modern high-performance computing systems*. Boca Raton, FL: CRC Press.

[6] "Contact," High-Performance Computing at HZDR - Helmholtz-Zentrum Dresden-Rossendorf, HZDR. [Online]. Available: https://www.hzdr.de/db/Cms?pOid=12231&pNid=852.

[7] Kuntzsch, M., Findeisen, S., Schurig, R., Al-Shemmary, A., Bousonville, M., Czwalinna, M., Golz, T., Schlarb, H., Schmidt, B., Schulz, S., Stojanovic, N., Vilcins-Czvitkovits, S., Gensch, M., Hass, E., Green, B., Hauser, J., Kovalev, S., Lehnert, U., Michel, P., Roeser, F. and Schneider, C. (2019). Electron Bunch Diagnostic at the Upgraded ELBE Accelerator: Status and Challenges. [online] Core.ac.uk. Available at: https://core.ac.uk/display/20520762.

[8] George, P., Strait, J., Dawlaty, J., Shivaraman, S., Chandrashekhar, M., Rana, F., and Spencer, M. (2008). Ultrafast Optical-Pump Terahertz-Probe Spectroscopy of the Carrier Relaxation and Recombination Dynamics in Epitaxial Graphene. Nano Letters, 8(12), pp.4248-4251.

[9] B. Green, "Superradiant Terahertz Sources and their Applications in Accelerator Diagnostics and Ultra-fast Science," Ph.D. dissertation, Fakultät für Physik, Karlsruher Institutes für Technologie (KIT)., Karlsruhe, 2013.

[10] S. Kovalev, B. Green, N. Awari, A. Fisher, N. Stojanovic, and M. Gensch, "High-field high-repetition-rate prototype user facility for the coherent THz control of matter," 2016 41st International Conference on Infrared, Millimeter, and Terahertz Waves (IRMMW-THz), 2016.

[11] S. Kovalev, B. Green, T. Golz, S. Maehrlein, N. Stojanovic, A. S. Fisher, T. Kampfrath, and M. Gensch, "Probing ultra-fast processes with the high dynamic range at 4th-generation light sources: Arrival time and intensity binning at unprecedented repetition rates," Structural Dynamics, vol. 4, no. 2, p. 024301, 2017.

[12] M. Caselle, L. A. Perez, M. Balzer, T. Dritschler, A. Kopmann, H. Mohr, L. Rota, M. Vogelgesang, and M. Weber, "A high-speed DAQ framework for future high-level trigger and event building clusters," Journal of Instrumentation, vol. 12, no. 03, 2017.

[13] A.-S. Müller, "KAPTURE-2. A picosecond sampling system for individual THz pulses with high repetition rate," Journal of Instrumentation, vol. 12, no. 01, 2017.

[14] Zapolnova, Ekaterina, Torsten Golz, Rui Pan, Karsten Klose, Siegfried Schreiber, and Nikola Stojanovic. "THz pulse doubler at FLASH: double pulses for pump–probe experiments at X-ray FELs." Journal of synchrotron radiation 25, no. 1 (2018).

[15] http://flash.desy.de

[16] http://doocs.desy.de

[17] Ordonez, C., Navas, M., and Garcia-Alvarado, C. (2011). Parallel Multithreaded Processing for Data Set Summarization on Multicore CPUs. Journal of Computing Science and Engineering, 5(2), pp.111-120.

[18] Shanthi, M., and Selvi, R. (2006). An Experimental Study of Hyper-Threading Technology for Various Platforms. i-manager's Journal on Future Engineering and Technology, 1(3), pp.26-29.

[19] P. Stoica, H. Li, and J. Li, Amplitude estimation with application to system identification," in Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on, vol. 4. IEEE, 1999, pp. 1777-1780.

[20] K. S. Narendra and A. M. Annaswamy, Stable adaptive systems. Prentice-Hall Englewood Cli s, New Jersey, 1989.

[21] Panomruttanarug, Benjamas & Longman, R.W.. (2006). Frequency-based optimal design of FIR zero-phase filters and compensators for robust repetitive control. 123. 219-238.

[22] L. Rota, M. Caselle, N. Hiller, A.-S. Müller, and M. Weber, An ultrafast linear array detector for single-shot electro-optical bunch profile measurements. : Proceedings of International Beam Instrumentation Conference (IBIC), Sept 14-18, 2014Monterey, USA, TUPD10, 2014.

[23] L. Rota, C. Michele, M. N. Balzer, M. Weber, A. Mozzanica, and B. Schmitt, "Development of a Front-End ASIC for 1D Detectors with 12 MHz Frame-Rate," Proceedings of Topical Workshop on Electronics for Particle Physics — PoS(TWEPP-17), 2018.

[24] Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Mons, B. (2016). Comment: The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*, 1–9. https://doi.org/10.1038/sdata.2016.18

[25] Barisits, M., Beermann, T., Berghaus, F., Bockelman, B., Bogado, J., Cameron, D., Wegner, T. (2019). *Rucio - Scientific Data Management*. Retrieved from http://arxiv.org/abs/1902.09857

[26] E. Deelman and A. Chervenak, "Data management challenges of data-intensive scientific workflows," *Proc. CCGRID 2008 - 8th IEEE Int. Symp. Clust. Comput. Grid*, pp. 687–692, 2008.

[27] Knodel, O., Charaf, N., Stephan, J. and Göhringer, D., "Optimizing the Utilization of Heterogenous Systems Using a Single-Source Approach with Dynamic Resource Mapping", ACACES, 2019.

[28] Helmholtz-Zentrum Dresden-Rossendorf, RODARE – Rossendorf Data Repository. [online]. Available at: https://rodare.hzdr.de/about/.