

Wojciech Wiślicki*, Tomasz Bednarski, Piotr Białas, Eryk Czerwiński, Łukasz Kapłon, Andrzej Kochanowski, Grzegorz Korcyl, Jakub Kowal, Paweł Kowalski, Tomasz Kozik, Wojciech Krzemień, Marcin Molenda, Paweł Moskal, Szymon Niedźwiecki, Marek Pałka, Monika Pawlik-Niedźwiecka, Lech Raczyński, Zbigniew Rudy, Piotr Salabura, Neha Gupta Sharma, Michał Silarski, Artur Słomski, Jerzy Smyrski, Adam Strzelecki, Anna Wieczorek, Marcin Zieliński and Natalia Zoń

Computing support for advanced medical data analysis and imaging¹⁾

Abstract: We discuss computing issues for data analysis and image reconstruction of positron emission tomography based on time-of-flight medical scanner or other medical scanning devices producing large volumes of data. Service architecture based on grid and cloud concepts for distributed processing is proposed and critically discussed.

Keywords: high-performance computing; medical image reconstruction; positron emission tomography.

¹⁾Based on talk given by the first author at the Symposium on Positron Emission Tomography, September 19–22, 2013, Jagiellonian University, Kraków, Poland

*Corresponding author: Wojciech Wiślicki, National Centre for Nuclear Research, 05-400 Świerk-Otwock, Poland, E-mail: wojtek.wislicki@fuw.edu.pl

Tomasz Bednarski, Piotr Białas, Eryk Czerwiński, Grzegorz Korcyl, Jakub Kowal, Tomasz Kozik, Wojciech Krzemień, Paweł Moskal, Szymon Niedźwiecki, Marek Pałka, Monika Pawlik-Niedźwiecka, Zbigniew Rudy, Piotr Salabura, Neha Gupta Sharma, Michał Silarski, Artur Słomski, Jerzy Smyrski, Adam Strzelecki, Anna Wieczorek, Marcin Zieliński and Natalia Zoń: Institute of Physics, Jagiellonian University, Kraków, Poland

Łukasz Kapłon, Andrzej Kochanowski and Marcin Molenda: Faculty of Chemistry, Jagiellonian University, Kraków, Poland

Paweł Kowalski and Lech Raczyński: National Centre for Nuclear Research, Świerk-Otwock, Poland

Introduction

According to the World Health Organization's program on cancer control, more than 40% of cancer cases can be prevented, and an even larger percentage can be cured if detected early [1]. To this end, one of the key practices to be implemented and made a medical routine is the early detection of small lesions and widespread monitoring of the functionality of organs. For these objectives to be

realized, accurate, frequent, and noninvasive medical examinations have to be made available to the public.

Persistent efforts in this direction are being continuously undertaken and are devoted to inventing higher-resolution and cheaper medical scanning detectors. As an obvious consequence of the detectors' refinement in precision and speed, dealing with data streams from such equipment and the need to simulate and optimize them by far exceed earlier experience. This situation has already been observed earlier in high-energy physics and astrophysics, where new-generation detectors and data acquisition electronics were introduced one or two decades ago. In addition, in medicine, it becomes clear that to efficiently handle growing streams of data, one has to equip both the detector developers and medical teams with fast data processing and digital image reconstruction methods and services.

In this article, we outline major computational problems related to data analysis and imaging. After the formulation of these problems in terms of information technologies, we discuss some real-life solutions. In addition to its purely technical side, the problem has other aspects such as requirements of medical data protection. We outline the program of research and prospective service support in medical image processing for the novel solution of the positron emission tomography (PET) based on the plastic time-of-flight (TOF) detector being developed at the Jagiellonian University [2, 3]. This research and support program in the information technology domain is foreseen, both in the near and more distant future, as a major task for the Computing Centre CIŚ at the National Centre for Nuclear Research [4].

Detector simulations, optimization, and data reconstruction

In the case of the PET-TOF scanner, the computing-intensive tasks before the final imaging include simulations

required for device design and optimization, scanner calibration, monitoring, and event reconstruction. Computational and analysis framework J-PET was designed for these purposes [5]. It is based on the BOOST programming framework and general-purpose libraries written in C++ [6], Doxygen code documentation support [7], ROOT analysis, and processing libraries [8]. Extensive simulations, accounting for detailed physical description of underlying processes of interaction of radiation and particles with detector materials, are performed using a dedicated GATE system based on GEANT4 package [9, 10].

The basic problem and computing demands for digital imaging

Two steps are usually discerned in the standard approach to digital representation of physical objects: *modeling* and *rendering*.

Modeling consists of the development of the mathematical representation of an object, be it numerical or closed-form formulas, with three-dimensional (3D) real volumes or two-dimensional (2D) surfaces embedded in a 3D space. Modeling is a very technically demanding process of gaining data from the real world and involves detection and data acquisition (DAQ) techniques. Real objects, being usually irregular and asymmetric, are almost always converted into nonparametric sets of volume elements (voxels). The first raw modeling is performed by the detector and DAQ system, which provides raw data representation of an object. These sets are further reduced and used for image reconstruction. Their careful processing must give unbiased results and be checked by simulations in order not to destroy important details. When transferring data from DAQ to reconstruction processors, bandwidths of the order of gigabits per second are normally sufficient. Because data are not stored on DAQ devices, transfers to servers next to them have to be very reliable in order not to lose data, and therefore, redundant links are needed.

The second step, rendering, is the technique of converting these sets of numbers or formulas into a realistic picture using methods of image reconstruction, simulating photo-realistic effects, combining 2D slices into complete 3D objects, smoothing, etc. Reconstructed data have to be kept in the memory and are not further streamed. In real-life solutions, where time is a critical factor, rendering has to be a CPU-intensive processing, enabling strong data volume reduction (typically factor at least 5) and usually requiring concurrent processing: *threading*,

parallelization, or *vectorization*. The first one, threading, is a decomposition of a process into independent, executed in parallel subprocesses, and for codes foreseen to be executed on a single processor, threading is normally done by compilers. This idea is implemented by switching the processor between threads located in different parts of the memory, depending on the process phase. A more general concept, parallelization, may be implemented either on the single processor or many processors. In the first case, it is implemented as instruction-level parallelism and pipelining. At higher levels, it may be realized as multicore, distributed, or grid computing. A special case of parallelism is represented by vector processing, which is the implementation of the single instruction-multiple data processing scheme.

One of the natural and most promising implementations of concurrent processing is given by the graphic processing units (GPU) with many parallel execution units. Their advantages over CPU are also due to deeper pipelines (thousands of instructions for GPU compared with 20 for a typical CPU) and much faster memory interfaces, as they have to shift around more data. A typical GPU architecture is presented in Figure 1, where basic stages of data transfer and picture rendering are presented. From the year 2000 onward, GPUs tend to outperform traditional CPU servers for many applications where concurrent data processing may be efficiently used, with image processing belonging to this class. Code development on GPU is not straightforward and requires familiarity with dedicated programming environments for heterogeneous computing platforms: *OpenCL* or *CUDA*.

Considering an appropriate processing scheme for medical imaging applications, one has to decide first if it is going to be a localized or distributed computing. For a local model (either a usual CPU cluster or a mixed CPU-GPU server, or even a vector machine), the obvious requirements of data discretion are usually easier to meet. However, these resources have to be large and reliable if big sets of high-resolution pictures need to be obtained almost interactively. This requirement exponentially increases the costs of an overall medical facility, sometimes in contrast to the initial hope that these costs can be kept low. In addition, it is not going to be a one-time investment cost but a continuous exploitation burden, which, in modern high-performance computing centers, even prevails over investments in terms of money spending. The specifics of interactive applications, usually demanding resources in narrow time peaks, frequently render the local computing facilities not sufficiently flexible. All this puts more weight on distributed solutions.

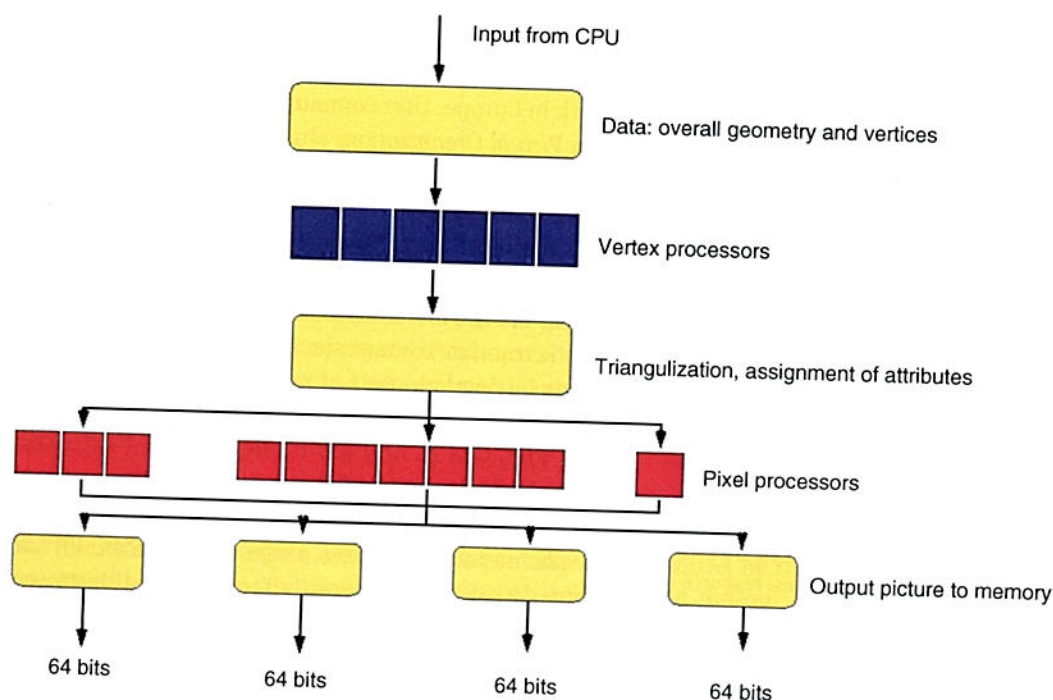


Figure 1 The GPU pipeline of data processing for image rendering, starting from the data stream from the CPU until the final picture output from pixel processors (based on a presentation by Professor Roger Crawfis, Ohio State Buckeyes).

Solutions for distributed medical computing

When choosing the distributed mode, one deals with the following questions:

- Who owns and who administrates computing, memory, and network resources? Two solutions are seen for today's distributed computing networks. In the first one, all contributing parties agree between themselves and nobody waives ownership. In the second one, there is one owner and administrator. The first is cheaper, but the second is more reliable.
- Who pays for the infrastructure and services? It may either be the resource provider who has to find finances outside of the user community or just the users. The first case is rarely met unless public agencies or other wealthy parties support these network and service layers. In the second solution, the service is payable and thus costly for users, but they keep better control on its quality.
- Because medical data are touchy and, in many countries, strictly protected by law, how are data security and reliability going to be ensured?

Basic security can be provided in the usual framework of the *public key infrastructure* that is extensively

implemented and used nowadays, where transferred data are encoded using a public and private cryptographic key pair that is obtained and shared through a trusted authority. In this scheme, typical in academic and research nonsecure networks, usually, neither financial nor legal responsibility for data is assumed and providers work on the best-effort basis. Although technically sufficient, this approach does not meet all the detailed security requirements if not complemented by software requirements, its validation procedures, system quality, and others [11]. For a certified medical data center, a high demand of reliability (>99.99%) and a predefined data cybersecurity with legal and financial responsibilities are specified.

The two architectural solutions for distributed medical computing, already proven to be efficient but still not implemented on a mass scale, are *grid computing* [12, 13] and *cloud computing* [14].

On grid computing, memory and computing resources may be scattered geographically and shared between owners. Resources are interconnected but not managed centrally. Users do not pay for using them. Task-to-resource matching is provided and optimized by one or more central services, which are called resource brokers. The fundamental logical units on the grid are sets of processors and mass storage servers called *computing elements* and *storage elements*, respectively, which

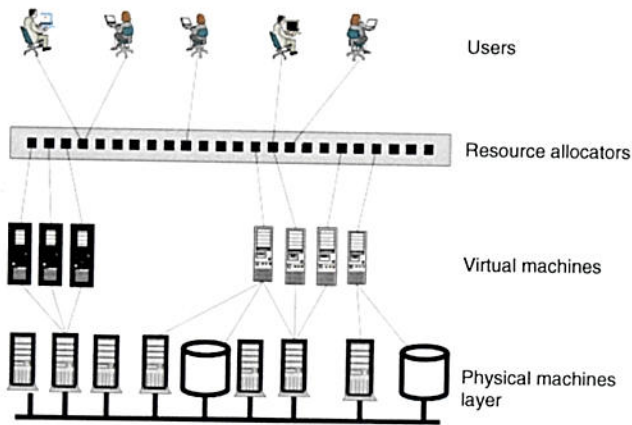


Figure 2 Scheme of the cloud service with the characteristic four layers (bottom-top order): the physical fabric consisting of the computing and storage hardware; virtual machines, which are software computing units not identical to physical machines; resource allocator consisting of basic scheduling and brokering services; and the users.

are equipped with queue systems and managed in sites by their local authorities. Leading examples of large grids are large, multipurpose scientific networks such as the *Worldwide LHC Computing Grid* (LCG) [15] and the *Open Science Grid* (OSG) [16], which are run by organizations in the EU and the USA, respectively, but involve both resource providers and users. Examples also exist for more specialized grids offering medical services [17, 18]. The middleware

and service layers on the grid are put on top of a robust and high-speed backbone networks, such as the GEANT network in Europe. User communities are logically organized in *Virtual Organizations* sharing resources located in many places. Medical applications, in particular medical imaging and image exchange, were, from the beginning, among the most important medical services, e.g., pharmacokinetics using contrast agent diffusion, radiotherapy planning using 3D simulations with GEANT4 and GATE, magnetic resonance image simulations, 3D volume reconstructions using large sets of radiological data. Recently, grid computing in medicine has entered its commercial phase.

The general scheme of cloud computing is presented in Figure 2. Its main advantage is smart virtualization that substantially optimizes usage of resources. Virtual machines are invoked specially for tasks and destroyed immediately after their completion, thus making disks and processors occupied when really needed. Tasks are matched to resources using allocator services similar to resource brokers on the grid.

These two approaches to distributed computing, the grid and the cloud, were initially designed for different purposes and different user communities. However, in recent solutions, computing elements on the grid tend to use typical cloud concepts. The remaining differences between these two originally separate technologies are organization and proprietary issues.

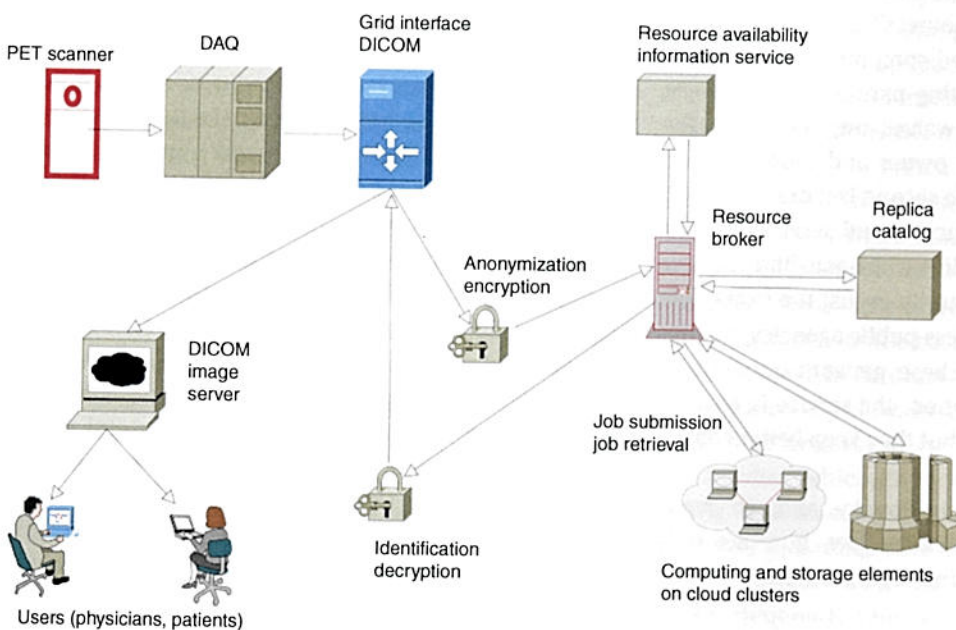


Figure 3 Workflow design for fast medical image reconstruction on the grid.

Workflow design for PET-TOF data processing

In Figure 3, we present the workflow for fast image reconstruction on the grid and the use of computing resources deployed in the cloud scheme.

The DICOM services [19] are dedicated to medical image handling on the grid and are available commercially. Compared with many nonmedical applications, medical data are sensitive and require special security treatment. Therefore, in addition to the usual security measures incorporated on the grid to any data propagated over the network outside of the scanner host site, real medical data need to be anonymized and encrypted and identification keys be stored in secure memory. This functionality is normally provided by DICOM servers, but due to its importance, it is indicated separately in Figure 3.

Successful merging of the original grid and cloud technologies ensures optimal resource usage. For the whole network, it is given by resource brokers, and in local computing elements and storage elements, it is ensured by virtualization techniques. To work in a satisfactory manner and provide interactive image provision, the whole data exchange needs to have a dedicated bandwidth secured at the level not less, and preferably higher, than 1 Gbit/s and appropriate computing resources booked. Virtual machines are invoked by local gatekeeper machines when jobs are submitted.

Conclusions

We outlined the concept of fast and highly efficient data analysis scheme intended for processing data from the TOF-PET scanner. Most of these solutions are already

References

1. Cancer control: knowledge into action. Early detection. Geneva, CH: World Health Organization, 2004.
2. Moskal P, Bednarski T, Białas P, Ciszewska M, Czerwiński E, Heczko A, et al. Strip-PET: a novel detector concept for the TOF-PET scanner. *Nucl Med Rev* 2012;15:C68–9.
3. Moskal P, Bednarski T, Białas P, Ciszewska M, Czerwiński E, Heczko A, et al. TOF-PET detector concept based on organic scintillators. *Nucl Med Rev* 2012;15:C81–4.
4. Świerk Computing Centre at the National Centre for Nuclear Research, 2014. Available at: <http://www.cis.gov.pl>. Accessed: 18 Mar 2014.
5. Krzemień W, Silarski M, Stola K, Trybek D, Bednarski T, Białas P, et al. J-PET analysis framework for the prototype TOF-PET detector. *Bio-Algorithms Med-Syst* 2014;10: 19–25.
6. Boost C++ Libraries, 2013. Available at: <http://www.boost.org>. Accessed: 18 Mar 2014.
7. Doxygen, 2013. Available at: <http://www.doxygen.org>. Accessed: 18 Mar 2014.
8. Brun R, Rademaker F. Root – an object oriented data analysis framework. *Nucl Instrum Methods A* 1997;389: 81–6.

known and partially implemented. We argue in favor of using distributed computing technologies to ensure the speed indispensable for interactive imaging. At the same time, it suppresses entailing unlimited blowup of local computing resources and costs. The implementation of the idea of cheap and finely granulated PET scanner, being a necessary condition for breakthrough in the field of cancer diagnostics, is going to provide a flood of data of the order of tens of gibabits per second or faster. It reminds of the initial phase of modern, high-precision scientific measurements and also contemporary everyday life, where enormous volume of detector outputs almost exceeds our capability to profit from them.

Acknowledgments: This work was financially supported by the EU and MSHE grant no. POIG.02.03.00-161 00-013/09, the Polish National Center for Development and Research through grant no. INNOTECH-K1/IN1/64/159174/NCBR/12, the Foundation for Polish Science through MPD Programme, and Doctus – the Lesser Poland PhD Scholarship Fund.

Conflict of interest statement

Authors' conflict of interest disclosure: The authors stated that there are no conflicts of interest regarding the publication of this article. Research funding played no role in the study design; in the collection, analysis, and interpretation of data; in the writing of the report; or in the decision to submit the report for publication.

Research funding: None declared.

Employment or leadership: None declared.

Honorarium: None declared.

Received January 26, 2014; accepted March 23, 2014; previously published online April 11, 2014

9. Jan S, Santin G, Strul G, Staelens S, Assié K, Autret D, et al. GATE: a simulation toolkit for PET and SPECT. *Phys Med Biol* 2004;49:4543–62.
10. Agostinelli S, Allison J, Amako K, Apostolakis J, Araujo H, Arce P, et al. GEANT4: a simulation toolkit. *Nucl Instrum Methods A* 2003;506:250–303.
11. Guidance for industry: cybersecurity and networked medical devices. Washington, DC: US Food and Drug Administration, 2005. Available at: <http://www.fda.gov/MedicalDevices/Device-RegulationandGuidance/GuidanceDocuments/ucm077812.htm>. Accessed: 18 Mar 2014.
12. Jacob B, Brown M, Fukui K, Trivedi N. Introduction to grid computing, 2005. Available at: <http://www.redbooks.ibm.com/redbooks/pdfs/sg246778.pdf>. Accessed: 18 Mar 2014.
13. Foster I, Kesselman K. The grid 2: blueprint for a new computing infrastructure. Amsterdam, The Netherlands: Elsevier, 2004.
14. Ivanov I, van Sinderen M, Shishkov B. Cloud computing and services science. Berlin: Springer, 2012.
15. Worldwide LHC Computing Grid. 2014. Available at: <http://wlcg.web.cern.ch>. Accessed: 18 Mar 2014.
16. The Open Science Grid, 2014. Available at: <http://www.opensciencegrid.org>. Accessed: 18 Mar 2014.
17. Benkner S, Berti G, Engelbrecht G, Fingberg J, Kohring G, Middleton SE, et al. GEMSS: grid infrastructure for medical image provision. *Methods Inf Med* 2005;44:177–81.
18. Garcia Ruiz M, Garcia Chavez A, Ruiz Ibanez C, Gutierrez Mazo JM, Ramirez Giraldo JC, Pellaes Echavarría A, et al. MantisGRID: a grid platform for DICOM medical images management. *J Dig Imaging* 2011;24:271–83.
19. The DICOM Grid, 2014. Available at: <http://www.dicomgrid.com>. Accessed: 18 Mar 2014.