Univerzita Karlova v Praze
1.lékařská fakulta
Charles University in Prague
1st Faculty of Medicine

Studijní obor: Biomedicínská informatika
Study domain *Biomedical informatics*

Mgr. Tomáš Kulhánek

Využití technologie GRID při zpracování medicínské informace

Utilization of GRID technology in processing of medical information

Dizertační práce
Dissertation

Školitel: Ing. Milan Šárek, CSc.
Supervisor

Konzultant: Doc. MUDr. Jiří Kofránek, CSc.
Consultant

Praha 2015

Prohlášení

Prohlašuji, že jsem závěrečnou práci zpracoval samostatně a že jsem řádně uvedl a citoval všechny použité prameny a literaturu. Současně prohlašuji, že práce nebyla využita k získání jiného nebo stejného titulu.

Souhlasím s trvalým uložením elektronické verze mé práce v databázi systému meziuniverzitního projektu Theses.cz za účelem soustavné kontroly podobnosti kvalifikačních prací.

V Praze, 2015 Mgr. Tomáš Kulhánek

Podpis

Identifikační záznam

KULHÁNEK, Tomáš. *Využití technologie GRID při zpracování medicínské informace*. Dizertační práce. Praha, 2015. 44 stran. 7 příloh. Univerzita Karlova v Praze, 1. lékařská fakulta, Ústav patologické fyziologie. Školitel Šárek, Milan. Konzultant Kofránek, Jiří.

Identification record

KULHÁNEK, Tomáš. *Utilization of GRID technology in processing of medical information*. Dissertation. Prague, 2015. 44 pages. 7 appendices. Charles University in Prague, 1st Faculty of Medicine, Institute of Pathological Physiology. Supervisor Šárek, Milan. Consultant Kofránek, Jiří.

Dedication

To my beloved children Karla and Matěj and my lovely wife Marie.

Acknowledgements

I would like to express my gratefulness to my supervisor Milan Šárek for his support and guidance throughout the beginning of my research. I would also express my gratefullness to the staff of association CESNET who gave me opportunity to work with computing resources connected via high speed academic network, especially Jiří Navrátil. My thanks to colleagues from Academy of Performing Arts, Marek Frič and others with their intention in voice science. I would like to thank my colleagues from Laboratory of Biocybernetics and Computer Aided Teaching who provided me pleasure and inspiring atmosphere for my research and development work during second part of my research. Especially, I would like to thank to my consultant Jiří Kofránek for his support and guidance within the topic of mathematical models and simulation of integrative physiology. Last but not least thanks to the staff and support of Czech NGI department of CESNET, METACENTRUM, for giving support and access to powerful computing infrastructure for scientific purposes.

Název práce: Využití technologie GRID při zpracování medicínské informace

Autor: Mgr. Tomáš Kulhánek

Katedra,ústav: Ústav patologické fyziologie 1.LFUK

Školitel: Ing. Milan Šárek, CSc., CESNET z.s.p.o.

Konzultant: Doc. MUDr. Jiří Kofránek, CSc.

Abstrakt:

Práce prezentuje výzkum a výsledky využití technologií, které umožňují sdílet výpočetní a úložné kapacity a to v oblasti biomedicínského výzkumu. Vedle technologie GRID se rozvíjí virtualizačních technologie (VMWare, XEN, ...), které dodali distribuovaným systémům novou vlastnost zdání vlastnictví a přímé kontroly konfigurovatelné infrastruktury jako služby, jež se dnes shrnují pod společný pojem CLOUD computing. V práci jsou diskutovány teoretické limity distribuovaných systémů a paralelních výpočtů v nich tak i praktické výsledky ve vybraných oblastech. V oblasti výměny medicínských snímků a souvisejících zdravotních záznamů byla ukázána snadná integrovatelnost se stávajícími systémy při respektování požadavků na bezpečnost dat. V oblasti analýzy a zpracování lidského hlasu v reálném čase byla ukázána možnost poskytování nadstandardních výpočetních služeb pro komunitu uživatelů. V oblasti modelování fyziologických systémů je prezentován systém pro odhad parametrů a identifikaci fyziologických systémů komplexních modelů, které by byli obtížně řešitelné za použití klasických metod.

Klíčová slova:

grid-computing, cloud-computing, matematické modelování, výpočetní fyziologie, fonetogram

Title: Utilization of GRID technology in processing medical information

Author: Tomáš Kulhánek

Department: Institute of pathological Physiology

Supervisor: Ing. Milan Šárek, CSc., CESNET z.s.p.o

Consultant: Doc. MUDr. Jiří Kofránek, CSc.

Abstract:

Two work presents results, which are researched in the field of utilization technology who offers distributed computing and storage capacity to exchange medical information, demanding computation and data exchange. Next to the GRID technology. The virtualization technology evolved and were utilized more massively next to the GRID technology and these gives distributed systems a new quality and services called today with the name CLOUD. This work summarizes result of different projects which implements selected technologies in the field of GRID and CLOUD to systems which are used in medical education and research and in neighboring fields. The multidisciplinary projects were solved within the cooperation between association CESNET, First Faculty of Medicine of Charles University in Prague and Academy of Performing Arts in Prague.

Keywords: grid, cloud, computational physiology, phonetogram

Contents

1 Introduction **1**

1.1 Thesis Goal 1
1.2 Thesis Contribution 2
1.3 Thesis Structure 3

2 State of the art **5**

2.1 Computational complexity 6
2.2 Parallelization 9
2.2.1 Programming model 12
2.2.2 Summary 13
2.3 Distributed computing technologies 14
2.3.1 Programming model 14
2.3.2 Service grid-computing 16
2.3.3 Desktop grid-computing 18
2.3.4 Virtualization 19
2.3.5 Cloud computing 19
2.3.6 Application model 20
2.3.7 Workflows and Gateways 21

3 Methods **23**

3.1 Sharing medical information 23
3.1.1 Methods to share medical images in grid 26
3.2 Voice Science 27
3.2.1 Methods for remote analysis of human voice 28
3.3 Computational physiology 29
3.3.1 Modeling methodology 30
3.3.2 Identification of physiological systems 31
3.3.3 Methods for Parameter Estimation 32
3.3.4 Parameter Sweep 34

4 Results **37**

4.1 Medical Images 37

4.2 Remote access to voice analysis 37
4.3 Parameter Estimation 38
4.4 Parameter Sweep 41

5 Conclusion **43**
5.1 Discussion 43
5.2 Future work 45
5.3 Summary 46

Appendix A Processing of medical images in virtual distributed environment **47**

Appendix B Remote Analysis of Human Voice – Lossless Sound Recording Redirection **51**

Appendix C Infrastructure for data storage and computation in biomedical research **57**

Appendix D Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud **63**

Appendix E Modeling of short-term mechanism of arterial pressure control in the cardiovascular system: Object-oriented and acausal approach **69**

Appendix F Simple Models of the Cardiovascular System for Educational and Research Purposes **79**

Appendix G Adair-Based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity* **89**

Bibliography **99**

Chapter 1

Introduction

The *grid-computing* is usually defined as sharing computational and data storage resources across organizational boundaries. In recent years, the development of virtualization technologies enhances the availability of services provided by grid-computing and additionally enabled an evolution of so called *cloud-computing*, which can utilize virtual environment on real powerful computing infrastructure too. Based on the development of technologies and also philosophy of providing them to end users, this thesis focus on multidisciplinary research related to grid-computing as well as to cloud-computing and it's utilization in biomedical research and application related to processing of medical information.

The term "medical information" is too wide and further work focuses on the following selected areas, which were part of: (1) exchange and processing of medical images, (2) analysis of human voice and (3) modeling and simulation of human physiology.

The author's work was published in a series of peer-reviewed papers of international journals and peer-reviewed conference proceedings [1, 2, 3, 4, 5, 6, 7] which are attached into this work as appendices. The author's work and contribution was also presented in international conferences and published in the respective proceedings and transactions [8, 9, 10, 11]. The work was also popularized on the local and regional conferences and their respective proceedings [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. Author contributed to the utility model registered by the Czech Industrial Property Office [24].

1.1 Thesis Goal

The hypothesis stated by this thesis is that the technologies related to grid-computing and cloud-computing may improve processing of medical information to perform demanding tasks which are almost impossible or may need onerous effort to achieve using classical local or institutional resources.

The aim of the thesis is to discuss the hypothesis in different areas of biomedical research and it's application and answer the following questions:

1. Study the latest achievements in the field of exchanging medical images and study possible improvements using the grid-computing and cloud-computing technology.

2. Identify use cases in other fields of biomedicine which are suitable to utilize the power of grid-computing and cloud-computing infrastructure.

3. Develop and test prototype application utilizing grid or cloud technologies.

1.2 Thesis Contribution

The author claims that the following contribution was made to the state of the art of biomedical research related to grid-computing and cloud-computing.

- Proposal of grid infrastructure and pilot implementation of grid-based system of exchanging medical images integrated with existing distributed systems. The results were published as [1] and popularized as [12, 13, 14]. The author of this thesis customized the existing results of Globus MEDICUS project and deployed a grid application in the servers networked via academic network CESNET and integrated with existing regional PACS system.

- Pilot implementation of more generic infrastructure as a service for the community within the biomedical research [3, 19]. The author of this thesis proposed the idea to share the physical resources and to provide virtual environment for specific needs of particular use-cases. With development of virtualization techniques and cloud-computing technologies the pilot infrastructure were tested on examples of selected research projects.

- Proposal of software architecture and implementation of web-based service for real-time remote analysis of human voice. The results were published as [2, 16, 20]. The author of this thesis contributed to the idea of utilizing interactive remote access to the computing environment. For this reason it was identified requirements and customized the existing network protocol to transfer voice signal losslessly. Other co-authors implemented the algorithms and application to analyse voice signal.

- Improved methodology for modeling of complex physiological systems [5, 6, 11, 10]. Author of this thesis contributed to the idea of building complex mathematical models from the basic components and keep them in an understandable and maintainable form utilizing some known good practices (patterns) and prevent bad practices (antipatterns) known from object-oriented programming. Author advised and implemented several basic blocks and models in the hydraulic domain and related exemplar models of the cardiovascular system in Modelica language. The other co-authors implemented the library to model physiology using integrative approach and implemented the complex models integrating different domains together.

- Design and implementation of system to estimate parameters of complex mathematical models to validate or calibrate models of human physiology [18, 4, 8, 9, 22]. Author of this thesis designed and implemented web service to distribute the simulation of mathematical models into remote resource in grid-computing and cloud-computing infrastructure and integrated complex model simulation with numerical identification algorithms. Other co-authors implemented complex models of human physiology in Modelica language and specific numerical methods to estimate model parameters.

- Improved mathematical model of oxygen, carbon dioxide and hydrogen ion binding to Hemoglobin [7]. Author of this thesis implemented this model in Modelica and calibrated the parameters of the model in web based application [4]. Other co-authors analysed and proposed the new mathematical model based on published laboratory experiments to improve theory related to acid-base balance in blood and saturation curve of oxygen binding to hemoglobin.

- Simulation of complex models of human physiology as part of virtual simulator on portable and mobile devices utilizing cloud-computing [9, 21]. Author of this thesis contributed to the idea of hybrid architecture of web simulators - utilizing the infrastructure for parameter estimation to simulate complex model remotely and process/visualize the results locally. Other co-authors implemented complex models of human physiology and specified and implemented simulation scenarios for educational purposes.

- Virtual patient simulator [24]. Author contributed to the development of prototype virtual simulator and their cooperation within teachers module, where progress

1.3 Thesis Structure

This thesis is interdisciplinar, therefore the following chapters will cover the topics not-only from technical and computer-science point of view, but touches some topics related to the medical science mainly human physiology and it's mathematical models and simulations. The chapter 2 provides an overview of the state of the art in the theory of computation, parallel computation, distributed computation and focus on task-parallelization in grid-computing and cloud-computing infrastructure.

Introduction, particular methods and results to selected areas of biomedical application and research are introduced in chapter 3.1 for sharing medical images, in chapter 3.2 about voice science and chapter 3.3 about computational physiology.

The chapter 4 summarizes general results obtained by the research methods in specific areas of biomedical research and applications. The chapter 5 discuss achievements and answers hypothesis and questions stated at the beginning of the work and recommends further direction of the research effort.

The appendices contain the selected papers [1, 2, 3, 4, 5, 6, 7] which are most relevant to the topic of this thesis and which were published in international peer-reviewed journals or in peer-reviewed conference proceedings:

Appendix A is the paper [1] *Processing of Medical Images in Virtual Distributed Environment* published by ACM as part of the proceedings of the 2009 Euro American Conference on Telematics and Information Systems: New Opportunities to increase Digital Citizenship.

Appendix B is the paper [2] *Remote Analysis of Human Voice – Lossless Sound Recording Redirection* published in Analysis of Biomedical Signals and Images. Proceedings of 20th International EURASIP Conference (BIOSIGNAL).

Appendix C is the paper [3] *Infrastructure for data storage and computation in biomedical research* published by Euromise s.r.o. in the European Journal of Biomedical Informatics.

Appendix D is the paper [4] *Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud* published in the IEEE Xplore Digital Library as part of the proceedings of the 2014 IEEE-EMBS International Conference on Biomedical and Health Informatics.

Appendix E is the paper [5] *Modeling of short-term mechanism of arterial pressure control in the cardiovascular system: Object-oriented and acausal approach* published by ELSEVIER in Computers in Biology and Medicine 2014, **IF(2013): 1.475**.

Appendix F is the paper [6] *Simple models of the cardiovascular system for educational and research purposes* published in Mefanet Journal 2014.

Appendix G is the paper [7] *Adair-Based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity* published in Scandinavian Journal of Clinical and Laboratory Investigation 2014, **IF(2013): 2.009**.

Chapter 2

State of the art

Processing of medical information deals with methods that connects different scientific domains, computer science, biomedical engineering and medicine together with a common goal.

Medical informatics (MI) deals with informatics methods used in clinical medicine and healthcare. While bioinformatics (BI) focus more on development and application of new informatics techniques in biological (mainly genomic) sciences. Computational Biology (CB) study the biological systems formalized as mathematical models and using computational methods to simulate and compare with real systems. Recent development in the fields joined a scientific effort to translate the successful techniques into clinical medicine and helathcare as well as to improve other scientific disciplines. Maojo et al. [25] and Martin-Sanchez et al.[26] described relationship between MI and BI and within the new field of Biomedical informatics.

In further chapters all aspects will be touched, the bioinformatics approach to develop new techniques, biomedical approach to deliver such techniques to clinical use as well to the basic research computational biology approach to do basic science.

From computer science point of view, it is assumed that a processing of medical information is in general a computational problem which is understood as a task that can be solved by a computer.

Because some computationally hard problems will be touched in further text whose exact solution need tremendous amount of time or space, therefore next sections introduces briefly theoretical and practical aspects of computation, parallelism, distributed computing. Section 2.1 introduces some important problem classes from the view of computational complexity theory.

Parallel computation may introduce in some condition speedup needed for computing complex problems, the theory is covered briefly in section 2.2.

Distribution of parallel task via computer network to another computers, servers and clusters is covered in section 2.3 with focus on grid-computing and cloud-computing.

2.1 Computational complexity

An algorithm is a set of operation to acomplish the task and solve the problem. There are several ways howto express algorithm, e.g. in text in programming language or pseudocode or flowcharts are used. In further text the kopenograms will be used as graphical language for structured algorithms to supplement UML proposed by Kofranek et al.[27][1].

The computational complexity theory studies and classifies problems into several classes according to the time or space needed by the algorithm solving the problem.

Time complexity of an algorithm is usually denoted by big O notation defined for function f and g and size of input data n that $f(n) = O(g(n)$ if f is not growing faster than g. Formally $f(n) = O(g(n)$ if and only if there exists constant c and positive integer n_0 that for each $n \geq n_0$: $f(n) \leq c \times g(n)$

$O(1)$ usually denotes algorithms that takes constant time regardless of size of the input. $O(n)$ denotes algorithms that takes time is linear. E.g. sequential search algorithm in fig.2.1 need to compare each record with a given key and is used to find some item in an unsorted list or array. Single comparison takes e.g. 0.03 seconds and list has n records, then algorithm will take at worst n steps and time complexity is $f(n) = 0.03 \cdot n = O(n)$. A better approach is usually achieved with B-tree data structure holding sorted list of records and binary search algorithm applied on this data structure takes logaritmic time complexity $O(\log(n))$ which is better than the previously introduced sequential algorithm [28, 29].

Polynomial time algorithms are defined as the one which time complexity can be bounded by $O(n^k)$ for some constant k. Class of problems which are solvable by polynomial algorithm are denoted as *class P* and are recognised as tractable[30, 31] and other algorithms whose time cannot be bounded by any polynomial function are called exponential and are recognised as intractable [32, 31].

E.g. brute-force search is general solving technique that generates all possible candidates of solution and checks if the problem satisfies the problem statement. All the algorithms to solve brute-force search suffers with exponential time complexity$O(k^n)$ even e.g. depth-first iterative-deepening algorithm for brute-force search was shown to be optimal compared to other standard brute-force search algorithm (depth-first search or breadth-first search)[33, 34].

In the past there were identified several class of problems, for which the current best known algorithm is based on brute-force search of all possible values and better does not exists, or it is open question whether it exists. E.g. For the class of

[1] http://www.kopenogram.org accessed March 2015

```
int function SequentialSearch(Array Records, int Key)
{
        for (int index=0;i<Records.Length;i++)
        {
                if (Records[index]=Key)
                {
                        return (index)
                }
        }
        return(-1)
}
```

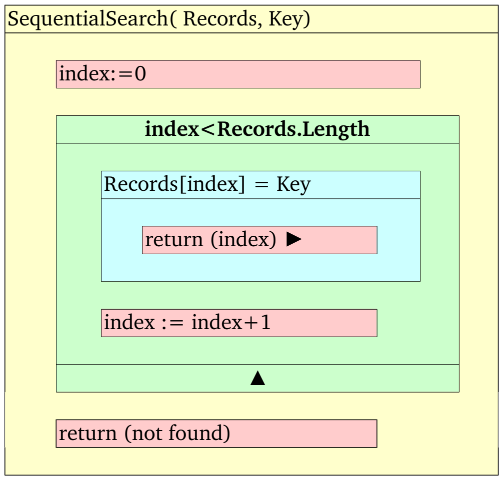


Figure 2.1: Pseudocode (left) and kopenogram (right) of sequential search algorithm with complexity $O(n)$. The red blocks are commands, i.e. setting the index or returning results. The green block represents the loop with entry condition. The index is incremented and in programming languages this can be done e.g. by *for* cycle statement. The blue block is condition (*if* statement), when fulfilled then the inner blocks are executed, in our case the found index is returned. If no record is found, the loop will end with the last index within *Records* and *not found* sign is returned.

NP-complete problems if a polynomial algorithm will be discovered for one problem of this class in future, it was shown that all other NP-complete problems would be solvable by a polynomial algorithm derived from the discovered one[35, 36].

The analysis of complexity of algorithms and problems mentioned above is important for further consequences on computability. For a relatively small input data the exact solution can be found using the known exponential algorithm with current computation power, however, for bigger input data the time needed to solve the problem is far beyond reasonable amount as seen from Table 2.1.

time complexity function \ input size n	10	20	50	100
n	00.01 s	00.02 s	00.05 s	00.10 s
n^2	00.10 s	00.40 s	02.50 s	10.00 s
n^5	01m 40.00 s	53 m 20 s	14h 48m 20s	116 days
2^n	01.02 s	17 m 28 s	35702 years	4.02×10^{19} years
3^n	59.05 s	40 days	2.28×10^{13} years	1.63×10^{37} years

Table 2.1: Computation time of algorithms with different time complexity functions, where one step of algorithm takes 1 milisecond. Examples of algorithm with polynomial time complexity $O(n^k)$ are compared with algorithm with exponential time complexity $O(k^n)$. Note that for the problems with input size of 50 and greater the exponential algorithm runs far beyond the reasonable time compared to the age of universe which is currently estimated to 13.8×10^9 years [37].

If we presume technological update and the computation speed will increace, the effect of technological speedup is visible in table 2.2. The effect on polynomial algorithm is multiplicative, however for exponential algorithm, the technological speedup will increase the size of computable problem only slightly, this is reason why the problems which best algorithms has exponential complexity are denoted as intractable.

	present computer	10 times faster	100 times faster	1000 times faster
n	3600000 $1\times$	36000000 $10\times$	360000000 $100\times$	3600000000 $1000\times$
n^2	1897 $1\times$	6000 $3.16\times$	18973 $10\times$	60000 $31.6\times$
n^5	20 $1\times$	32 $1.59\times$	51 $2.51\times$	81 $3.98\times$
2^n	21 N_{2^n}	25 $N_{2^n} + 3.32$	28 $N_{2^n} + 6.64$	31 $N_{2^n} + 9.97$
3^n	13 N_{3^n}	15 $N_{3^n} + 2.09$	17 $N_{3^n} + 4.19$	20 $N_{3^n} + 6.29$

Table 2.2: Effect of computation speedup. First value is input size of data computable in one hour, second value is speedup achieved compared to the value in first column.

NP-complete problems (currently exponential algorithm can solve them and it was not found better algorithm) are covered in book of M. R. Garey and D. S. Johnson [32]. The whole complexity theory is covered e.g. in book by Ch.Papadimitriou [38] or in a book by M.Sipser [39].

To summarize this section. If there will be technological speedup, this will impact mainly the class of problems which are solvable by polynomial algorithm. For the problems where the computation needs tremendous ammount of time, because current known algorithm is exponential, there can be used non-exact methods to find at least some solution if not the exact one.

- The *heuristic methods* tries to eliminate the number of steps of computation by some implicit or explicit knowledge of the specific problem itself E.g. eliminating solution classes that seems not to go to optimal solution. With combination of brute-search the heuristic methods reduce the size of all possible solution candidates to check.

- The *randomization methods* use non-deterministic methods in some level of computation.E.g. Monte-Carlo method is used to compute problems using pseudo-random generated values and after several iterations statistical methods are used to compute expected value and standard deviation.

- *Restriction on input data* - is another form of using the explicit knowledge of the problem instance ad it may reduce all possible values to be checked.

- *Approximation algorithm* - may find not only some good solution, but can quantify how far from the optimal solution the found is good with some degree of probability.

2.2 Parallelization

If a sequence of instructions can be divided into parts which can be computed independently in parallel by multiple processors, then it is possible to achieve some computation speedup using current computational technology.

We can define a speedup of a computation on P processors as follows:

$$S(P) = \frac{\text{time on 1 processor}}{\text{time on P processors}} \tag{2.1}$$

We can estimate speedup of an computation of a fixed-size problem and then we can ask how can we speedup this problem on P processor. In fig. 2.2 is example of serial and parallel computation of the same problem.

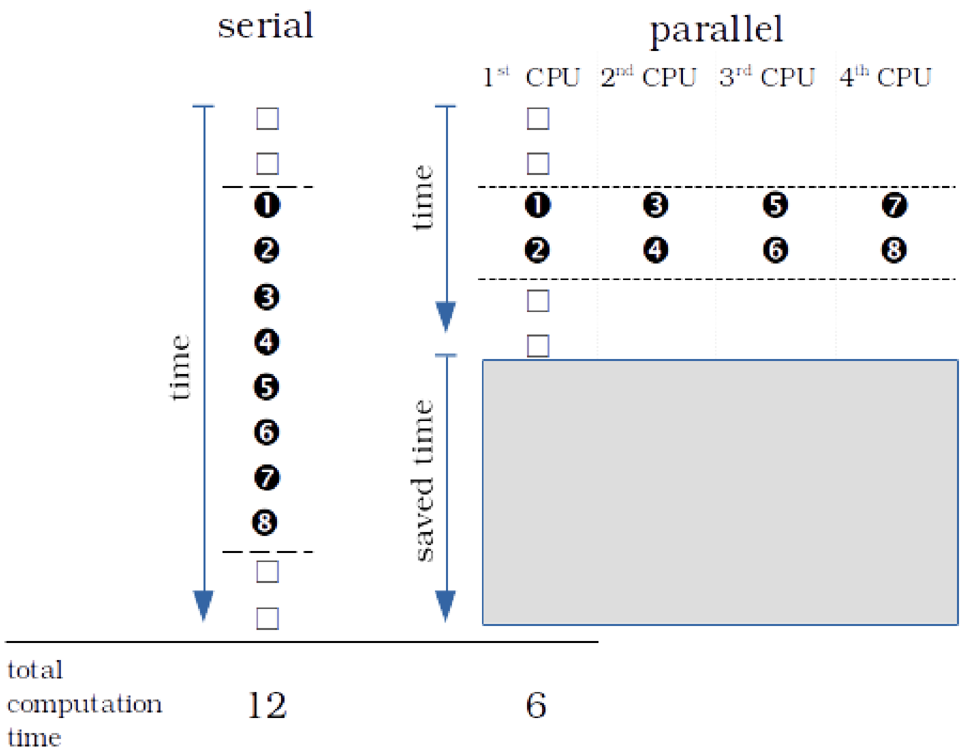


Figure 2.2: Comparison of serial and parallel processing of instructions, the instructions which can be computed in parallel are with numbers. In this case the serial computation takes 12 cycles, parallel computation on 4 CPU takes 6 cycles. The speedup is 2 times. If we have 8 CPUs, then a computation will be finished in 5 cycles and the speedup will be 2.4 times.

Assume $\alpha \in (0, 1)$ as a fraction of the computation in one processor which cannot be parallelized, $(1 - \alpha)$ is fraction of the computation in one processor which can be parallelized by P processors and t is time needed to compute the process on one processor. Assume that overhead of parallelization is small and can be disregarded.

Then speedup can be computed as:

$$S(P) = \frac{t \times \alpha + t \times (1 - \alpha)}{t \times \alpha + \frac{t \times (1-\alpha)}{P}} = \frac{1}{\alpha + \frac{1-\alpha}{P}} \tag{2.2}$$

On unlimited number of processors it can be formulated theoretical upper bound of speedup which depends on α only:

$$S = \lim_{P \to \infty} \frac{1}{\alpha + \frac{1-\alpha}{P}} = \frac{1}{\alpha} \tag{2.3}$$

This is recognized as Amdahl's law [40]. E.g. when a 10% of computation cannot be parallelized($\alpha = 0.1$) then the speedup on 10 processors can be $S(P) = \frac{1}{0.1+0.9/10} = 5.26$ and theoretical speedup on unlimited number of processors is $S = 1/0.1 = 10$. See more at fig 2.3.

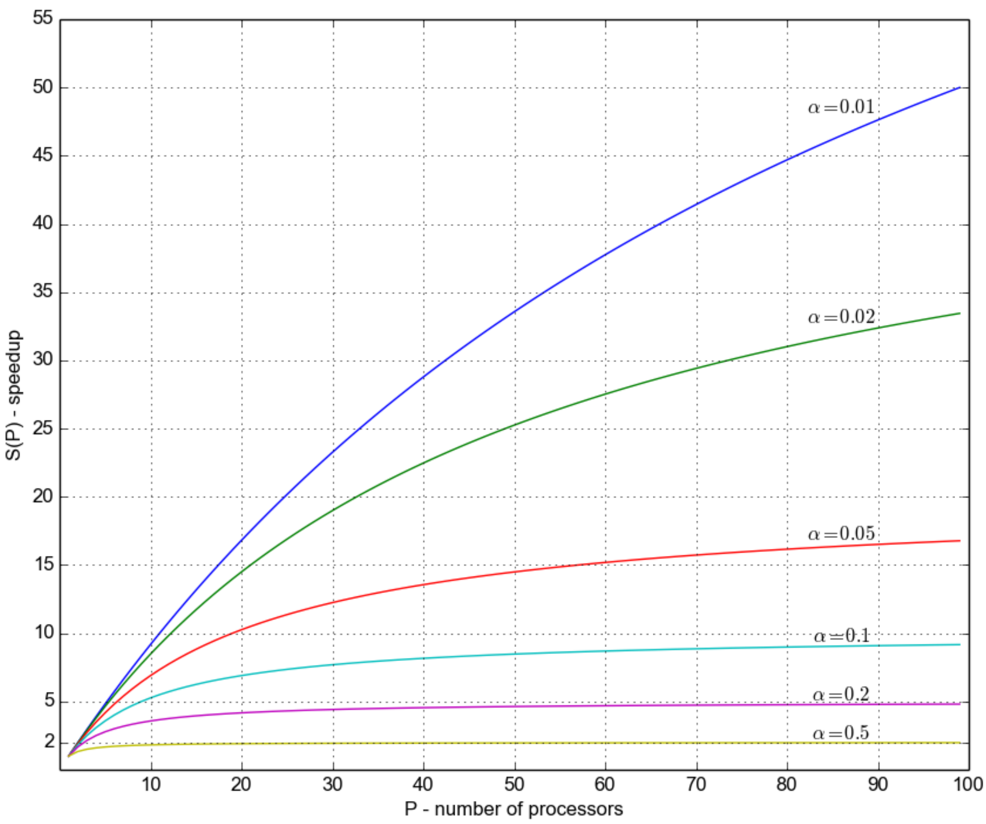


Figure 2.3: Speedup gained on 1 to 100 processors per Amdahl's law for different α values.

However, α could be sometimes hard to estimate and computing the fixed size problem on high number of processors can misrepresent the speedup and performance discrepancy. Therefore, Gustafson reformulated the law and described another approach. Measure the fraction of the computation which cannot be parallelized from computing on P processors and estimate the speedup from how long will such computation take on single processor. Assume that overhead of parallelization is small and can be disregarded. The β is "scaled fraction" of computation on P processors which cannot be parallelized [41]:

$$S(P) = \frac{t \times \beta + t \times (1-\beta) \times P}{t \times \beta + t \times (1-\beta)} = \beta + (1-\beta) \times P \tag{2.4}$$

This law presumes that the fraction β will not change on different number of processors, as seen at fig 2.4.

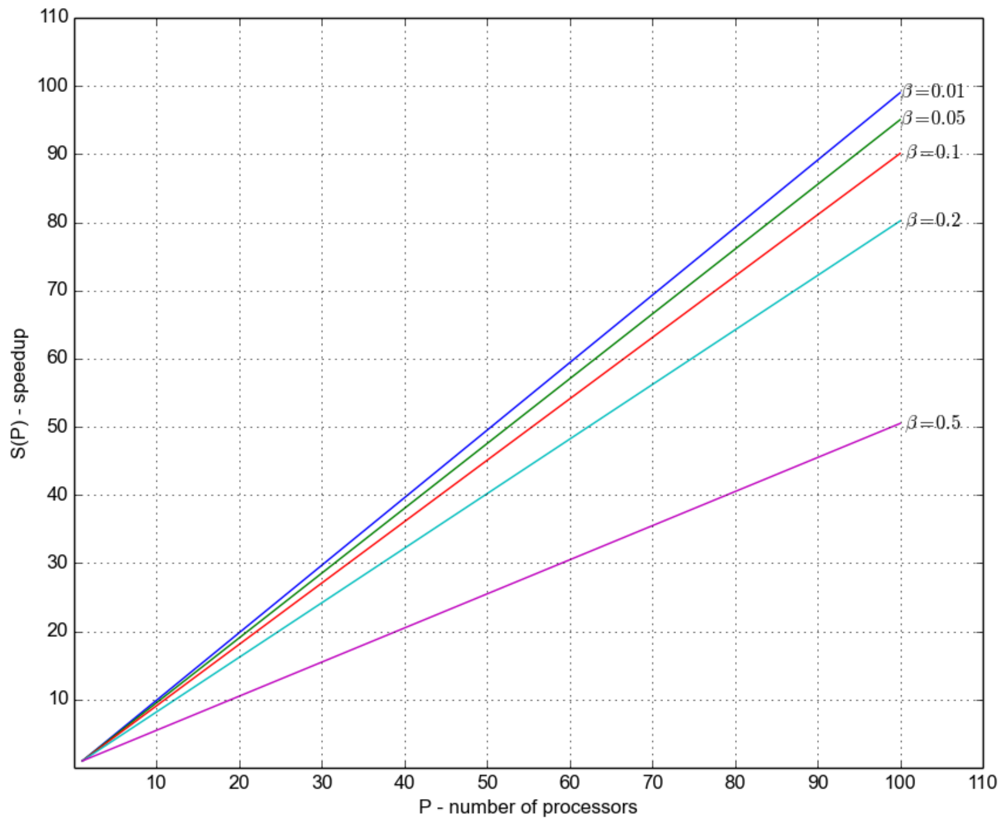


Figure 2.4: Speedup gained on 1 to 100 processors per Gustafson's law for different β values.

Both laws disregard parallelization overhead, however if there is significant one, the speedup of parallelization will be degraded by the overhead. The Amdahl's law (2.3) is argument that the speedup is limited for current sized problem, however, bigger problems can be addressed with higher number of processors and this should be considered with regard to Gustafson's law (2.4).

2.2.1 Programming model

Looking into how the parallelization is realized, there are several levels of parallelism.

- *Instruction level parallelism* - instructions of a program can be executed at the same time, if they are independent. Programs are usually written as a sequence of instruction and it depends on compiler capability to recognize or reorder the instruction so some set of instruction can be executed by a specific processor in parallel. Instruction parallelism started to be systematically utilized in multi-core processor.

- *Data parallelism* - the same operation is performed on multiple data, usually arrays. The instruction is distributed into multiple processors or processor cores and is executed on elements of data structure in parallel. This is currently characteristic feature of *General purpose graphical processing unit*(GPGPU) computing and heterogenoust programmable API such as CUDA[2] and OpenCL[3]

- *Loop parallelism* - the computation may contain a iterative processing of a large data structure. Usually such iterative processing is programmed as a loop and if i^{th} iteration is independent on $(i-1)^{th}$ then the iteration can be executed in parallel by different processors.

- *Task parallelism* - the computation contains parts which are independent of each other. Computation of such parts can be scheduled and distributed into multiple processors and can be computed concurrently. Typical Master/-Worker pattern is realized(master sets up a pool of worker processed and a set of tasks which is distributed to them). Fork/Join pattern - main process forks into several threads executing concurently and waits for their results to join back into a single process which may after some computation again fork.

Looking into the way how the processes interacts, these are the most common forms:

- The *threads* are several concurent execution paths which are independent, but in general share the same memory. It was standardized e.g. as a POSIX threads (Pthreads) and are implemented by standard C libraries in different platforms[42].

- *shared memory*. The *OpenMP*[4] is a shared memory application interface standardized and implemented in by several compilers for C, C++ and Fortran. It

[2] https://developer.nvidia.com/cuda-zone accessed February 2015
[3] https://www.khronos.org/opencl/ accessed February 2015
[4] http://openmp.org/ accessed February 2015

uses also multithreaded model, however programming is task oriented and more abstract[43].

- *message passing*. The *Message Passing Interface*(MPI) is specification for message passing between tasks. This is usually used, when tasks doesn't share the memory [44].

More about parallel programming models can be found in survey by Diaz et al. [45].

2.2.2 Summary

Some algorithms can be divided easily into independent tasks, which can be computed in parallel. If there is no need to communicate among the parallel tasks, such algorithms are called embarrassingly parallel. E.g.

- Operation on matrices [46] are currently used to render 2D and 3D graphics.
- Parameter study, where the same computation is performed using different set of input parameters [47].
- Brute-force search algorithm, where subset of possible candidates for solution are generated and checked in parallel.
- Genetic algorithm and other evolutionary algorithms [48].

In contrast to embarrassingly parallel problems, the opposite are algorithms that are inherently sequential or are believed that cannot be significantly speeded up by parallel computing. The theory about this problems and related algorithm which are denoted as problems/algorithm in class *P-complete* are believed not to get significant speedup using parallel computing, see e.g. book of Greenlaw et al. [49].

The same problem can be solved by different algorithms with different scalability (how they can be speed up by parallel computing) and different effectivity (what is the time complexity) and these both aspects should be considered together [50].

Further aspects of parallel computing can be found in the book of D.Culler et al [51], book of T.Rauber and G.Räuder [52] or earlier book about design and build parallel programs of I.Foster [47].

To summarize this section; parallel computing can introduce speedup on current computational technology and some computation problems may become feasible. Also overhead caused by parallelization and fraction of non-parallelizible parts should be considered as it may degrade expected speedup. In case of exponential algorithm (e.g. for NP-complete problems) the speedup will increase the size of solvable problem only slightly (see table 2.2) and some problems cannot be (or it is believed) significantly speedup by parallel computing. In further text a focus will be given mainly to task parallelism and distributed computing.

2.3 Distributed computing technologies

Distributed computing is based on the idea to spread the computation task into set of computers which are connected via computational network.

Main motivation of using distributed computing technologies are (1) sharing, storing and exchanging data (2) provide and consume computational services (3) access the much higher capacity of storage and computation than available locally.

To manage distributed computing several challenges are maintained such as synchronization(exchange of messages in computation workflow) to achieve e.g. mutual exclusion(when a task needs exclusive access to some resource) and prevent deadlock(no progress is possible) or resource starvation(when a resources - e.g. processor time - is not scheduled for particular task for some reasons and task cannot finish it's computation). Distributed systems offer some sort of fault tolerance (managing fault of a node during computation) or security (encryption of communicating channels and stored data, authentication and authorization to access some resources or data) etc. The topic of distributed computing is covered e.g. in book of Tannenbaum [53].

An extreme example of distributed computing is Internet where the computers are interconnected via TCP/IP protocols, the services are provided by servers (e.g. web servers) with some degree of security and fault tolerance. E.g. world wide web is based on HTTP protocol and HTML language and related technologies, peer to peer services are based on TCP/IP or UDP and streaming of data.

For scientific purposes, the distributed computing infrastructures evolved into set of clusters, computing centers or individual computing resources owned by different subjects. And an effort is continuously done to join such resources into a federation of computational capacity via high speed network to obtain optionally better virtual capacity in case of need. There were formulated some minimal requirements and defined and implemented standards for network protocols and services that a distributed infrastructure should fulfil and provide. Such infrastructures are currently distinguished as grid-computing or cloud-computing infrastructures and the users of it can get access to much higher virtual capacity than accessible locally. Users also may access remotely specialized devices which are not available within their institution.

2.3.1 Programming model

When we look how the distributed computing is realized, the parallel programming model (section 2.2.1) is used and additionally higher level of task interaction is realized via shared distributed file system or messages passed over computer network. Looking into the software layers, distributed computing incorporates usually one or several new layers. Middleware is a software which delivers services or APIs for application and hides specific implementation across heterogenous computing platforms(operating system and hardware).

System Architecture

As the algorithms and programs needed to solve increasing number of problems and changing requirements, the systematic approach and order in several programs and algorithms is adopted to constructing robust software system which will solve the complex problems – software architecture.

The decision about software architecture are made at the begining of a project and are hard to change in implementation is influenced by the experiences that building an application and solution from scratch is too expensive and time consuming. Integrating non-compatible modules might be also time consuming therefore several architecture paradigmas, constraints and patterns are followed to facilitate joining different blocks of software to complex system solving complex problems.

The major software architecture within distributed computing are based on client-server architecture (Example in fig. 2.5), peer-to-peer architecture or more layered architecture patterns.

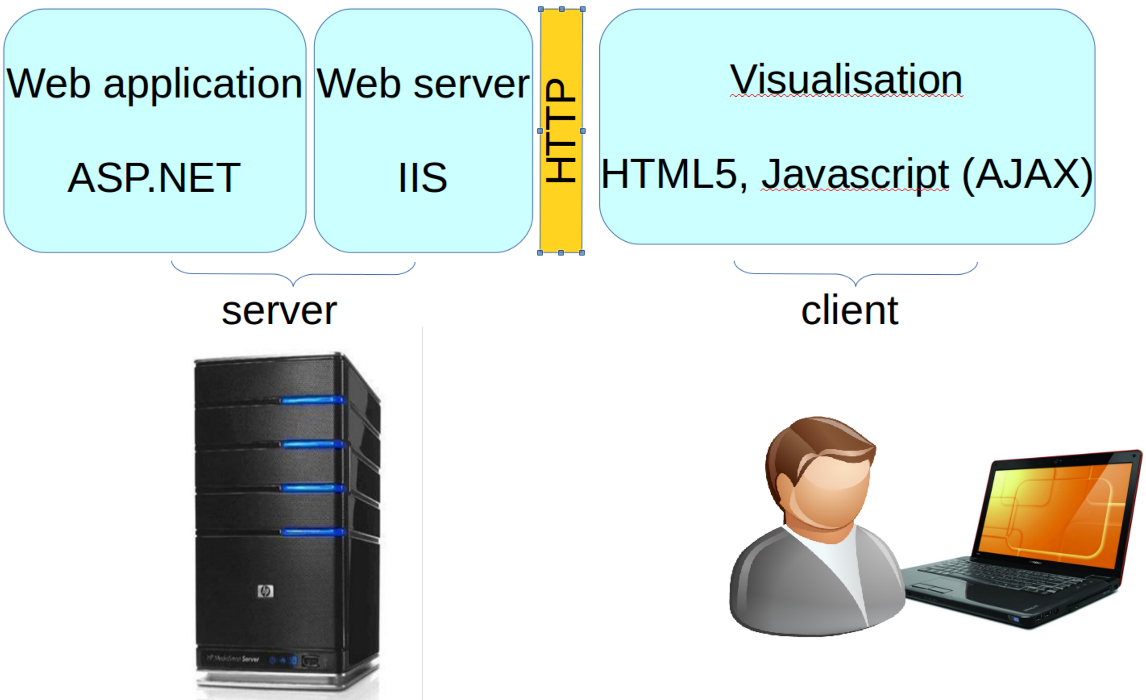


Figure 2.5: Example of client-server architecture involving web server which is middleware between web application and server platform. Client visualization in HTML5 web page communicates with server via HTTP protocol.

Service oriented architecture (SOA) is high level programming model based on self contained units of functionality. SOA introduces a new service layer in the client-server architecture which separate service interface from it's implementation. SOA principles and paradigms can be studied further in book of T. Erl. [54].

Another approach represents objects and data of the system as resources with standard set of operations, create, read, update, delete (CRUD) and Representation State Transfer (REST) specifies several architectural constraints that helps scalability, performance and presents functionality via fixed number of operation and uniform resource location proposed by Fielding [55]. The REST style constraints towards the application is statelessness, resource orientated with uniform interface (CRUD) and hypermedia driven which should facilitate and optimize processing of resources via web based technologies, mainly HTTP protocol.

While SOA focus on application design and easily turning the application objects into distributed services, REST is rather set of constraints to the architecture to handle the issues of distribution within web, as noted e.g. by Vinoski [56].

Software architecture of the enterprise application and distributed systems are studied in general and some repeating patterns are catalogued e.g. by Fowler et. al[57]. Integration patterns are discussed with focus on the ways of connecting heterogenous parts of the system as presents Hohpe et al.[58].

Types of computing infrastructure

When we focus on the architecture of the middleware and philosophy of building a computing infrastructure, these main types of distributed infrastructures are distinguished for scientific computing:

1. *Service grid-computing* is based on the idea that a computing resources (servers, clusters, special hardware) is owned by some organization but may be maintaned by some collective organization with an effort to provide a collection of services in best effort approach.

2. *Desktop grid-computing* is based on the idea to connect generic desktop computers and provide the iddle computation time e.g. as a screen saver or background process to the projects.

3. *Cloud-computing* provides architecture to services, platform or whole infrastructure in a way that an access to them is provided as a service with an impression of sole use it by user or process.

2.3.2 Service grid-computing

The Service-grid computing is based on a basic set of services implemented by a middleware to provide uniform interface for job scheduling and execution within the computing infrastructure. The term *Grid* is used to emphasize the analogy with the electric power grid providing access to electricity [59]. Foster et al.[60, 59] and

Chervenak et al. [61] describe "data" and "computational" grid as shared hardware and software resources that provide reliable, consistent, pervasive and cheap access to high performance computational capacities and effective and reliable execution of requests over data, which needs sensitive controlling of terabyte storages, data transfers to gigabits per seconds over global computer networks and scheduling such data transfer with respect to computational needs. The services provided by grid are either tools or web services following *Service Oriented Architecture*(SOA) for grid computing – Open Grid Service Architecture (OGSA)[62]. The security model and access to the grid infrastructure is proposed and implemented mainly by a mutual authentication between users and resources via public key infrastructure using X.509 certificate [63].

The fundamental part of any Grid is the computer network connecting resources that are distributed in different geographical location, generally Internet.

The national grid initiative in Czech Republic -METACENTRUM[5] is interconnected via high-speed network CESNET 2 utilizing the technology of transfering data over optical cables using Dense wavelength division multiplexing (DWDM) [64]. see fig. 2.6 for further information.

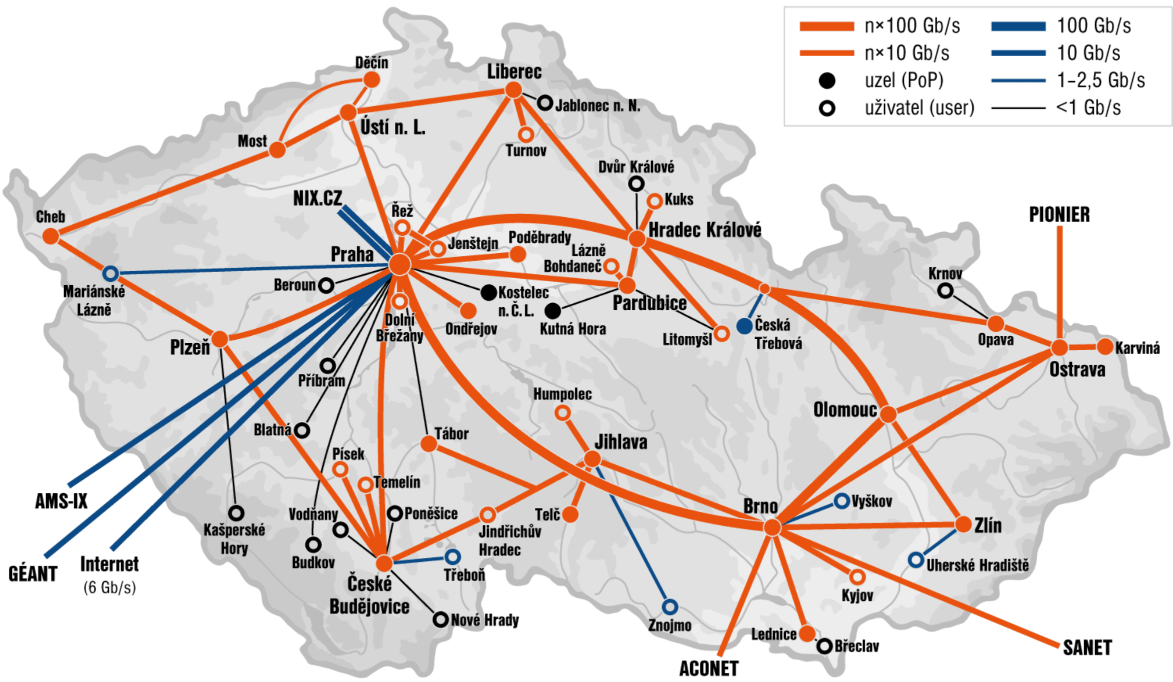


Figure 2.6: CESNET2 network topology as from december 2014 is maintaned by association of universities and academy of science CESNET. Interconnects mainly departments of universities and academy of sciences via rented physical network. It provides connection to general Internet via the czech NIX.cz (Neutral Internet eXchange), AMS-IX (Amsterdam Internet eXchange) and other channels to european research and education network GÉANT. Sources: http://www.cesnet.cz

The administration and maintenance of such networked infrastructures are not

[5] http://www.metacentrum.cz department of CESNET as national grid infrastructure

trivial tasks and they are performed by experts of institutional computing centers or national laboratories and interconnected sites are managed and coordinated at national level or at international level. Such national organizations cooperate with similar national grid infrastructures in other countries. An umbrella organization in Europe –European Grid Infrastructure (EGI)[6]– was established during 2010 and supports integration and coordination activities of NGI accross national boundaries with respect to the scientific collaboration which goes also across national boundaries.

The grow of several production grid infrastructure for science was accelerated in last decade mainly by the needs of experiments of high energy physics to process large number of observed data in a reasonable time [65]. The Worldwide Large Hadron Collider Computing Grid (WLCG) has been designed to store and process almost 30 PetaBytes of data per year in the period of 2009-2013 [66] and is one of the largest grid deployed in the grid infrastructure. Other scientists from different scientific disciplines became users of this powerful infrastructure as well and with development of technologies, virtualization and cloud computing, the infrastructure may employ larger set of application and services and may become attractive for smaller scientific collaboration.

Several grid infrastructures were established based on different grid-middlewares. Condor is one of the earliest effort to provide access to unutilized computers while preserving rights of the owners [67]. glite [68][7], ARC[8] and Globus [69][9] are major grid-middleware operational in EGI.

Currently there is an effort to maintain interoperability between the infrastructures with different grid-middleware to reach the goal of shared world-wide grid [70].

2.3.3 Desktop grid-computing

Joining desktop computers from an individual user to form a voluntary or desktop-grid was popularized by a project trying to identify uncommon signals from space to search extraterestrial intelligence (SETI@Home)[10]. It's based on the idea that a volunteer will download a small client program, which executes in background or instead of a screen saver; it downloads some ammount of raw data from a server on Internet to analyze and sends back the result to the server. In contrast to service grids, the authorization of users can't be so strong for volunteer individuals and some other policies, e.g. redundancy, is implemented to eliminate bad or cheating results [71]. After the success of the SETI@Home there were built general-purpose

[6] http://www.egi.eu
[7] http://glite.cern.ch
[8] http://www.nordugrid.org/arc/
[9] http://toolkit.globus.org/toolkit/
[10] http://setiathome.ssl.berkeley.edu/

frameworks to facilitate development of projects that will use similar philosophy of computing on desktop computers connected via Internet such as BOINC [72], SZTAKI extension to BOINC [73, 74], XtremWeb [75] and others. Currently there exists lot of similar projects gaining computer power as the SETI@Home project. E.g. the LHC@Home and it's successor LHC@Home 2 projects were established and used to execute some selected tasks of the LHC project on the desktop grid infrastructure [76, 77].

The desktop grids and service grids are two different approaches to gather computing power from large number of computing resources and e.g. the EdgE project achieved interoperability between these two worlds so the tasks using the first type of grid infrastructure can be scheduled in a second type of infrastructure [78, 79].

2.3.4 Virtualization

Virtualization technology separates the physical hardware layer from the software environment emulating a new virtual hardware layer. The hypervisor manages the guest virtual machines and translates the I/O from virtual device to physical device and instructions from virtual CPU to physical CPU. This introduces some overhead and performance degradation of virtual system compared to physical one. However, with recent virtualization technology introduced several techniques which reduces overhead and eliminate specific hardware features and instructions in OS which are hard to virtualize [80, 81]. Thanks to them, a virtual environment fine tuned for an application can be executed on almost any hardware and platform and virtualization becomes part of the solution to execute jobs of desktop-grid or service-grid computation on different physical platforms [82]. Currently there exists several commercial, free or even opensource virtualization implementation which are provided by different vendors and it's hypervisor - VMWare, XEN, KVM, VirtualBox etc. And several vendors provides provisioning of virtual infrastructure for computing which are now distinguished as cloud-computing.

2.3.5 Cloud computing

In contrast to grid-computing where user scheduled jobs accessing shared environment and might be influenced by other users or by the environment, the cloud-computing provides access to a virtual software, platform or whole infrastructure so the user or process has impression of sole use. Virtualization techniques enabled expansion of cloud-computing mainly on infrastructure which was built for another purpose and can be rented in times when the primary infrastructure is not fully utilized [83].

Cloud-computing can be characterized as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [84].

Cloud computing makes computational power and storage as utility or commodity that can be rented. It evolves in commercial area to facilitate scaling up the business needs with computational demand with current commercial clouds such as Amazon EC[11], Microsoft Windows Azure [12], Google cloud[13] and others.

Cloud computing in research infrastructures were recently deployed next to the already existing grid-infrastructures and utilizes the same hardware resources. Some methods to integrate grid-computing and cloud computing were introduced [85]. Access is provided to the same users of grid-infrastructure and currently most used platforms are OpenNebula [86] and OpenStack [87].

An interoperability among cloud providers and standardization on cloud-computing, virtualization and related technologies is important as it would keep users from being locked into a specific cloud provider[88].

2.3.6 Application model

The application computed within grid or cloud can be characterized by the quantity of tasks being performed, the size of input data and of the communication needs to be done between concurrent tasks. Grid-computing infrastructures were primarily utilized for computation, which tasks take long time, were relatively loose coupled and resources were used over long period of time. Performance/capacity is usually mentioned in operations/CPUs per month or year and for such computation the term High throughput computing (HTC) is used.

While HTC takes long time, the High Performance Computing (HPC) is usually characterized by computing the problems which have small number of tasks and are relatively tightly coupled and can be executed in highly parallel environment. Performance is measured usually in operations per second [89, 90]. The grid infrastructure can involve HPC servers or clusters, thus a job or tasks that requires such HPC hardware are scheduled and executed there.

Many Task Computing(MTC) aims to bridge HTC and HPC, while the computation usually takes shorter time, the data exchange is in MB rather in GB, performance is measured in tasks per seconds rather than jobs per months or years and involves computing relatively much more heterogenous problems which are not "happily" parallel. However, middleware for HPC or HTC which are present in grid-computing infrastructures may introduce some shortcomings, therefore a prototype task execution framework suitable for MTC was proposed and implemented

[11] http://aws.amazon.com/ec2/
[12] http://azure.microsoft.com/
[13] https://cloud.google.com/

[91, 92, 93]. The MTC seems suitable to be performed on cloud-computing technologies, which introduces some benefits over classical grids, however such clouds should be oriented for HPC systems and generic public cloud may introduce low performance than expected [94].

2.3.7 Workflows and Gateways

A workflow is an abstract description of the process of computation and data manipulation specified by an expert to express what should be done within the distributed system. It automates the process of computation by composing data manipulation steps and tasks and solving failure events.

The workflow can be encoded in any programming or scripting language, however some higher level languages evolved. In business domain a Business Process Execution Language (BPEL) is an OASIS standard and becomes one of the most used language for describing workflow of orchestration of web services and transaction steps [95]. In scientific domain different workflow systems are operational including BPEL with different capabilities. Taxonomies of some existing workflow systems were published by Yu, Zhao et al. [96, 97, 98]. Workflows in cloud computing are covered by web technologies programming languages (Javascript) [83].

The workflow system which implements the concrete workflow language is usually tightly coupled with the specific grid-computing or cloud-computing infrastructure.

To connect different grid-infrastructures a mutual workflow management system is used to integrate them as proposed by Kacsuk et al. [99, 100] or an interoperability is solved by separating abstract workflow representation and concrete implementation showed on selected existing workflow systems introduced by Planensteiner et al. [101].

Scientific gateway incorporates higher level services for specific scientific community e.g. as a web portal or desktop application to control the process of computation via a workflow[102]. For building the scientific gateways several frameworks were developed e.g. Apache Airavata [103, 104] or WS-PGRADE/gUSE[105]. And the concrete instances are available for broader area of scientific community.

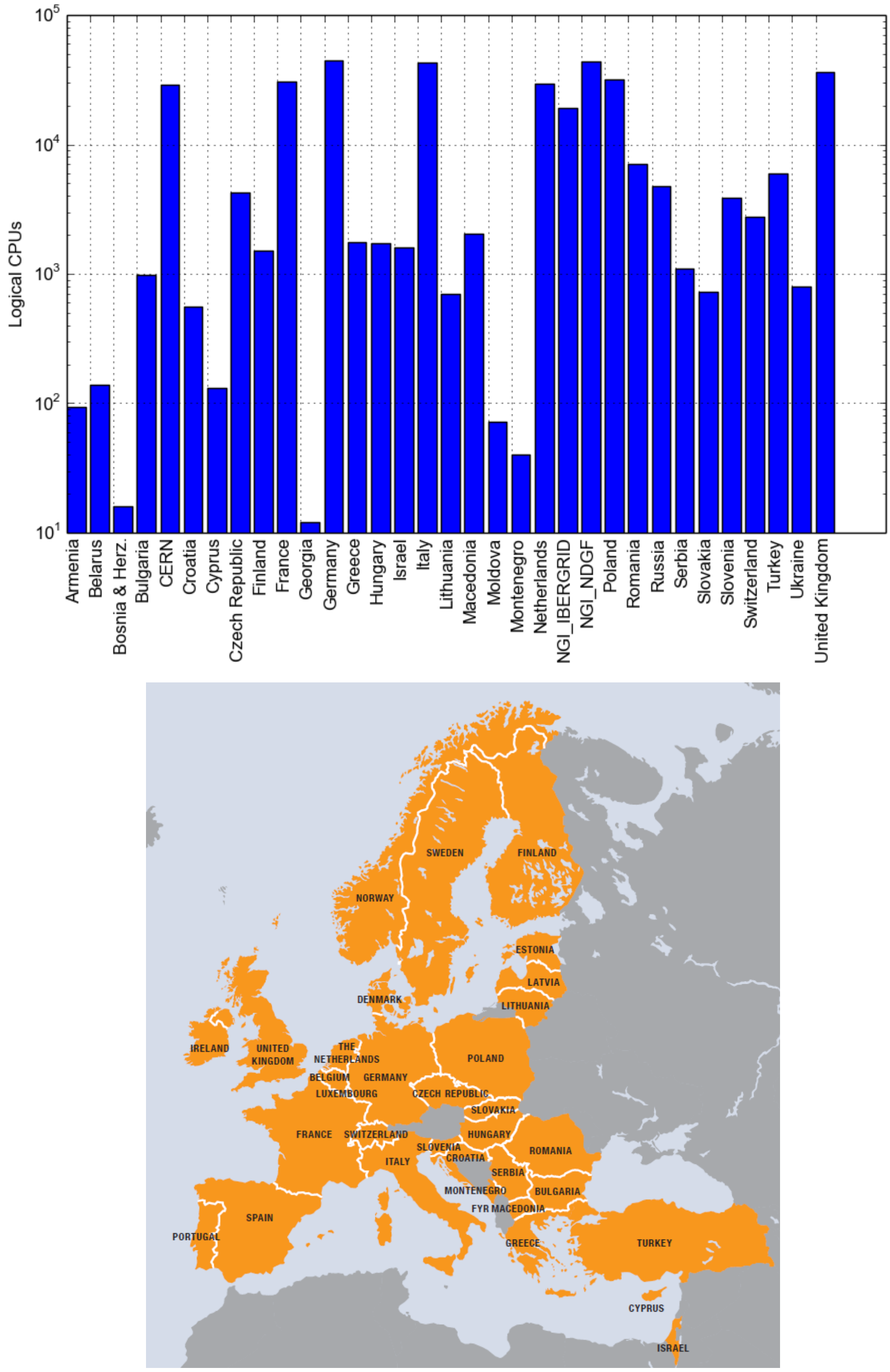


Figure 2.7: Countries involved in EGI and number of CPUs within EGI infrastructure in 2012, the total logical CPU capacity at the end of 2012 was 349,720 cores(in graph per countries). CPU capacity was 433,957 cores in March 2014. Sources: EGI Compendium 2013, EGI statistics at http://www.egi.eu.

Chapter 3

Methods

The general issue of utilizing grid computing or cloud computing infrastructure is to select appropriate method to integrate domain specific computation into the grid or cloud infrastructure of a concrete provider.

A lot of tools are already available within current grid infrastructure including open-source or licensed software for computation. A list of available application is usually given by the local scientific provider[1] or application database are available in broader environment e.g. in EGI.eu application database[2]. Additionaly workflow systems and scientific gateways mentioned in section 2.3.7 tries to hide the complexity of grid-computing or cloud-computing infrastructure and may be used to integrate specific domain too. The programming model of parallel computing and/or distributed computing (in section 2.2.1 and 2.3.1) needs to be followed when designing new application utilizing benefits of grid-computing and/or cloud computing.

The general approach to port application to grid infrastructure is to automatize what can be automatized, i.e. make scripts, configure system, prepare some UI, integrate with existing applications, utilize protocol compatibility etc. An effort to obtain first results is high, however for further computational request, the prepared templates, scripts are reused and effort is much lower.

3.1 Sharing medical information

Use cases related to digital medical images involves the image acquisition, preprocessing, storing and searching. Clinicians use patient image mainly for visualization and diagnostic purposes. Computer assisted methods facilitate the diagnostic process and involves image enhancement (to reduce image noise and increases the

[1]applications available in CESNET METACENTRUM https://wiki.metacentrum.cz/wiki/Kategorie:Applications accessed February 2015

[2]https://appdb.egi.eu/ accessed February 2015

contrast), image segmentation (to separate different types of structures from background and from each other), quantification methods (to determines the structure shape, size, volume), registration methods (to process and join multiple different images into one). Comprehensive concepts and digital techniques in medical imaging are presented e.g. in book edited by I.N.Bankman[106].

Acquisition of the medical image is covered with different modalities (different types of equipment and sensors) by radiologists or other specialists. DICOM[3] format and protocol becomes the de-facto an industrial standard to exchange medical images electronically and picture archiving communication systems (PACS) holding the acquired DICOM images with metadata and description noted by experts are currently part of information systems in hospitals. See the typical workflow of medical image in hospital in fig. 3.1.

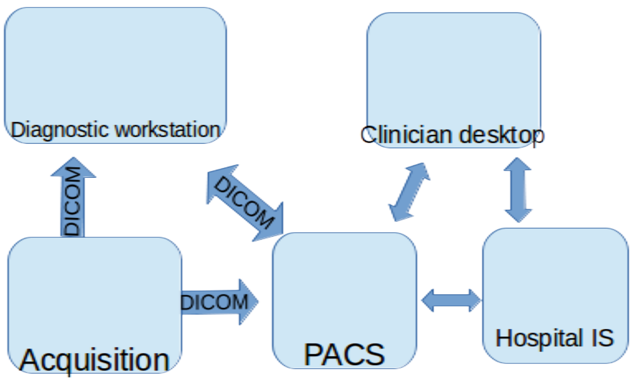


Figure 3.1: Typical workflow of medical image in hospital. Data acquisition is made by modalities (magnetic resonance, ultrasonography, X-ray radiography, etc.) and using DICOM format and protocol it can be directly transferred and visualized by diagnostic workstation. With metadata filled by an expert physician the image is stored in PACS. Other desktops within hospital can retrieve the image and review the report. The hospital information system may be involved in other workflows and communicate with other formats and standards (HL7,...)

As the data processed in hospital information systems contains sensitive information of real patients, these are protected and processing and storing is regulated by the national or international laws or agreements. Development of telecomunication and network technologies a enabled telemedicine – providing healthcare over remote distance. It requires to share and exchange sensitive data of real patient among different helthcare providers and additionaly such data may be very valuable for further research. Security and encryption should be addressed and DICOM

[3] DICOM: http://dicom.nema.org/ accessed January 2015

standard itself doesn't solve security issues appropriately, thus encyption during transferring the data over computer network must be ensured by other techniques.

In the Czech Republic, there exists several projects in production interconnecting different hospitals, clinics and other healthcare organization to exchange medical images. Project ePACS allows interconnecting each participant's PACS system via dedicated VPN channel to the central node and exchange of medical images are realized by routing the data flow from one VPN channel to the other[4]. Another approach is used in the project MEDIMED held by Masaryk University in Brno. Instead of dedicated VPN channel, they use SSL encryption over standard TCP/IP communication and regional hospitals and healthcare providers are interconnected via the MEDIMED servers [107].In other countries, there were tested cross-border teleradiology in projects Baltic e-health, R-Bay and others [108, 109]. These projects are focused on sharing the medical images and other knowledge and information.

Access to the wide range of medical images is needed for research of new processing and diagnostic methods, rare diseases, developing new detection algorithm etc. DICOM records "de-identified" (identification of patient records are deleted, only date of the bird and other data are kept) or anonymized (additional information are manipulated to prevent disclosure) for research purposes to protect sensitive personal data, but keep important information for research purposes. The Globus MEDICUS project published by Erberich et al.[110, 111] is based on Globus Toolkit middleware to federate clinical and research application via a grid-computing infrastructure. Currently the project is hibernated since 2008 and no further development was published[5]. Similar effort was done with a project Medical Data Manager which uses gLite grid middleware published by Duque, Montagnat et al.[112, 113] [6] or MediGRID project published by Krefting et al.[114, 115]. Additionally to the sharing medical images, processing of images within selected use-cases supported by the grid-computing infrastructure is introduced[115]. Health-e-child project aimed to interconnect research institution and hospitals in UK, France and Italy for the purpose of grid-based healthcare platform for pediatric health-care [116]. Neurist project developed architecture and connects clinicians and researchers to improve research and treating of cerebral aneurysm to provide tools to analyze and interpret patient data and researcher can have access to set of aneurysm data, published by Benkner et al.[117]. SEAGRIN research project aimed to share knowledge mainly for educational purposes in semi-formally described semantics and such proposal and implementation was published by Kuba et al.[118].

Storing the sensitive medical information even de-identified or anonymised is

[4]ePACS:http://www.epacs.cz, accessed January 2015
[5]https://dev.globus.org/wiki/Incubator/MEDICUS accessed February 2015
[6]http://modalis.i3s.unice.fr/softwares/mdm/start accessed February 2015

usually restricted and this lead to an idea to store such information within trusted institution e.g. hospital and move and facilitate deployment of the grid services storing medical data to that institution. E.g. pre-installed virtual machines can contain grid-services and deployed in as a sealed grid as proposed by Kuba et al. [119].

To summarize this section, digital medical image acquisition, store, exchange and processing became common in the past years and is currently using distributed computing techniques. There are several efforts to implement medical data management within grid or cloud infrastructure for research purposes and integrate them with the production infrastructures. Security is solved by authentication, authorization mechanism as well as by encrypting the data and/or de-identification or anonymization but keeping minimal information required for research purpose. A related question is how easily the previously mentioned grid-based technologies can be integrated with current systems in hospitals or institutions. The following section describes selected methods used to integrate a pilot deployment of Globus MEDICUS with current regional system for exchanging medical images - MEDIMED.

3.1.1 Methods to share medical images in grid

Globus toolkit belongs to the group of most used grid middleware (see section 2.3.2). The core services included in Globus Toolkit are GridFTP – grid extension to file transfer protocol(FTP) implements strategies such as *stripping data* into multiple pieces, *parallel transfer of data* utilizing stripped data parts to be transfered via different channels, *partial file transfer* some application may not need to acess the whole file, but a smaller portion of it. etc. [120, 121]. Other core services are Replica Location Service aiming to localize data, Globus Resource Allocation Management (GRAM) provides web service and proxies to the lower level job schedulers implementation [120].

Next to core services, the domain specific services might be implemented for the purpose of application using the open grid service architecture (OGSA). Globus MEDICUS [110, 111] contains a DICOM Grid Interface Service (DGIS) and integrates the open source PixelMed™ Java DICOM Toolkit[7] into a web service communicating via DICOM protocol and on the other side it forwards the queries to underlying services within Globus toolkit.

DGIS behaves as a gateway to a grid infrastructure. Because communication via DICOM protocol is not secured, the DGIS is recommended to be installed on the location of the PACS system or DICOM ready modality or software. When a DICOM study is uploaded into DGIS, it is anonymized and stored and a record is made into another services Meta Catalog service which resides in the same domain or any-

[7] http://www.pixelmed.com/ accessed February 2015

where in grid accessible via Globus Toolkit. Such anonymized database of DICOM records can be used to query via DGIS interface and to e.g. integrate with web based application showing records for research purposes, authentication and authorization can be done in this level. To integrate this system with existing system for sharing the medical images (e.g. the MediMed project[107]) the special client software "RediMed console" needs to be installed next to the DGIS and configured it as a local PACS system whose records might be exchanged to other MediMed participants. The results of this particular deployment and integration is in section 4.1.

3.2 Voice Science

With introduction of objective data analysis and laryngoscopy methods the voice science emphasized the cooperation among laryngologist, speech pathologist and voice teacher. The human voice ranges from 50 Hz to something about 1000 Hz, but there are large individual variation. For analysis of digitally recorded voice, either habitual or singing, the Discrete Fourier Transformation(DFT) is used to produce frequency and amplitude analysis of recorded input voice samples. One of the most used class algorithm to compute DFT is class of Fast Fourier Transformation with computational complexity $O(n \log(n))$ [122, 123]. The result of analysis can be visualized in a voice range profile and there can be seen significant difference between untrained and trained voice as well as quantitatively seen some disorders [124, 125].

Other methods to analyse vocal chords is laryngoscopy. The videostroboscopy and high speed video in laryngoscope methods produce video for analysing the real movement of vocal chords. The videokymography method introduced by Švec et al. complements the videostroboscopy and allows to visualize and analyze movement of vocal cords recorded by high speed camera on standard TV or monitor with an artificial image built from recorded sequence of selected section [126, 127].

In case of recorded sound and further analysis there is a question about how such a service can be integrated in grid-computing or cloud-computing environment to provide access to a complex application for non-technical voice specialists. Additionally, the analytical software was already developed and calibrated for selected sorts of microphones in MS Windows platform [128, 129]. Therefore I proposed and implemented a method that provides access to the analytical software remotely. The section 3.2.1 describes how the analytical software was customized with a remote desktop protocol (RDP). Results are described in section 4.2. Similar approach might be used for processing the video recordings from laryngoscope, however, the practical limits are discussed in section 5.

3.2.1 Methods for remote analysis of human voice

Terminal access to some remote computational capabilities, e.g. remote command-line or remote execution is another integration strategy to some remote infrastructure. Secure Shell (SSH) is used to establish secure channel via unsecured network (e.g. the Internet) from SSH client to SSH server and it is basic method to access grid-computing infrastructure.

Remote Desktop Protocol(RDP) is a proprietary protocol for desktop sharing developed primarly in Microsoft Windows platform, however, today clients and servers exists for several other platforms. Next to remote command-line, remote execution it allows to access remote graphical desktop environment. The software for parameterized Voice Range Profile (ParVRP) and Voice Range Profile in Real time (RealVoiceLab) was already developed and calibrated for selected sorts of microphones in MS Windows platform by Fric et al.[128, 129]. The implementation is done in MATLAB environment utilizing Signal Processing Toolbox[8]and compiled with MATLAB Compiler and distributed as an executable.

Instead of migrating the application into some compatible platform for grid-middleware, a virtual machine was introduced and access to the software is provided via RDP protocol. RDP itself contains redirection of several services, e.g. sound recording or drive access. Because the default sound recording redirection introduces some sound degradation without control, I proposed, implemented and integrated the custom RDP plugin with the ParVRP and RealVoiceLab software to redirect the sound recording without loss of information. Technical details are in Appendix B.

The computation of frequencies and amplitude from the recorded samples utilizes Fast Fourier Transformation which has time complexity $O(n \log(n))$ and current implementation are fast on any even modile devices. The benefit from deploying such application in distributed infrastructure is immediate access to updated software and a collection of anonymized records of voice samples with analyzed results for further research and education purposes.

This type of application can be packaged as virtual machine template and configured within different types of cloud infrastructures and together with a script or web portal the on-demand deployment can be automated. The client part (RDP client) needs to connect to the appropriate instance. The results of such deployment are discussed in section 4.2.

[8] http://www.mathworks.com/products/signal/ accessed February 2015

3.3 Computational physiology

A mathematical formalization of the fundamental knowledge and relation among biological system - mathematical model - is used as a base abstraction to utilize current discoveries of the genomics and proteomics and formalize the knowledge and construct a "Physiome Model". Model by it's definition is simplification of the complex reality.

Constructing the models and integrating them into complex entity which can be used for further purposes is schematically illustrated in fig. 3.2. The measurements are done in laboratories or in hospitals. Lumped parameter models are usually represented as ordinary differential equations and differential algebraic equations and characterize the reality as topology of discrete elements. The imaging methods for processing and analysis (section 3.1) are used to construct 3D models from segmentation and generating of mesh representation connected to physical principles.

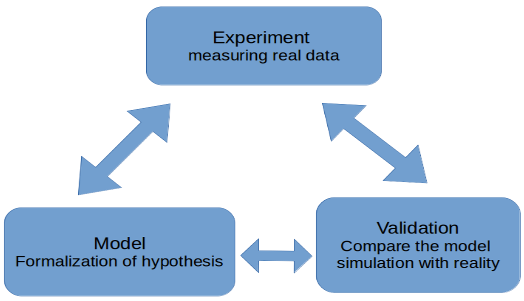


Figure 3.2: Schematic illustration of scientific process. The experiments produces data which are interpreted and hypothese is formalized as a model. Validation compares the model simulation with experiment, if model satisfies the criteria - is in agreement with real experiments, then the validated model can be used for other purposes.

Application of the mathematical modelling techniques towards the biomedical research is sometimes called as systems biology approach combining the reductionism and integration as denoted by Kohl et al.[130]. Application towards the clinical praxis include the quantification of the diagnostic index or treatment strategy and it is a goal to develop tools, database models and methods of several Physiome projects, e.g. VPH-Physiome project presented by Hunter et al.[131].

One of the earliest complex and integrative modelling effort was a model of circulation and it's regulation published by Guyton et al. in 1972 [132] which via derivative and technological upgrade continues as "Human Model" or "HumMod" introduced by Hester et al. [133, 134] with a focus on integration effort. Different approach of modelling human physiology is a database of smaller models focusing on some particular physiological phenomenon. E.g. the NSR Physiome project introduces JSIM[9] Java based simulation system to support modeling in physiology. Repository of several hundred of models were published using this system [135]. The similar effort is done by IUPS Physiome project and repository of models are based on XML standard languages CellML and FieldML [136, 137]. The Systems Biology Markup Language (SBML) is used for modeling biological system at the level of biochemical reaction and regulatory network and another database collects several hundreds of curated and non-curated models [138, 139].

JSIM, CellML, SBML or HumMod are domain specific languages and the tools able to work with them are primarily developed within physiological or systems biological communities. Other authors use commercial or industry standard tools for mathematical modelling and computing. E.g. Kofranek et al. describes Guyton's 1972 model in MATLAB® Simulink [140] and the derivative HumMod in acausal object-oriented Modelica language [141, 10]. Fernandez et al. describes models of cardiovascular pulsatile system using MATLAB Simscape [142] and recently in Modelica [143].

Thus there is an open debate whether in-house domain specific language and tools like JSIM, CellML and FieldML,SBML or HumMod reached it's capabilities for representing complex models. Only the HumMod reached the integrative approach building the complex integrative model of human physiology using lumped parameter approach. I contributed to the idea of key features which involves acausal modeling technique and object orientation which keeps the complex model structure decomposed into understandable and maintainable parts and allows to cover complexity of models like HumMod.

The methods and examples of modeling cardiovascular system are described in the next section 3.3.1. The methods of estimating parameters of complex models are described in section 3.3.3 and particular results are described in section 4.3.

3.3.1 Modeling methodology

The methodology of formalizing mathematical models is influenced by the abilities of underlying modeling language used. The Modelica language is an object-oriented, equation based and acausal modeling language standardized by Modelica association[10].

The paper [5] *Modelling of Short-term Mechanism of arterial pressure in the cardiovascular system: Object-oriented and acausal approach* in Appendix E published disputation about causal and acausal approach in using Modelica for modeling lumped parameter CVS model.

The paper [6] *Simple Models of the Cardiovascular System for Educational and Research Purposes* in Appendix F published detailed methodology of modeling pul-

[9] JSIM: http://www.physiome.org/jsim/ accessed January 2015
[10] http://www.modelica.org accessed February 2015

satile CVS in Modelica.

Common guide to the Modelica language and it's capabilities are in the book of Fritzson [144] or in the on-line book by M.Tiller [145].

3.3.2 Identification of physiological systems

Usually some knowledge of the system - the structure is available and unknown coefficients (parameters) remain unknown. Once the model is formalized and constructed, further problem is to estimate the model parameters so that the model reproduces real world system. This procedure is called system identification [146, p. 159]. The objective of parameter estimation is usually to minimize the following function (to find least squares of the differences between predicted and measured values):

$$f(\vec{p}) = \sum_{i=1}^{n} (M(t_i, \vec{p}) - d(t_i))^2 \rightarrow min \tag{3.1}$$

where $\vec{p}$ is vector of values of parameters, $M(t_i, \vec{p})$ is model simulated at time t_i with the given parameter values $\vec{p}$ and $d(t_i)$ is the measured experimental value at time t_i. Algorithmically, this problem was shown to belonging to the *NP-complete* problems [147] thus the best known exact algorithm is based on brute force search - e.g. trying all possible values of parameters and simulate the model with them and find minimum of the objective function 3.1. Thus global optimization methods are based on some heuristic to reduce the number of simulation. The evolution strategies were identified as robust with potential to utilize parallel computing as shawn by Moles at al.[148]

After the parameter estimation a further problems arise with structural identifiability and analysis of sensitivity to the estimated parameter values[146, p. 176].

Parameter estimation and further analysis methods are part of specialized mathematical software. E.g. Pruet et al. used Metropolis algorithm to produce a distribution of parameters to calibrate the model of human cardiovascular physiology, which were further tested against predictive ability of circulatory failure and statistical methods performed in the software Wolfram *Mathematica* [149]. The iterative improvement method in the software MATLAB Simulink® was used in estimating 2 parameters of simple cardiovascular model by Takahashi et al. [150]. Several methods were compared in estimating multiple parameters of cardiovascular system in MATLAB Simulink® by Abbass et al. [151].

Maffioletti et al. published GC3Pie framework utilizing evolutionary algorithms and introduced workflow to identify parameters of models for economical predictions using grid computing [152]. Humphrey et al. calibrated hydrology models utilizing commercial Windows Azure cloud computing infrastructure with a significant speedup on the modified dynamically dimensioned search algorithm[153, 154].

3.3.3 Methods for Parameter Estimation

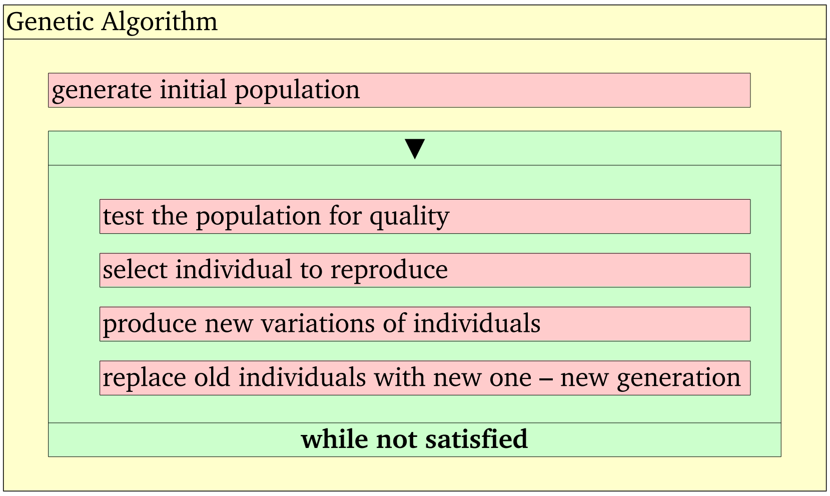


Figure 3.3: Kopenogram of genetic algorithm.

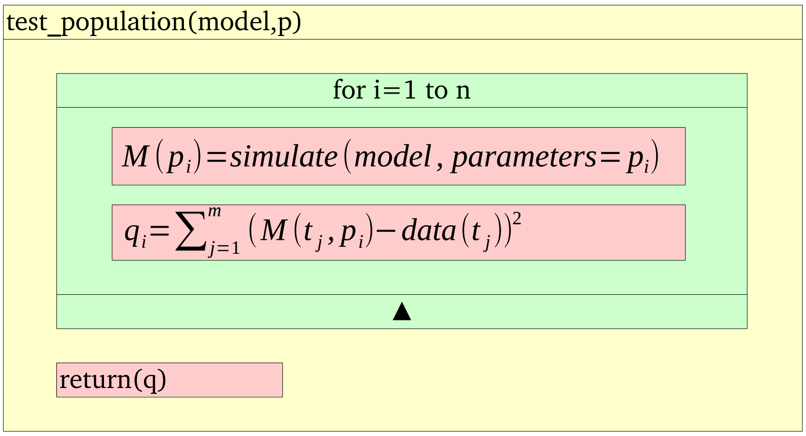


Figure 3.4: Kopenogram of genetic algorithm and specific test of the population for quality in case of parameter estimation. Model is simulated according to individual i with parameters p_i and the quality q_i is counted per the objective function 3.1.

Evolutionary algorithms can be used as heuristic strategy for finding global minimum or maximum and it can be used to estimate the parameters of the model. Genetic algorithm a type of evolutionary algorithm which encodes individuals as binary string was introduced e.g. by Holland[155] and the algorithm steps are schematically presented in figure 3.3.

The iteration within the loop "▼... *while not satisfied*" depends on previous iteration, thus it cannot be parallelized. The step *test the population for quality* has algorithmical structure in fig.3.4 for parameter estimation tasks. Each iteration in the loop "*for i=1 to n*" is independent and a therefore loop parallelism (section 2.2.1) can be utilized and implemented here.

Architecture of system for parameter estimation

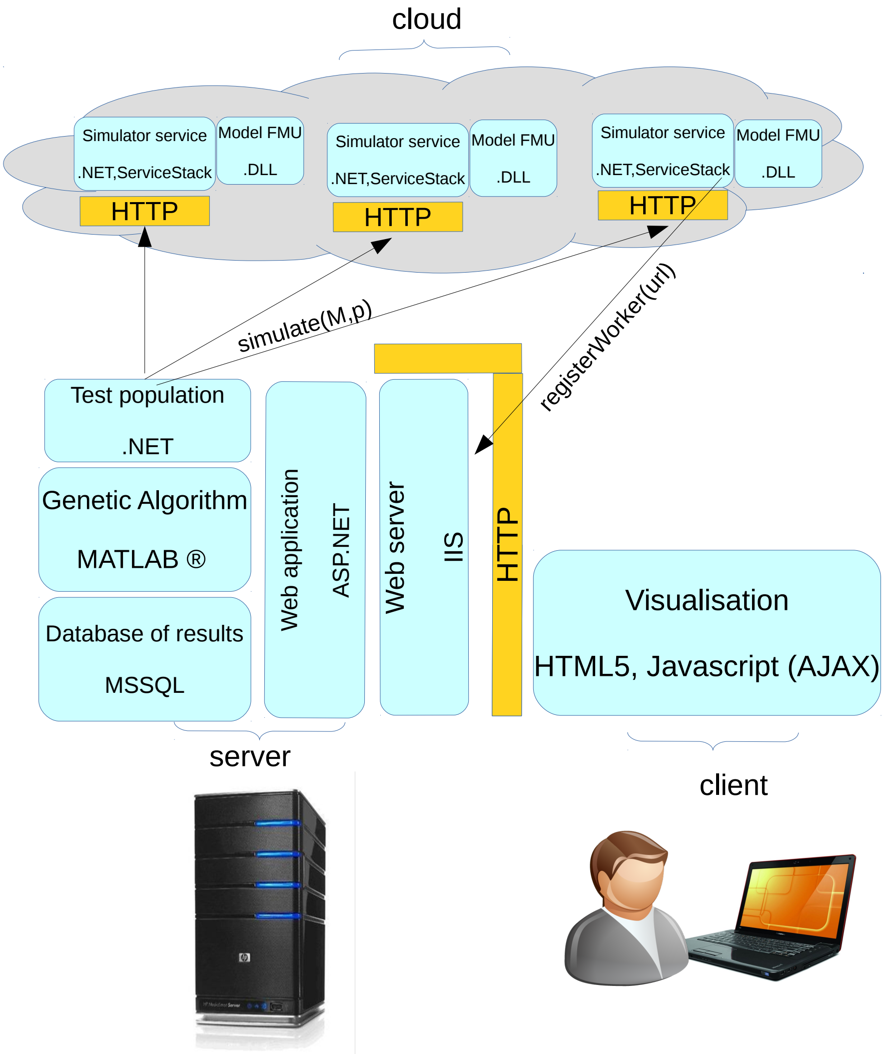


Figure 3.5: Architecture of the system employing genetic algorithm and distributing the *simulate* task into cloud computing environment.

Proposed architecture of the system for parameter estimation (fig. 3.5) was influenced by the need of some interactivity and overall accessibility for users which is fulfilled by the web UI. The key part of the system in opposite side is a model exported into a binary platform dependent library. The specific model of a studied system implemented in Modelica is exported into standard Functional Mockup Unit(FMU) with is standardized XML metadata packaged together with binary library .DLL (or .SO) following standardized API [156][11]. In the time of writing the thesis the most stable Modelica tool export was Dymola[12] export to FMU for MS Windows platform.

The parallelization is implemented using threads in *test_population* method which within a loop follows fork/join pattern – the created threads simultaneously

[11] https://www.fmi-standard.org/ accessed February 2015
[12] http://www.dynasim.se - Dymola tool, accessed March 2015

asks for simulation results with a parameter set and main process waits until all results are returned to compute full vector of quality evaluation q.

Packaged with .NET ServiceStack framework[13] it exposes a simulation functionality as a RESTful web service which can be accessed and orchestrated by the *test_population* algorithm. The implementation of genetic algorithm is reused from MATLAB ™and with a database of results in a SQL database is integrated with ASP.NET web application presenting a web user interface and functionality to a user. The result of applying the methods and deploying the designed system in local cluster and cloud computing infrastructure is described in section 4.3.

3.3.4 Parameter Sweep

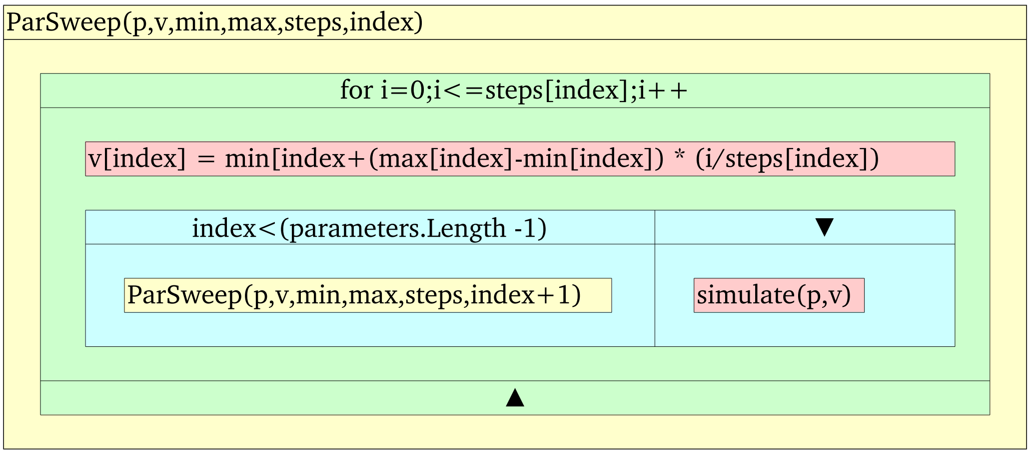


Figure 3.6: Kopenogram of recursive parameter sweep algorithm. p,v,min,max and $steps$ are arrays with the same dimension holding parameter name, value, starting and stoping value and number of steps which needs to be performed between starting and stopping value per each $index$.

Parameter sweep (PS) is one of the techniques used for sensitivity and uncertainty analysis which is based on changing selected parameters and simulating whole model and quantifying the change on model behavior with different parameters. Uncertainty and sensitivity analysis tries to determine how a change of the value of parameter will contribute to the model output and how the estimation of parameter values are robust to errors of measurement of the real data. Various methods to do uncertainty and sensitivity analysis can be found e.g. in a reviews by Helton et al. [157] or a books by Saltelli et al.[158, 159].

Recursive algorithm of parameter sweep for exploring parameter space (in fig.3.6) generates tremendous number of simulation. Presuming that *simulate* operation takes constant time for any parameters (which is not true in general) the time complexity of PS is $O(\prod_{i=1}^{n})\text{steps}_i) \approx O(k^n)$ where $k = \max_{i=1}^{n}(\text{steps}_i)$ and n is number of parameters to be swept. E.g. for 1000 values for each parameter: $O(1000^n)$.

[13] https://servicestack.net/ accessed February 2015

The large number of distinct simulation can take tremendous time on single computer. However, in contrast to parameter estimation, each of the simulation is independent and PS algorithm is determined as embarrassingly parallel and is implemented in many grid-computing projects and workflows e.g. P-Grade portal as published by Kacsuk et al.[100].

To perform parameter sweep algorithm on the models of human physiology in Modelica language an export from the Modelica is needed. The FMU standard supported by many tools exports FMU as

a BOINC platform[72][14] is customized following the task parallelism and master/worker programming model (reffered in section 2.2.1). The Modelica model exported as FMU for Windows platform is integrated with BOINC wrapper and as a whole it is integrated into BOINC platform deployed on a server as seen in fig. 3.7.

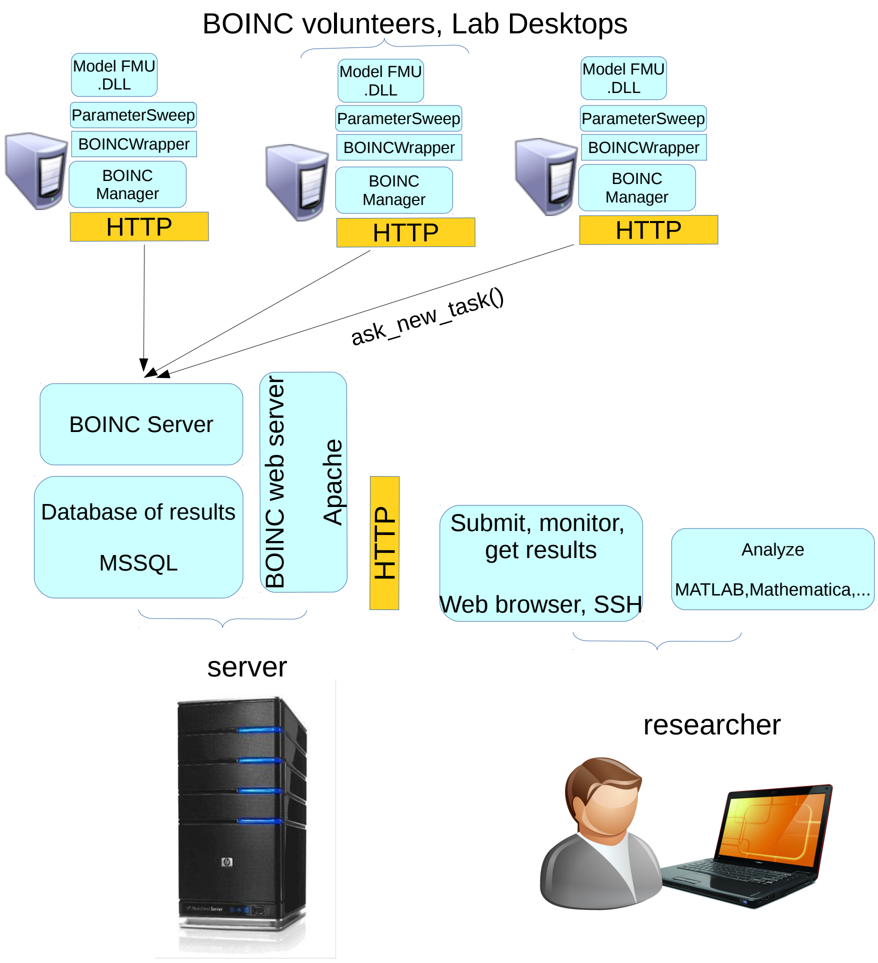


Figure 3.7: Architecture of parameter sweep application. The whole parameter space is divided into smaller spaces which are resolved by the BOINC workers

The results are described in section 4.3.

[14] http://boinc.berkeley.edu/ accessed February 2015

Chapter 4

Results

In previous chapters, there were introduced different methods available for selected use cases in research in biology and medicine. As each of the use cases and available system was proposed on different operating system platform, different architecture and or different middleware the virtualization was utilized to build the virtual infrastructures for purposes of each project. The paper [3] *Infrastructure for Data Storage and Computation in Biomedical Research* in Appendix C describes result of establishing the virtualization on physical infrastructure to share computational power among different platforms.

4.1 Medical Images

The pilot infrastructure of several servers were installed in several institutions in Prague,Czech Republic and Globus Toolkit and Globus MEDICUS was installed on them. The paper [1] *Processing of Medical Images in Virtual Distributed Environment* in Appendix A published details about the integration of Globus MEDICUS instance with MeDiMed project with conclusion that such integration via DICOM protocol is almost seamless and may bring high benefits for researcher if such grid-based system is joined with a production system for exchanging clinical DICOM data.

4.2 Remote access to voice analysis

The paper [2] *Remote Analysis of Human Voice–Lossless Sound Recording Redirection* in the Appendix B published technical details and results of customizing RDP protocol for lossless sound recording redirection and remote access via remote desktop feature of Windows platform to the application to analyze human voice and produce voice range profile for further use.

Additionally the custom RDP plugin with the ParVRP and RealVoiceLab software to redirect the sound recording without loss of information was packaged as a virtual machine template and deployed in the pilot virtual infrastructure next to the test instance of Globus MEDICUS. The virtual machine template was also deployed to different cloud computing infrastructures. One to the Amazon EC2[1] and second to the pilot scientific cloud launched in the begining of 2012 –MetaCloud[2]. Such comparison was presented to the user and technical community within CESNET and EGI organization in EGI Technical Forum 2012[23].

4.3 Parameter Estimation

The paper [4] *Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud* in Appendix D published the architecture and measurement of speedup achieved on estimating parameters of three different types of models from the non-complex, medium-complex and complex model with conclusion that only medium-complex and complex model may benefit from the architecture as the communication overhead may become major for simple models and decrease overall performance.

Additionally, scientific result was published in the paper [7] *Adair-Based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity* in Appendix G where a mathematical model of hemoglobin integrating O_2, CO_2 and H^+ binding based on theoretical principles, which were verified on the parameter estimation algorithm system[4] together with methods available in Wolfram *MATHEMATICA* 9.0[3].

Thus overall performance and speedup estimation were tested against the Modelica implementation of complex physiological model HumMod [141], the Modelica implementation of model of haemodynamics of cardivaoscular system presented by Meurs [160], the model of binding gases to hemoglobin named as Matejak2014 [7] and trivial model of a curve $f(x)$ with 4 parameters a, b, c, d defined as $f(x) = a \cdot sin(b \cdot (x - c)) + x \cdot d$ and named as "SinusCurve".

complexity	name	$T1_{[s]}$	$T2_{[s]}$	$T3_{[s]}$	$T4_{[s]}$	α	S
high	HumMod [141]	4639	4639	4618	4616	8.858×10^{-5}	11 290
medium	Meurs2011[160]	661.8	661.5	634.7	634.5	0.000 494 1	2024
low	Matejak2014[7]	17.87	17.61	1.399	1.123	0.014 44	69.26
trivial	SinusCurve	0.073	0.020	x	x	0.7260	1.377

Table 4.1: Time spent in different parts of the parameter estimation algorithm for 1 processor deployment. Genetic algorithm works with population of 120 individuals for 10 generations. T1 – is the whole time of the computation, T2 – is the time of the computation, which can be parallelized, T3 – time spent within worker node, T4 – time spent in simulation, α – computed as $1 - (T2/T1)$ and S is theoretical speedup limit per Amdahl's law $(1/\alpha)$ eq.(2.3).

[1] http://aws.amazon.com/ec2/ accessed February 2015
[2] http://www.metacentrum.cz/en/cloud/ accessed February 2015
[3] http://www.wolfram.com/mathematica/accessed February 2015

complexity	name	$T1_{[s]}$	$T2_{[s]}$	$T3_{[s]}$	$T4_{[s]}$	α	S
high	HumMod [141]	6463	6461	6451	6458	0.000 352 8	2835
medium	Meurs2011[160]	699.6	699.2	697.9	696.9	0.000 576 0	1736
low	Matejak2014[7]	2.893	2.373	1.228	1.149	0.1797	5.563

Table 4.2: Same as table 4.1, but measured on local cluster deployment with reduced communication overhead.

The computation time of single simulation depends on model complexity number of compared values. Based on the findings, the simulation of the models were divided into 4 groups depending on it's demand to compute 1200 simulations. Fraction α and speedup limit per Amdahl's law was stated in Tables 4.1 and 4.2.

Difference between $T2$ and $T3$ is an overhead introduced by the network communication between genetic algorithm and worker nodes deployed in cloud deployment provided by CESNET NGI department METACENTRUM[4]. The network overhead can be eliminated in serial implementation by directly integrating simulation into genetic algorithm, therefore, a hypothetical serial execution time estimated without the network overhead is considered and compared in Table 4.3.

	distributed in cloud				distributed in local cluster			
	overhead		est. serial		overhead		est. serial	
model name	$T2\text{-}T3_{[s]}$	$\text{fraction}_{[\%]}$	$T_{des[s]}$	S_{des}	$T2\text{-}T3_{[s]}$	$\text{fraction}_{[\%]}$	$T_{les[s]}$	S_{les}
HumMod [141]	20.98	0.4523	4619	1.005	9.858	0.1525	6453	1.002
Meurs2011 [160]	26.80	4.049	635.0	1.042	1.321	0.1888	698.3	1.002
Matejak2014[7]	16.21	90.73	1.657	10.78	1.145	39.58	1.748	1.655

Table 4.3: Comparison in cloud deployment vs. local cluster deployment of communication overhead, it's fraction in whole computation introduced by network transfer and latency. And estimated time and speedup if the worker will be replaced by serial version of computation without communication overhead: T_{xes} – estimated time of serial version of computation. S_{xes} – estimated speedup of serial version of computation against the parallel on 1 processor.

Measurement of real speedup was taken when the parameter estimation was done using 10 - 60 CPUs. In figure 4.1 are compared theoretical and measured speedup, which in general is in expected magnitude.

To sumarize the results, the simple models scale up to the 20 processors with speedup of 15, while the hypothetical serial version of the algorithm eliminating communication overhead can gain the speedup of 10. Thus this is the boundary case, and any more complex models, like the medium and complex models scales in 60 processors with speedup 39 and 50. The deployment on local cluster reduces the communication overhead, however is limited by available processors to compute concurrently, thus should be considered for the boundary cases like the simple models. The following statement could be made:

- If the alpha fraction is major, then serial computation of parameter estimation

[4]http://www.metacentrum.cz accessed March 2015

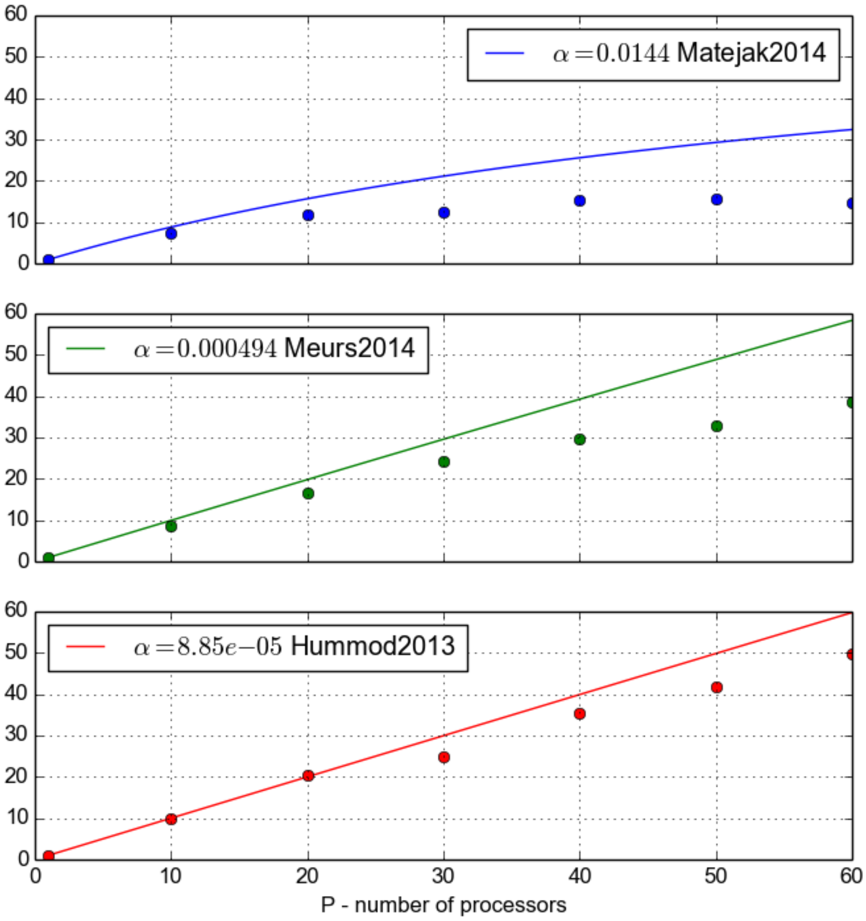


Figure 4.1: Estimated speedup (lines) per Amdahl's law (eq.2.3 [40]) for different α of several Modelica models and real measured speedup (points) on cloud deployed on 1-6 virtual machines on physical hardware (2x 6-core Intel E5-2620 2GHz, 1Gbit/s Ethernet.)

algorithm without communication overhead will perform best. This is case of the trivial function.

- If the alpha fraction is minor, but the network overhead is still major a computation on local cluster or virtual HPC cluster should be considered. This is the case of the low complex model simulation "Matejak2014"[7].

- If the alpha fraction is minor and network overhead is also minor, then distributed computation e.g. in cloud-computing environment is worth to be used. This is case of medium and high complex model simulation of "Meurs2011"[160] and "HumMod2013"[141].

4.4 Parameter Sweep

The desktop grid BOINC system was established for parameter sweep application. The established project *Physiome@home* and it's project web page http://physiome.lf1.cuni.cz/ident3/physiome manages workunits tasks which are sent to and executed by BOINC workers. The worker application is a packaged model exported as FMU for Windows platform and wrapper application which communicates with boinc manager on the desired volunteer computer.

This project is primarly used for testing purposes and for the purposes to distribute computational tasks into computers in scholar labs, which may in iddle time contribute to the computing demands. Because BOINC is very popular and users joins to a teams to compete with several types of competition, this particular project attracts after 2 weeks 78 participants from all around the world.

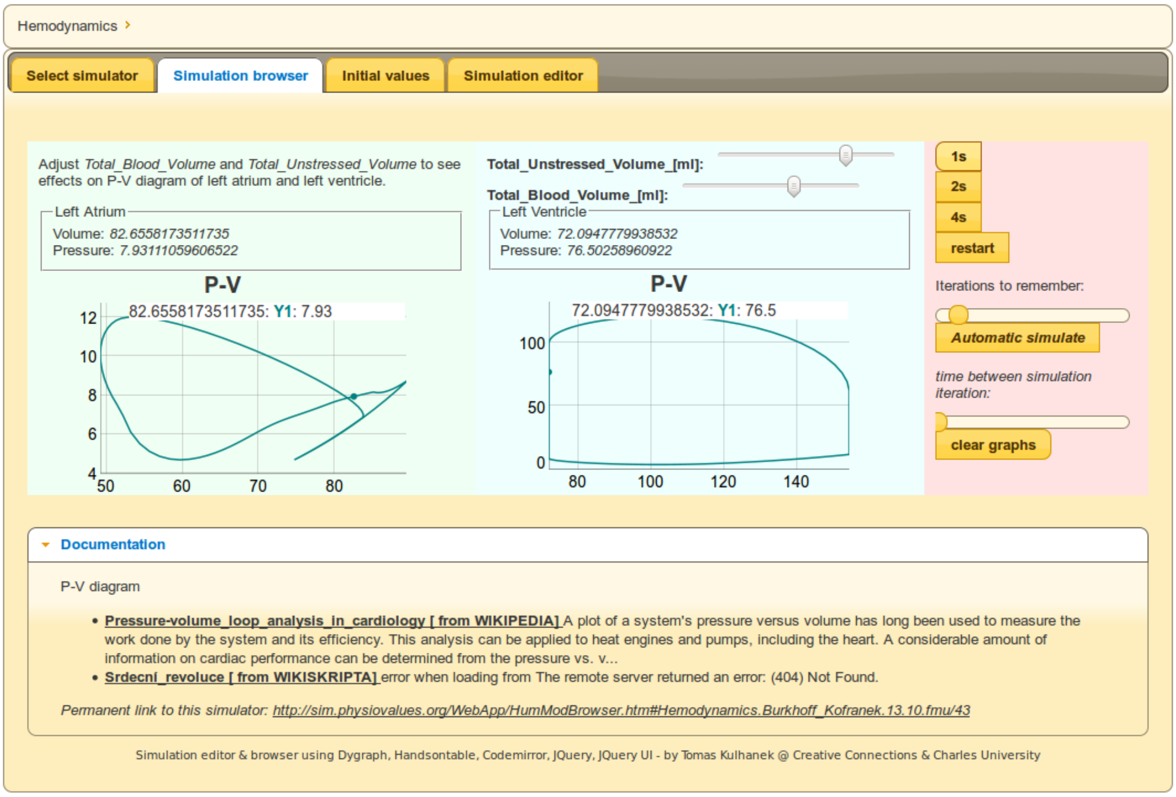


Figure 4.2: Web application to visualise simulation. In this case pressure volume diagrams of left atrium and left ventricle of the model of haemodynamics of cardivascular system.

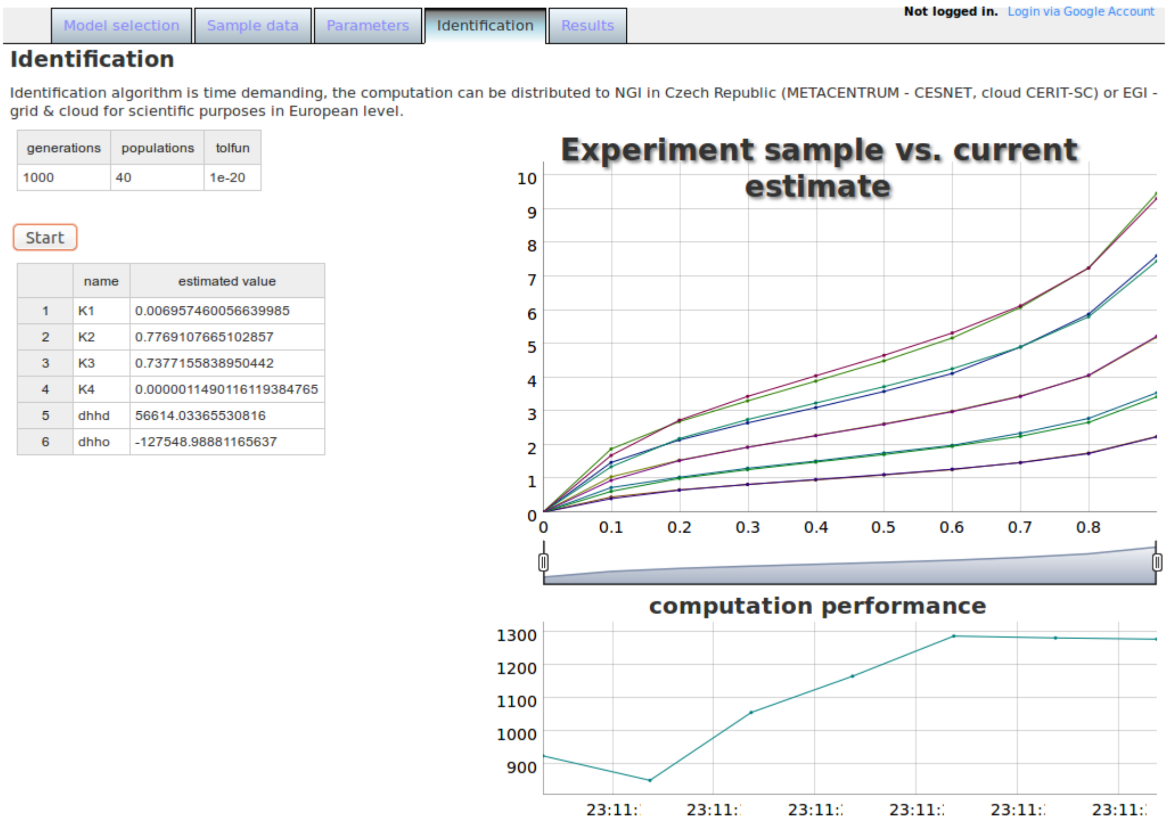


Figure 4.3: Web application for parameter estimation. in this case for the model Matejak2014[7]. Chart shows how the sample data fits with the model simulation.

Chapter 5

Conclusion

5.1 Discussion

The result presented in section 4.1 is an example how a standard format and protocol (DICOM) is utilized to integrate current production system to exchange medical images (MEDIMED) and a grid based solution (Globus MEDICUS) where an underlying technology is hidden for common user. This research was originally motivated by the idea to investigate benefits and show robust grid-based technology against proprietary distributed technology, which may face up to scalability and maintenance issues. Another issue is the philosophy of storing medical images. The presented solution based on the Globus MEDICUS is in general a data warehouse storing one or more copies of DICOM images, in contrast to federated files and metadata stored within home institutions which shares only network infrastructure to interchange the DICOM studies. This seems to be more acceptable by hospitals and by the national and international regulation for clinical and diagnostic use, e.g. The authors of Globus MEDICUS in their further development followed a way of federation of medical images stored within home institutions rather than in a grid infrastructure published by Chervenak et al. [161]. Thus the grid-computing infrastructure for sharing medical images is worth to use in the cases when additional demanding computation e.g. for processing of medical images are needed or for educational, training and knowledge sharing purposes.

The majority of user experience is kept also in the case of the application of remote voice analysis presented in section 4.2. The processing/analyzing application was kept on it's original platform but moved to remote server and a remote desktop protocol is used to redirect interaction and voice recording from user's computer to remote server and vice versa. Manipulating the . The recordings are stored on remote server for secondary use for further research and analysis algorithm improvement and in case of growth, the long-term storing issue of scientific data will be needed as in the case of medical images. Such kind of service can be deployed on any web server and a occasional need to educate or perform higher number of analysis concurently can be satisfied with cloud-computing deployment

In the case of application for parameter estimation presented in section 4.3 the user must fill the data in a form that is a spreadsheet like table and respecting some simple convention. As seen from results, the computation is sensitive on communication overhead therefore more high performance computing (HPC) hardware would be beneficial for such computation. However for medium and highly complex models the deployment of worker nodes into cloud-computing environment to for virtual HPC cluster is worth to consider. The application for parameter sweep is embarassingly parallel and suitable for high throughput computing (HTC) which is the main focus of current grid-computing infrastructures.

Platform

One of the important decision when porting an application to the grid environment is the platform of the used system.

The architecture which involves computational nodes deployed in cloud-computing infrastructure was influenced by the fact, that the model implementation is exported from third party tool to the standard FMU library as mentioned in section 3.3.3 for the MS Windows platform, which determines the platform of the worker node. The virtualization - or in case of parameter estimation a cloud computing is utilized on prepared platform with MS Windows Datacenter license. In case of parameter sweep a desktop-grid computing BOINC worker and application for MS Windows platform only is prepared for volunteers with the compatiblet system.

To utilize service-grid infrastructure an export of the model into FMU library and implementation of the wrapper service should be done in the grid-computing platform which is usually Linux based system. Another option could be to use WINE[1] – compatibility layer capable of running Windows applications on several POSIX-compliant operating systems, such as Linux, Mac OSX, BSD.

Porting

Each of the introduced systems and application was from it's beginning prepared for serial workflow. To achieve higher level of programming model, some manual intervention is usually needed on the system or source code of application.

For the smaller types of application and scientific community with their own tools it is the question, whether to invest on porting their tools to grid specific platform and parallel programming model. In the case of voice science, the analytical application was deployed on virtual machine and made available via remote desktop feature. This caused that users and researchers may stay at their platform and

[1] https://www.winehq.org/ WINE. Accessed March 2015

focus on their key technologies and research rather than to learn new one.

For the case of algorithms, that are already present in grid-computing middleware is key factor the worker/simulation part which is specific for each research community.

5.2 Future work

The "long-tail" movement described by Anderson [162] is business strategy of companies such as Amazon and Apple focusing on offering and delivering not only very popular products but also products with relatively small quantities sold each to final consumers in a price acceptable for them due to reduced sales, marketing and delivery costs. The expansion of Internet and it's related technologies caused this strategy to be profitable and successful and sales of minor products outperforms the most popular products .

There is a discussion about how to preserve the scientific data in a long-term way to prevent loss of them [163, 164] and to facilitate an access to computational resources for large amount of small scientific groups which have limited resources to port, integrate or customize their current tools and processes – the so called long-tail of science. Cloud-computing technologies seems to be customizable and may be an enabling technology to focus on long-tail science consumers as noted e.g. by Weinhardt et al.[165].

The future work in this area may contribute to design and implement policies for long-term scientific data storage and service available for them to gain access to the powerfull capacity of grid-computing and cloud-computing infrastructure.

The medical imaging and processing methods are used to identify the parameters of models of human physiology for further diagnosis statement and treatment decision. E.g. Ralovich et al.[166] proposed a noninvasive method based on computational fluid dynamic and magentic resonance imaging (MRI) to identify pressure difference in aorta for further hemodynamics analysis based on four element windkessel model. However, such type of studies usually focus on particular phenomenon and tries identify parameters of very simplified models that are currently known in systems biology domain. In the time of writing this thesis, Magnetic Particle Imager [2] (available only for animal models and not yet for human medicine) can produce high resolution images with fast shutter speed (20 ms). Several computational and storage demanding biomedical application were shown in animal models by Saritas et al.[168]. There are research infrastructures which were established to coordinate the research in biology and medicine, e.g. Integrated Structural Biology Infrastructure for Europe (INSTRUCT)[3], European Life Science Infrastructure

[2]introduced first by Gleich and Weizenecker[167]

[3]https://www.structuralbiology.eu/ accessed March 2015

for Biological Information (ELIXIR)[4], European Biomedical Imaging Infrastructure (Euro-BioImaging)[5] and others. As noted by Hunter et al.[169] the purpose of these initiatives is to understand high-level phenotypes from genomic, metabolomic, proteomic, imaging and other types of data which requires multiscale mathematical models and simulation delivered e.g. by virtual physiological human (VPH)[6] project. Future direction of research may focus not only on integrative effort of the multiscale modeling to integrate current complex models of human physiology which are based mainly on lumped parameter approach towards an integration with multidimensional models of geometrical, mechanical properties and the time-dependence of the compartments data taken from the medical and biological repositories.

5.3 Summary

This thesis presents the infrastructure which thanks to virtualization technology joined several domain specific tools in the field of sharing and processing medical images, performing real-time voice analysis and simulating human physiology.

A seamless integration of grid-based PACS system was established with current distributed system to share DICOM medical images. An access to real-time voice analysis application via remote desktop technology brings this type of service to any computer capable to connect to Internet. A system to support analysis and building complex models of human physiology in the phase of parameter estimation and parameter sweep was introduced and additional computational nodes can be flexibly joined by starting prepared virtual machines in cloud-computing deployment. Methodology of building complex models of human physiology was contributed with the idea of acausal and object-oriented modeling techniques.

[4] http://www.elixir-europe.org/ accessed March 2015
[5] http://www.eurobioimaging.eu/ accessed March 2015
[6] http://www.vph-institute.org/ accessed March 2015

Appendix A

Processing of medical images in virtual distributed environment

The paper [1] published as

Tomáš Kulhánek and Milan Šárek. Processing of medical images in virtual distributed environment. In *Proceedings of the 2009 Euro American Conference on Telematics and Information Systems: New Opportunities to increase Digital Citizenship*, page 10. ACM, 2009. doi:10.1145/1551722.1551732

Available online at http://dx.doi.org/10.1145/1551722.1551732

Processing of Medical Images in Virtual Distributed Environment.

Tomáš Kulhánek
CESNET, z.s.p.o
Zikova 4, 16000 Praha 6
Czech Republic
tomaton@centrum.cz

Milan Šárek
Academy of Sciences Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague 8
Czech Republic
sarek@euromise.cz

ABSTRACT

The processing of medical images within a PACS system depends on high capacity of communication channels and high performance of computational resources. We introduce pilot project utilizing grid technology to distribute functionality of PACS system to several machines located in distant places which allows economizing utilization of network channels. We also discuss benefits and disadvantages of virtualization techniques allowing to separate physical machine capabilities from the operating system. We compare this pilot project utilizing high speed CESNET 2 network with similar mature projects based mainly on P2P secure connection, centralized system and proprietary protocols.

Categories and Subject Descriptors

J.3 [**Medical information systems**]

General Terms

Management, Design

Keywords

Virtualization, Grid, PACS

1. INTRODUCTION

The Digital Imaging and Communications in Medicine (DICOM) standard is widely used in medical devices and applications. Picture archiving and Communication Systems (PACS) to archive DICOM are currently used in information systems within hospitals and today's effort is focused on connecting the systems among hospitals. The additional security and authorization mechanism must be kept with respect of data privacy and safety as DICOM itself doesn't provide such features [1]. DICOM series represents also usually large amount of data, which has specific requirements of capacity of communication channels.

Dostal et al. [2] introduced the client-server message brokering system with a centrally located server cluster and client application on user computer, the MeDiMed project. It was primarily used in national education network CESNET2; however other clients may connect via public Internet channels too. Client application retrieve DICOM series from the client's local or institutional PACS and send it via proprietary protocol using SSL encryption to server. Client application identifies the receiver and sets some other metadata regarding the message. The receiver must have the same client application and get the DICOM series from the server later. This solution based on the central point of the system architecture may become a bottleneck or single point of failure. There are other commercial solution using SSL encryption and authentication which are based on establishing VPN connection between peer endpoints.

Erberich et al. [3] utilized grid technology and open standards and protocols to process DICOM images securely in distributed environment to prevent some issues coming from VPN and proprietary protocols. They introduced project named Globus MEDICUS which integrates DICOM interface as a service of a grid infrastructure. Montagnat et al. [4] used similar approach in their Medical Data Manager which integrates grid middleware gLite with a DICOM interface providing strong security and encryption mechanism to preserve patient's privacy.

Different systems and technologies have different requirements on hardware and software environment. Virtualization techniques allow providing separation between software and underlying hardware. However virtualization introduces some overhead when translating isolated application instruction to lower level of a system. Current virtualization techniques allow full operating system isolation. Youseff et al. [5] showed that XEN paravirtualization doesn't impose an onerous performance penalty comparing to non-virtualized OS configuration.

We deployed the selected grid middleware and DICOM grid interface service from the Globus MEDICUS project to the virtual machines within the physical servers geographically spread throughout various institutions. We successfully exchanged the DICOM series between the DICOM grid interface and MeDiMed project without any proprietary modification of the systems used.

2. METHODS

The Globus Medicus project [3] provides a DICOM grid interface service (DGIS) able to communicate in DICOM standard, metacatalog service and storage service provider. Each service may run on separated host machine. The DGIS service is a bridge to grid infrastructure and hides the fact that the data are processed

EATIS'09, June 3–5, 2009, Prague, CZ.

throughout a grid. The metacatalog service and storage service provider are deployed on the Globus Toolkit.

We modified the DGIS to be able to communicate with the client application of MeDiMed project and accept the DICOM images exchanged in this project.

Because of specific requirements of the services of Globus MEDICUS, we chose to utilize virtualization techniques to fulfill the requirements dynamically. We installed the opensource XEN paravirtualization implementation, which adds a modification to the kernel of a guest system to be able to be executed and monitored by the host machine. Modification of the host system is, however, not required on hardware with virtualization support.

We installed the services of Globus Medicus within the virtual grid nodes on the paravirtualized guest systems Centos 5.2 Linux, kernel version 2.6 which are hosted on 64-bit Intel XEON running XEN 3.0.3.

DICOM standard uses separated direct IP connection to the user's location to send the results of the user's request. Because of that the DGIS service must have direct access to the user's application or DICOM device via IP transport level. So we decided to deploy DGIS service together with other services of Globus Medicus within the same guest system. The DGIS connects to the other local or remote services of the grid infrastructure via HTTP and gridFTP protocol. The communication between nodes and services is secured by asymmetric encryption and x.509 certificates.

3. RESULTS

We deployed nodes of the pilot grid infrastructure into the following pilot location: CESNET association, First Faculty of Medicine of Charles University and Central Military Hospital. All three locations are in Prague and are connected via high speed national educational and research network CESNET2 operated by CESNET association. We plan to use the pilot grid infrastructure also for another purpose and the XEN paravirtualization allows us to deploy and test another isolated projects next to this one.

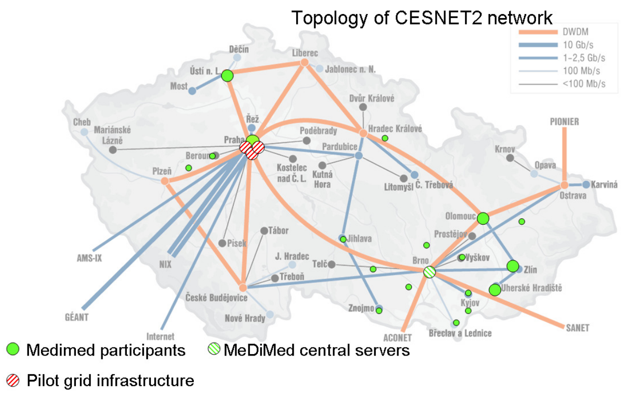


Figure 1. pilot grid infrastructure and MeDiMed project participants

We configured the guest virtual machines to share the same IP connection with the host system with IPv4 address. We configured the transport to virtual machine via network address translation (NAT) and we use Linux ipfiltering "iptables" ruleset to forward incoming connection to the grid services.

Some institutions hosting pilot project servers follow strict security policy, so they require the installation and execution of the grid services in demilitarized zone next to the institutional firewall with restricted access to local resources. With administrators of the institutional firewall we explicitly agreed and configured the firewall exception for the gridFTP protocol as the transport of such protocol uses TCP port number usually restricted by default.

We uploaded initial DICOM studies with about 1300 DICOM images for demonstration purposes. We successfully exchanged and processed the DICOM studies with the desktop application K-PACS.

We demonstrated that connection and DICOM studies exchange is possible between the MeDiMed project and the Globus Medicus. We used the client application of the MeDiMed project to retrieve and send selected DICOM series from the grid Globus MEDICUS to the participant connected in the MeDiMed project successfully and vice versa.

4. CONCLUSION

The grid technology is able to serve medical image processing in secure and reliable way as well as current systems. The only unsecured communication is between DGIS and DICOM compliant client, which is same for other types of solution (MeDiMed or VPN based) and is not usually recognized as security issue if unsecured connections are within trusted local network.

The grid services operate on specific TCP port numbers, the access to them was restricted by default in some institutions and explicit exception had to be implemented on the institutional firewall. Comparing to current production systems to share DICOM images (e.g. the MeDiMed project), they don't need such network configuration or they use VPN. The other problem regarded to network communication is sharing one IP connection among multiple virtual machines on the same host physical server. In such case we set the NAT and IP filtering rules statically on each physical server. Challenge for future development would be dynamic routing to virtual servers.

The DICOM grid service interface behaves as another DICOM compliant device and the whole system with the utilizing grid services may be considered as another PACS system e.g. as a remote backup or an external PACS for exchanging e.g. educational DICOM studies. In contrast the MeDiMed client's application doesn't allow to be controlled via DICOM protocol thus cannot be accessed by institutional application and the proprietary MeDiMed client application must be used to process DICOM studies from MeDiMed project.

The MeDiMed project will have to face problems of scalability and single point of failure. The grid technology and virtualization might be an answer to such problem for future enhancement and development as it can benefit from live network topology and doesn't need to maintain virtual topology established by VPN based solution. The MeDiMed client uses the proprietary protocol to communicate with server in contrast to the pilot grid infrastructure which is based on open standards.

Virtualization techniques allow dynamic allocation and management of physical resources. The pilot physical infrastructure of the servers might be utilized to deploy another virtual application or systems next to the DICOM and PACS services. This benefit is currently considered by the other participated institutions.

5. ACKNOWLEDGMENTS

Our thanks to CESNET z.s.p.o. and grant MSM6383917201.

6. REFERENCES

[1] Sarek, M. New Aspects of Pacs in Dwdm Network. World Congress on Medical Physics and Biomedical Engineering 2006 (2007), 417-420.

[2] Dostal, O., Javornik, M., and Ventruba, P. Collaborative Environment Supporting Research and Education in The Area of Medical Image Information. INTERNATIONAL JOURNAL OF COMPUTER ASSISTED RADIOLOGY AND SURGERY 1 (2006), 98.

[3] Erberich, S., Silverstein, J., Chervenak, A.,Schuler, R., Nelson, M., and Kesselman, C. Globus MEDICUS-Federation of DICOM Medical Imaging Devices into Healthcare Grids. STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS 126 (2007), 269.

[4] Montagnat, J., Frohner, A., Jouvenot, D., Pera, C., Kunszt, P., Koblitz, B., Santos, N., Loomis, C., Texier, R., Lingrand, D., Guio, P., Rocha, R. B. D., de Almeida, A. S., and Farkas, Z. A Secure Grid Medical Data Manager Interfaced to The Glite Middleware. J. Gri Comput. 6, 1 (2008), 45-59.

[5] Youseff, L., Wolski, R., Gorda, B., and Krintz, C. Paravirtualization for Hpc Systems. Frontiers of High Performance Computing and Networking a ISPA 2006 Workshops (2006), 474-486.

Appendix B

Remote Analysis of Human Voice – Lossless Sound Recording Redirection

The paper [2] published as

Tomáš Kulhánek, Marek Frič, and Milan Šárek. Remote Analysis of Human Voice–Lossless Sound Recording Redirection. *Analysis of Biomedical Signals and Images. Proceedings of 20th International EURASIP Conference (BIOSIGNAL)*, pages 394–397, 2010. URL: http://bs2010.biosignal.cz/papers/1092.pdf

Available online at bs2010.biosignal.cz/papers/1092.pdf.

Remote Analysis of Human Voice – Lossless Sound Recording Redirection

Tomáš Kulhánek [1,š], Marek Frič [2], Milan Šárek [1]
[1] CESNET z.s.p.o., [2] Music and Dance Faculty of the Academy of Performing Arts, [3] First Faculty of Medicine, Charles University
tomaton@centrum.cz

Abstract. This paper introduces a new approach to lossless sound recording redirection over remote desktop protocol. This approach is also demonstrated on an application in the field of Analysis of Human Voice used by phoniatric examination developed in the Musical Acoustics Research Centre of the Music and Dance Faculty of the Academy of Performing Arts. There was developed plug-ins for Remote Desktop Protocol which adds functionality to redirect sound recorded on the client side to the remote application without loss of information. Using these plug-ins the analytic application can be deployed on the remote server and accessed via RDP clients from different locations connected to the Internet.

1 Introduction

The human voice can be analyzed with computation methods during the phoniatric examination and a result of such analysis is produced into a voice range profile, also called as a phonetogram. It shows dynamic range of the human voice in terms of both fundamental frequency and intensity. The analytic application introduced in this paper ParVRP is being developed in the Musical Acoustics Research Centre of the Music and Dance Faculty of the Academy of Performing Arts. It implements computation methods in MATLAB environment and analyses the human voice stored in a sound file or recorded with a local microphone attached to the computer. The ParVRP application allows segmentation of the recording into separated voice events and produces phonetogram. [1]

There is being prepared the complex system which should allow usage of the ParVRP application from different location. This system is planned to support phoniatric examination in more physical locations is being prepared. As a pilot project, the analytic application is deployed into a virtual server which is part of the virtual infrastructure built in the past for another project exchanging medical images using data grid techniques [2]. The access to the analytic application is provided with remote desktop protocol (RDP) following the concept of thin client. Among other features, sound recording redirection from thin client to the remote application was introduced since RDP version 7.0 (server side since Windows Server 2008 R2 and client side since Windows XP SP3). The older version of RDP protocol 5.2 (since Windows Server 2003) needs external modification by third party product. Anyway, when testing the sound recording redirection in both versions, it was founded that the used codecs in both versions degraded the sound characteristics when transferred to remote application and thus this type of sound recording redirection is not acceptable for exact voice analysis [3].

2 Methods

There were discussed an option to use dedicated independent TCP/IP connection between client and server to redirect sound recording over it. This will bring some more configuration effort for the client and remote Internet providers, which may need to configure other communication channel behind the institutional firewall. Thus it was decided to reuse the existing RDP connection and customize virtual channels over RDP to transfer binary data.

The whole schema of this solution is shown on Figure 1. The system consists mainly with RDP plug-ins for client and server side. The main part of these plug-ins were developed in C# using basic .NET libraries. Together with the Mono project, these plug-ins can be deployed on the most used platforms which are Microsoft Windows and Linux-like operating systems [7]. The minor part of the plug-ins were developed with dependency on platform specific libraries for Microsoft Windows as well as for Linux operating systems.

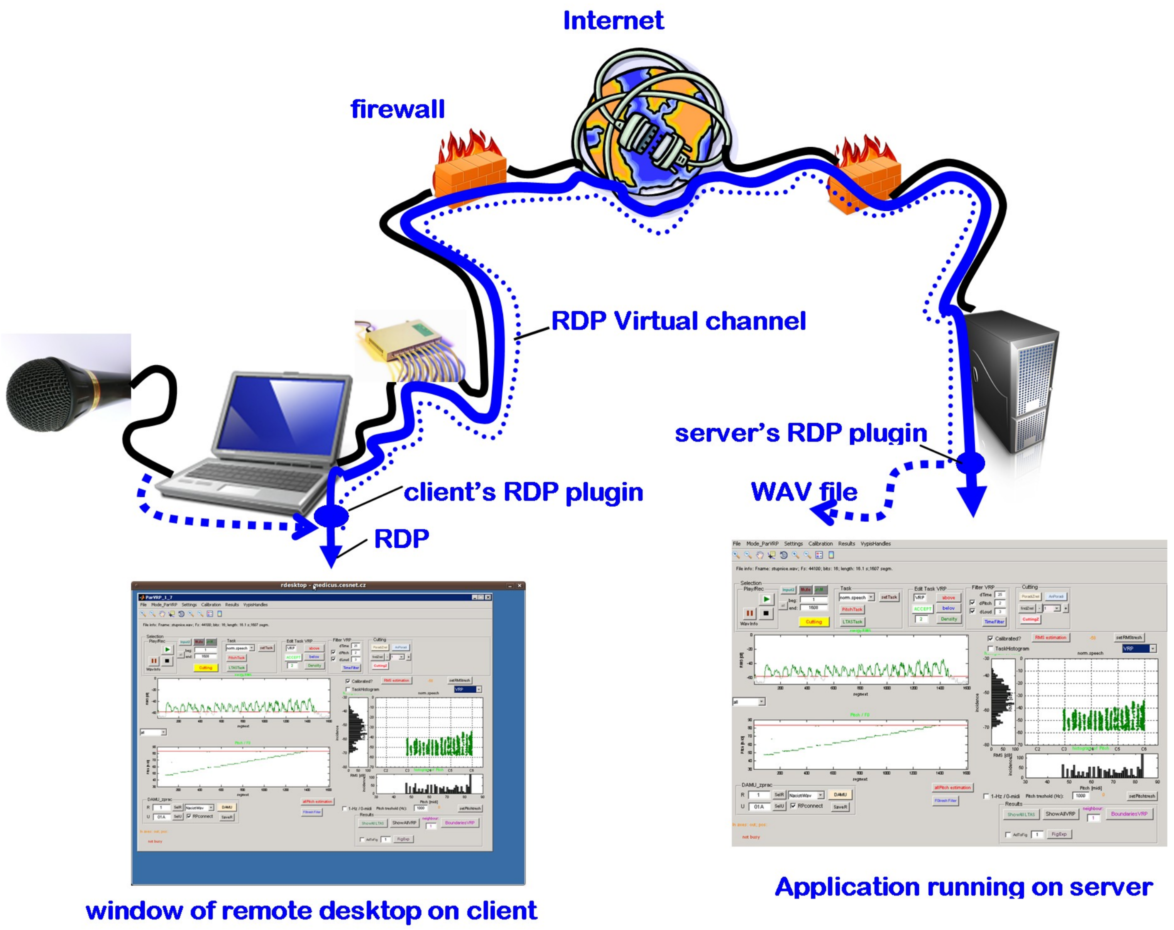


Fig 1. Schema of the sound recording redirection with the client's and server's RDP plug-in

On the remote side, the server's plug-in resides on MS Windows platform and opens a named virtual channel to the RDP connection using Remote Desktop API (Wtsapi32.dll). The plug-in then send messages via this virtual channel to control recording on the client. The received data from the virtual channel is written to a file. A proprietary API is provided for the ParVRP application which controls the recording.

On the local side, the client's plug-in to RDP receives message, starts and stops the sound recording from the system default sound-card input and sends binary data of the recorded sound via virtual channel. In case of Linux (or Unix-like) operating systems, the client's plug-in uses an external patch for the rdesktop [5] which is usually part of a common Linux distributions. It adds an option for rdesktop to redirect named RDP channel communication to standard input and output of a external executable plug-in [4]. The client's RDP plug-in listens standard input and utilize another command-line application arecord [6] also available in a common Linux distributions. Arecord records sound from system default sound card input and returns binary data in WAV format. The plug-in streams the binary data into the standard output which is then redirected by the patched rdesktop to RDP virtual channel.

In case of remote desktop client within Microsoft Windows operating system, the Remote Desktop API (Wtsapi32.dll) is used to register a client's plug-in for receiving events regarding the named virtual channel. However the plug-in in .NET is a so-called managed code which however needs to be called from an unmanaged system API, thus e.g. Selvin proposes automated solution which decompiles managed code from .NET DLL and after modifications it compiles it back into DLL which is then usable from unmanaged API calls[8]. The The WINMM multimedia API (winmm.dll) is used to record sound from the system default sound card input into data stream which is then written to the RDP virtual channel.

3 Results

Using the client's and server's RDP plug-in, there can be transferred audio data via the named RDP virtual channel without any loss of information, which is the main lack of the existing solution of sound recording redirections available for RDP version 5.2 and RDP version 7.0. The plug-ins redirect the binary data in a stream. However the binary stream is transferred in uncompressed WAV format with 44.1 kHz sampling rate and 16 bits per sample thus the bitrate of such recording is 705,6 kbit/s which needs to be transferred via the network. The overhead of base RDP protocol, which uploads events from client's keyboard and mouse to remote application is minimal. This bitrate doesn't cause any problems on the server deployed on the CESNET2 network, where the network bandwidth is above 1 Gbit/s. The bandwidth on the client's side location might be limitation if it is less than 1Mbit/s.

In contrast, the average bitrate of sound recording redirection of RDP 7.0 or Sound over RDP in RDP 5.2 is only 80 kbit/s thus this solution are not demanding on the network bandwith on client's side.

Protocol	**Upload to remote application**	**Type of sound transfer**
RDP v 5.2 + Sound Over RDP	80 kbit/s	Lossy
RDP v 7	80 kbit/s	Lossy
RDP v 5.2 + plugins for WAV via virt.channel	705,6 kbit/s	Lossless

Tab 1. Comparison of upload rate and type of sound recording redirection

The RDP plug-ins are distributed as DLL libraries which can be integrated into server's Windows platform. Proprietary .NET API is provided for ParVRP application developed in MATLAB. On the client side it can be integrated into general RDP clients on both Linux and Windows platforms.

4 Conclusions

In contrast with sound recording redirection of RDP 7.0 or Sound over RDP in RDP 5.2, the introduced redirection of sound recording over RDP doesn't provide a virtual sound card which may be used by a general remote application. This breaks a so-called “transparency” and any application like ParVRP needs modification to use the proprietary API provided by the RDP plug-ins.

Anyway the introduced sound recording redirection is acceptable for planned deployment in production environment to support phoniatric examination and provide high quality on-line analysis with a planned possibility of remote corroboration among more specialists.

The introduced approach of lossless sound recording redirecting over RDP to the remote application is also 10 times demanding for transfer upload rate than the mentioned options using lossy codecs, however still far bellow the bandwidth provided by CESNET2 network, which will generally connect the collaborating workplaces.

Comparing to the mentioned dedicated TCP/IP channel, which can be used to transfer sound recording instead of the introduced virtual channels in RDP, there would be needed to solve authentication and encryption of the communication together with additional effort needed to configure institutional firewalls to allow this channel. RDP protocol provides already authentication and encryption mechanism thus the RDP plug-ins don't need to solve that issues.

Acknowledgement

This work has been supported by the grant projects 1M0531 Musical acoustics research centre and by the research plan Optical National Research Network and its New Application, in association CESNET z.s.p.o.

References

[1] Frič M., Parametrizovaný fonetogram obecných řečových a hlasových projevů –ParVRP, MARC-Technologický list čís. 12, Zvukové studio HAMU, Praha 2007, http://web.hamu.cz/zvuk/vyzkum/dokumenty/TL12x.pdf
[2] Kulhánek, T. and Šárek, M. 2009. Processing of medical images in virtual distributed environment. In Proceedings of the 2009 EATIS'09. ACM, New York, NY, 1-3. DOI= http://doi.acm.org/10.1145/1551722.1551732
[3] Kulhánek T., Frič M., Šárek M.: Remote Access to Virtual Education and Research Application, MEFANET 2009 proceeding, Brno, ISBN: 978-80-7392-118-7, http://www.mefanet.cz/res/file/mefanet2009/prispevky/kulhanek_full.pdf
[4] Yakimenko S.,"Writing plugins for Rdesktop", The Code Project, Apriorit Inc, 1.12.2009, web, http://www.codeproject.com/KB/cross-platform/writing-plugins-forrd.aspx
[5] rdesktop: A Remote Desktop Protocol Client for accessing Windows Terminal Services, web: http://www.rdesktop.org
[6] Arecord, command line wav recorder, web: http://alsa.opensrc.org/index.php/Arecord
[7] mono, cross platform, open source .NET development framework, web: http://www.mono-project.com/Main_Page
[8] Selvin, How to Write a Terminal Services Add-in in Pure C#, The Code Project, 14.11.2006, web: http://www.codeproject.com/KB/system/TSAddinInCS.aspx

Appendix C

Infrastructure for data storage and computation in biomedical research

The paper [3] published as

Tomáš Kulhánek. Infrastructure for data storage and computation in biomedical research. *European Journal for Biomedical Informatics (EJBI)*, 6(1):55–58, 2010. URL: http://www.ejbi.org/img/ejbi/ejbi2010-1.pdf

Available online at www.ejbi.org

Infrastructure for Data Storage and Computation in Biomedical Research

T. Kulhánek[1]

[1]Institute of Pathophysiology, 1st Faculty of Medicine of Charles University, Laboratory of Biocybernetics and Computer Aided Teaching, Prague, Czech Republic

Supervisor: Ing. Milan Šárek, Csc.

Summary

Infrastructure as a service (infrastructure which is offered to customer in the form of service of the provider) is a deployment model which allows utilize data and computing capacity of a cloud as a set of virtual devices and virtualized machines. Infrastructure as a service can be offered separately to each project. The same capacity of connected physical machines and devices can be shared. Currently, the concept of an Infrastructure as a service is tested on several projects within activity of the CESNET association, First Faculty of Medicine, Charles University, Prague and Musical and Dance Faculty of Academy of Performing Arts in Prague.

The current research in the field of computation physiology is demanding on a high computation capacity. The computation tasks are distributed to computers, which are provided by the infrastructure. The project in the field of the analysis of a human voice is demanding on high throughput of a computer network between an acoustic or video device on the local side and an analytic application on the remote high performance server side. This paper describes features and main challenges for infrastructure dedicated for such a type of an application. Infrastructure as a deployment model of cloud computing might be beneficial for a multi domain team and for collaboration and integration of a high specialized software application.

Keywords: cloud computing, infrastructure as a service, virtualization, computation physiology, identification of physiological systems, validation of physiological models, remote desktop protocol, grid computing, voice range profile

1. Introduction

Several tasks can be found in the field of an application dedicated to support biomedical research of the current distributed computing systems. The main tasks cover exchanging, storing and retrieving data. The other task is to support the analysis of data and allow long lasting parallel computation. The requirement to keep privacy of patients data is the important feature of these systems and thus it must be ensured that only authorized users may access to application and data. The high level of security is a must, or an appropriate anonymization should be implemented.

Systems focused on data exchange among different organizations try to optimize data flow via the computer network, they encrypt the data which are sent via the network, ensure the required level of reliability and integrates several incompatible systems.

The first example of distributed systems are PACS (Picture Archiving and Communication Systems). DICOM (Digital Imaging and Communications in Medicine) is the most often used standard of format and protocol to exchange medical imaging information in the field of radiology. The security of transfered images is kept on other levels of the system. PACS systems are built upon the DICOM and solves the storage and maintenance of medical images. These systems are mainly deployed and closed within a hospital or within a network of hospitals maintained by the same owner. There became a requirement to join these PACS systems from different locations. Although the DICOM protocol is used, PACS systems have proprietary implementation of management and maintenance of DICOM images and there appeared an issue caused by incompatibilities of PACS systems.[1]. There were introduced systems for exchanging DICOM, which followed the classical structure of central storage and distributed user access (e.g. MeDiMed)[2]. There were introduced systems built as a communication centers with ability to send data among the different PACS systems (e.g. ePACS[3] or ReDiMed).

The project R-Bay was the another example of distributed systems for medicine. There were researched the possibilities how to join general systems, including exchange of DICOM images among institutions from different European countries. There was researched also the ability to provide and consume services of radiologists remotely on an international level. [4]

The systems which use computation and/or data grid is the other example of the distributed system in medicine. The project Globus Medicus is built upon the grid middleware Globus Toolkit. It provides services which presents a the DICOM interface to the user. The usage of grid middleware brings some beneficial features like reliability, security and effective transfer of data [5].

The system built within the project FONIATR is an example of the system demanding on the data transfer rate. This system supports a phoniatric examination and provides an application for the analysis of the human voice over the remote desktop protocol (RDP) in the MS Windows platform [6].

The system built within the project IDENTIFIKACE is an example of the computation system. Computational models of human physiology are developed in the MATLAB/Simulink environment and the new models also in the Modelica language [7]. Current models cover the whole complex functional parts of human physiology and reuse published relations and schema [8]. The models are validated against the data measured on patients within the project work, this is so-called identification of physiological systems. Some of the model parameters cannot be measured thus they are estimated by optimization techniques, which are demanding on performance and take a lot of a computational time. The estimation of parameters may take several hours or days on some more complex models. There are developed techniques to parallelize the computation and distribute the computation tasks to desktop computers in the laboratory or computers in the academic grid centers [9].

It's possible to built an own infrastructure for each of the previously mentioned systems. This may be, however, demanding on money, time and human resourcess when purchasing, installing and configuring all the needed servers and devices.

2. Methods

The infrastructure provided in an academic environment in the form of shared computational and data clusters is one of the way how to streamline the process of building up the computational and data resources. This possibility is offered by a national or international computational and data grid. Some of the projects may require a specific environment or a specific version of software library, which is not present in general grid systems. This requirement might be solved by using virtualization and cloud computing.

Computation and data grid

The computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [10]. Data grid may be characterized as an integrating architecture to do an efficient and reliable execution of data queries, which requires careful management of terabyte caches, gigabit per second data transfer over wide area networks, coscheduling of data transfer and supercomputer computation, accurate performance estimations to guide the selection of data replicas and other other advanced techniques that collectively maximize use of scarce storage, networking and computing resources [11].

It needs an additional effort to administer and maintain the grid infrastructure. This task is typically provided by a national grid initiative and the grid infrastructure is shared among different independent users. The national grid initiative was established in the Czech Republic and is maintained by the METACENTRUM activity which is one activity of the NREN (national research and educational network) provider, the CESNET association [12] and coordinates also the work with NGI from neighboring countries in the European Grid Initiative (EGI).

The system needs to be customized and use the API provided by the grid middleware Globus Toolkit, gLite and others.

Desktop grid

Anybody (from the academic and research community) can join and use the grid built by the NGI. Usually only the provider of NGI maintains and enlarges this infrastructure and grid middleware.

There exists different concepts how to built the grid. Everybody can enlarge the grid infrastructure by joining their computer and decides which application can use their computer for computation. The most well known example of such a grid system is SETI@home [13]. The concept of SETI@home follows the idea that anyone connected to Internet can join a grid by downloading a small client program and execute it in the background which periodically asks for computational jobs and computes in the background or as a screen-saver. The grid nodes are typically PCs owned by individuals. Such systems are usually referred as Volunteer grid systems or Desktop Grids and a general desktop grid system BOINC is used to build such systems, customized server site and client application to form a custom desktop grid application[14], [15].

Cloud computing

Virtualization is a technology which provides separation between a software layer and an underlying hardware layer. It allows execution of one of more so-called virtual machines sharing one physical hardware. The virtual machine is fully or partly (paravirtualization) separated from the physical layer of the hardware and thus different platforms (Windows, Linux) may work together on one physical machine concurrently. Virtualization techniques introduce some overhead when translating an isolated application instruction to the lower level of a system. However, the open source paravirtualization system XEN does not impose an onerous performance penalty comparing to non-virtualized operating system configuration [17].

The virtualization is sometimes chara-terized as a key technology which enables cloud computing and execution of different isolated systems on shared hardware.

Virtual infrastructure

Virtual organization is a group of users, who share the same resources. The virtual infrastructure belonging to a virtual organization is built from virtual machines connected via the network, which may be virtual too and accessible only to users from the virtual organization. Figure 1 shows an example of several virtual organizations and their infrastructures. On the right part there is a schematic view on physical connections among different organizations (hospitals, research institutions) via the academic network or the Internet. The physical resources are shown as vertexes and network connections are shown as edges. Each cloud shows one virtual infrastructure. On the left part there is a physical server executing more virtual machines, each machine belongs to a different virtual infrastructure.

3. Results

The pilot infrastructure dedicated for the medical application was established within the CESNET's activity “Application support” in several locations in Prague. It utilizes the open-source virtualization system XEN and tools of the operating system Linux for configuring virtual machines and virtual infrastructure as a service.

The instance of the grid system Globus MEDICUS was the primary system deployed to a set of virtual machines. It was shown that the grid system based on open standards can be easily integrated with current medical systems using the DICOM format [18].

The application to support a phoniatric examination was deployed next to the previous system to exchange medical images. It was needed to develop an enhancement of the RDP protocol, which adds the transfer of an audio signal from local computer's microphone to the remote application [6]. This system is currently deployed on one virtual machine. See Figure 2.

The pilot infrastructure contains also the system for identification of physiological systems, which offers a web service distributing the computational task to desktop computers connected via the desktop grid system BOINC and SZTAKI Desktop Grid API [9]. The schema in Figure 3 shows the architecture of the system. The server is in operation as an independent virtual machine and contains the web service. Some of the BOINC workers are in operation as independent virtual machines deployed on less used physical servers of the pilot infrastructure. Some of the desktop computers of laboratory and classroom of the First Faculty of Medicine are connected to this desktop grid system. Other computers may be easily joined. Current research is focused on the possibility to enhance the computational capacity of the infrastructure by the resources provided by the NGI MetaCentrum. There is also researched an utilization of GPU computing.

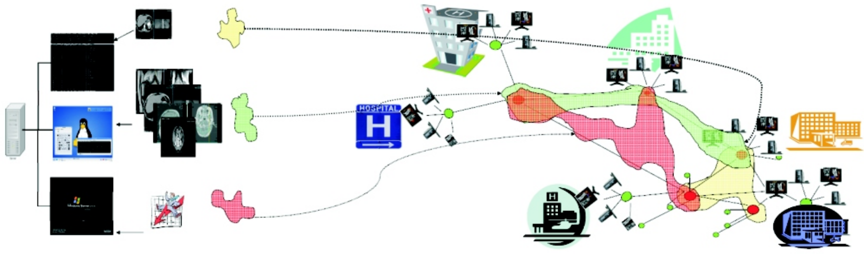


Fig. 1. General schema of virtual infrastructure.

All the physical servers of the pilot infrastructure have a virtual environment built upon the XEN system and they share the IP network addresses. Each virtual machine has its own disk partition within the Logical Volume Management (LVM) on the physical server. The virtual machines are administered by the tool virt-manager and the network environment is configured with the tool iptables.

4. Discussion

The pilot infrastructure can be characterized as a private cloud, which is accessible only to the limited community of users from the field of biomedical research. Virtual machines share the physical network connection via IPv4. Because of the lack of numbers of unique IPv4 addresses, the configuration of network services (webserver, RDP) is realized using network address translation and port mapping. If the network devices passed to the version 6 of IP protocol, there would be opened again the possibility to provide unique IP addresses to virtual machines and there would not be needed extra configurations of network address translation and port mapping.

There are not used special tools to administer cloud within the pilot infrastructure, because the number of projects is relatively small currently. Anyway, there exist free or commercial products (Eucalyptus, OpenNebula, VMWare vSphere), which provide a set of tools to automatize the maintenance of private cloud, including virtual network configuration, live migration of virtual

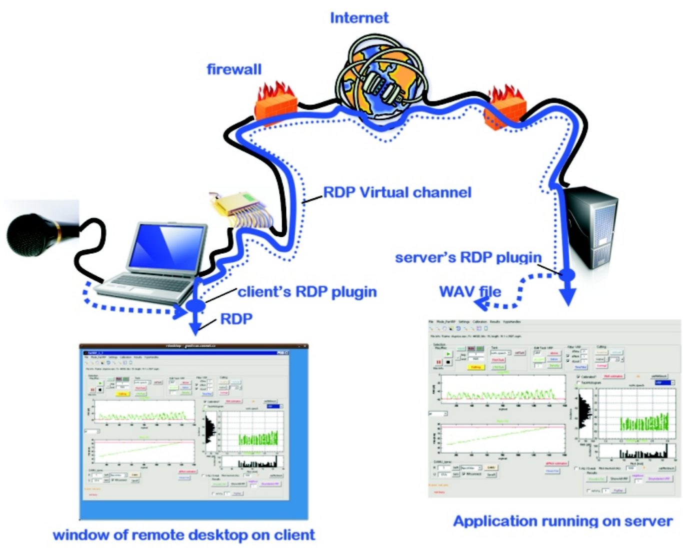


Fig. 2. Schema of system for human voice analysis and remote recording via RDP protocol.

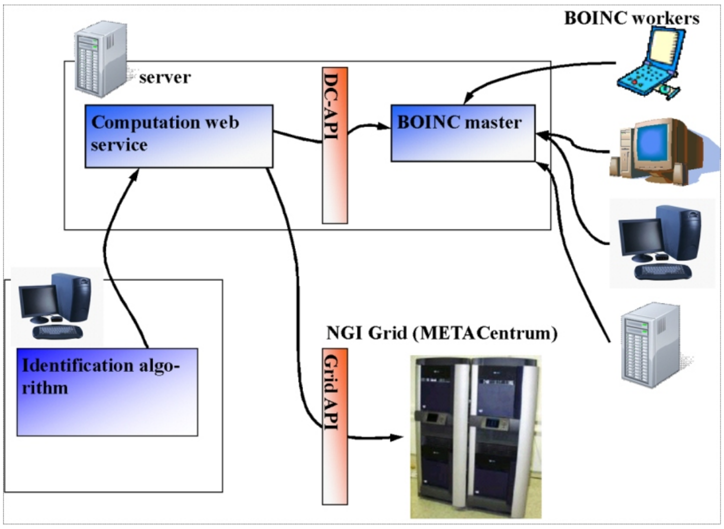


Fig. 3. Schema of computational infrastructure for identification of physiological systems.

machine, etc. Deploying these tools will be necessary in future after expected enhancement of the physical capacity, which is planned to be built within the academic environment of the Czech Republic.

The important question is: which type of the application is suitable for clouds operating on physical resources spread in different geographical locations compared to clouds operating in supercomputing centers. Clouds in supercomputing centers are suitable for highly parallel tasks which need fast communication between parallel computational tasks. Cloud operating on physical servers in different geographical locations can offer a free capacity in the time period, when the owner does not utilize its physical resources and offers them to other users of cloud.

5. Conclusion

It is possible to operate a private cloud on the physical infrastructure and to provide the virtual infrastructure to the users, who can utilize it to execute their own applications and systems. The infrastructure as a service can open an access to distributed systems to a higher amount of users, who have been so far prevented from using them by a complicated administration, too long process of purchasing and installing computing resources.

The cloud operating on physical servers in different geographical locations can be a suitable complement to the clouds in supercomputing centers.

Acknowledgment

The work was supported by the grant SVV-2010-265 513.

References

[1] KulhánekT., Šárek M.: Virtualizace a integrace v gridovém PACS systému, MEFANET 2008, Brno, ISBN 978-80-7392-065-4
[2] Dostal, O., Javornik, M., and Ventruba, P.. "Collaborative environment supporting research and education in the area of medical image information". Internation Journal of Computer Assisted Radiology and Surgery. Sringer Science+ Business Media. 1. 98. 2006.
[3] E-PACS, DICOM komunikace mezi zdravotnickými zařízeními, 2010, czech language, web: http://www.epacs.cz/
[4] Pedersen C. D., Rasmussen J., Beck C.: R-Bay; Creating an eMarketplace for the transfer of imaging related eHealth services in Europe. Final project report, version 2, web: http://www.r-bay.org
[5] Erberich S.G., Silverstein J.C., Chervenak, A., Schuler, R., Nelson, M.D., and Kesselman, C.: "Globus MEDICUS-Federation of DICOM Medical Imaging Devices into Healthcare Grids". Studies in Health Technology and Informatics. IOS Press; 1999. 126. 269. 2007.
[6] Kulhánek T., Frič M., Šárek M.: Remote Analysis of Human Voice Lossless Sound Recording Redirection,Proceedings of Biosignal 2010: Analysis of Biomedical Signals and Images, ISSN 1211-412X, ISBN 978-80-214-4106-4
[7] Stodulka, P., Privitzer, P., Kofránek, J., Tribula, M., and Vacek, O.. Development of Web Accesible Medical Educational Simulators". Proceedings of EUROSIM 2007, Ljubljana. 2. 16.
[8] Abram S.R., Hodnett B. L., Summers R.L., Coleman T.G., Hester R.L.: Quantitative circulatory physiology: an integrative mathematical model of human physiology for medical education. Adv. Physiol. Educ., 31, 202-210, (2007).
[9] Kulhánek T., Šilar J., Mateják M., Privitzer P., Kofránek J., Tribula M.: Distributed computation and parameter estimation in identification of physiological systems, VPH conference 2010, Brussels
[10] Foster I., Kesselman C. "Computational grids", Vector and Parallel Processing, VECPAR 2000, 2001 Springer
[11] Chervenak A., Foster I., Kesselman C., Salisbury C., Tuecke S.: The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientic Datasets. Journal of Network and Computer Applications, Elsevier, 2000
[12] Šustr Z., Sitera J, Mulač M, Ruda M., Antoš D., Hejtmánek L., Holub P., Salvet Z., Matyska L.: MetaCentrum, the Czech Virtualized NGI, 2009, web: http://egee.cesnet.cz
[13] Anderson D. P., Cobb J., Korpela E., Lebofsky M., and Werthimer, D. 2002. SETI@home: an experiment in public-resource computing. Commun. ACM 45, 11 (Nov. 2002), 56-61.
[14] Anderson D. P. 2004. BOINC: A System for Public-Resource Computing and Storage. In Proceedings of the 5th IEEE/ACM international Workshop on Grid Computing (November 08 - 08, 2004). International Conference on Grid Computing. IEEE Computer Society, Washington, DC, 4-10.
[15] Balaton Z., Gombás G., Kacsuk P., Kornafeld A., Kovács J., Marosi C.A., Vida G., Podhorszki N., Kiss T.: SZTAKI Desktop Grid: a Modular and Scalable Way of Building Large Computing Grids. Proceedings of the 21st International Parallel and Distributed Processing Symposium (IPDPS 2007), 26-30 March 2007, Long Beach, California, USA

[16] Velte T., Velte A., and Elsenpeter R.: Cloud Computing: A Practical Approach”. McGraw-Hill Osborne Media. 2009.
[17] Youseff L., Wolski R., Gorda B., Krintz, C. Paravirtualization for HPC Systems. Frontiers of High Performance Computing and Networking â€“ ISPA 2006 Workshops. 474486. 2006.
[18] Kulhánek T., Šárek M.: Processing of medical images in virtual distributed environment. ACM. 13. 2009.

Contact
Mgr. Tomáš Kulhánek
Institute of Pathophysiology,
First Faculty of Medicine of Charles University
Laboratory of Biocybernetics and Computer Aided Teaching
U Nemocnice 5
128 53 Prague 2
Czech Republic
e-mail: tomaton@centrum.cz

Appendix D

Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud

The paper [4] published as

Tomáš Kulhánek, Marek Mateják, Jan Šilar, and Jiří Kofránek. Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud. In *Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on*, pages 712–715, 2014. doi: 10.1109/BHI.2014.6864463

Available online at ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6864463

Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud

Tomáš Kulhánek*,[1], Marek Mateják[1], Jan Šilar[1] and Jiří Kofránek[1]

Abstract— **A generic system for estimation of model parameters —calibrate models— is introduced. The proposed system architecture is built of several loosely coupled modules behaving as RESTful web services and allowing to integrate other parts of the system via HTTP protocol and data exchanged in JSON format. The system was designed in such a way that the most demanding computational part is computed in parallel and computation may be distributed to remote computational resources. A test deployment was done in scientific cloud provided by czech NGI CESNET. Parameter identification of complex models got significant speedup on cloud computing resources.**

I. INTRODUCTION

There are several methodologies and technologies how to model a complex reality in biological domain. The large-scale mathematical description of physiological systems was introduced by Guyton et al. in 1972[1] and continues today by Hester et al. who introduced HumMod - a derivative of the Guyton's model and in-house modeling language and simulation tool[2][3].

Kofranek et al. implemented Guyton's 1972 model in MATLAB®Simulink[4]. However, the complexity of the model increased from Guyton's model to HumMod and it becomes too complicated keeping the model up-to-date using block oriented tools like MATLAB®Simulink. Therefore an implementation of the HumMod model was introduced in object oriented Modelica language[5][6]. Modelica is an acausal object oriented language introduced by Fritzson et al.[7] and it is curently mantained by the international Modelica association and implemented by several vendors[8].

The parameter identification is a task to estimate the unknown model parameters in such a way that the model simulation fits the experimental data[9]. The objective of this task could be for example to minimize the following function (least squares):

$$f(\vec{p}) = \sum_{i=1}^{n}(M(t_i, \vec{p}) - d(t_i))^2 \rightarrow min \quad (1)$$

where $\vec{p}$ is vector of values of parameters, $M(t_i, \vec{p})$ is model simulated at time t_i with the given parameter values $\vec{p}$ and $d(t_i)$ is the measured experimental value at time t_i.

The models of human physiology are in general set of linear and non-linear algebraic and differential equations, some of them may change it's behavior based on discrete conditions, thus output of such model can be non-differentiable and non-continuous. Thus global optimization methods that work without derivatives has to be used in general to find minimum of the objective function. There were tested several global optimization methods to identify parameters of models in Modelica language[10][11]. The simulation must be performed many times using this methods and it may take extremely long for complex models or larger space of parameters. The computation time can be however reduced using distributed computing techniques. Maffioletti et al. introduced GC3Pie framework and shown workflow to identify parameters using grid computing[12]. Humphrey et al. calibrated hydrology models utilizing cloud computing [13]. We proposed a system which should support the process of parameter identification mentioned in above scientific publication in more generic way, so the researcher may focus on experimental data, selecting the parameters from a model to identify and interpreting the estimates during computation and hide the technical details of configuring the computational modules in distributed systems.

The proposed system integrates visualization, identification algorithm and simulation into loosely coupled modules opened to any modeling and numerical technology. We implemented the system and tested it with the models of human physiology in Modelica language, we selected genetic algorithm as a global optimization method for parameter identification and we distributed simulation into scientific cloud.

II. SYSTEM DESCRIPTION

The proposed system as seen on Figure 1 consists of several loosely coupled distinct pieces of software modules which can be replaced by another technology or implementation with no or minimal intervention into other related modules. The communication among modules is done via HTTP protocol and endpoints follows REST architectural style[14].

The simulation module consist of Modelica model exported as a Functional Mockup Unit (FMU) conforming the standardized Functional Mockup Interface[16]. The FMU is in fact DLL library for MS Windows platform. We wrapped this FMU by the ServiceStack [17] framework to provide web interface and control the simulation via HTTP protocol and JSON format. The simulation module can be deployed in multiple instances and each one can be executed in parallel.

*corresponding author tomas.kulhanek@lf1.cuni.cz

[1]Institute of Pathological Physiology, First Faculty of Medicine, Charles University in Prague, Czech Republic

978-1-4799-2131-7/14/$31.00 ©2014 IEEE

TABLE I

TIME AND SPEEDUP OF ESTIMATING THE PARAMETERS OF MODELS.

CPUs involved	1	2	3	4	10	20	30	40	100
model HumMod[6]									
time	71d21h	38d1h	25d11h	20d8h	7d16h	3d21h	2d10h	1d20h	18h
speedup	–	1.8x	2.8x	3.5x	9.3x	18.5x	29.7x	39.2x	95.8x
model Rossi-Bernardi[15]									
time	50 min	27 min	20 min	19 min					
speedup	–	1.8x	2.5x	2.6x					

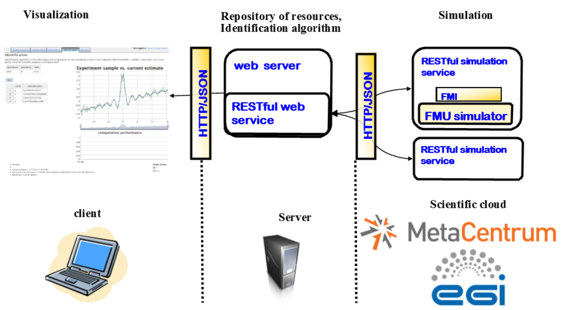


Fig. 1. Architecture of the system for parameter estimation

The simulation module instance registers it's unique endpoint URL to the repository module.

The repository module is a RESTful web service which provides storage of several data entities and provides access to the data via HTTP protocol and JSON format. The parameters are presented in web application as a table of names and values, the experiment and simulation data are visualized in a graph.

The identification algorithm is executed within this module and is used for model parameter estimation. We selected genetic algorithm from MATLAB-Optimization toolbox exported into a DLL library able to be executed with Matlab Common Runtime (MCR) environment. Each step of the algorithm produces a vector of parameter values which needs to be simulated and can be computed parallely by the instances of simulation modules.

The visualization module is implemented in HTML (version 5) and Javascript. This module gives user a list of available models. The table for experimental data allows to insert manualy or to copy&paste data from a desktop application. The table for parameters defines names, initial value and estimated maximal and minimal value, which will be taken into account by identification algorithm. After starting the identification process, the current best estimation is visualized periodically in a graph together with experimental data. From this perspective the identification algorithm behaves as a curve fitting process.

The whole system can be deployed in single computer, however it was designed to be deployed in several different computing elements. We deployed simulation modules in a local cluster and in a virtual infrastructure within scientific cloud provided by the Czech grid infrastructure provider CESNET[1], member of the European Grid Infrastructure foundation (EGI[2]). The repository module and identification algorithm controls the connected simulation modules.

III. RESULTS

For testing purposes we selected one known parameter from the HumMod model[5] and identify it again. We also tested to identify 4 parameters of model of hemoglobin saturation curve in variable condition of acidity and concentration of carbon dioxid based on the model of Rossi-Bernardi1967[15] implemented in Modelica language. We set the genetic algorithm to finish after 200 thousands single simulations for both models giving the best result found during the computation. The simulation was distributed into local cluster (up to 4 parallel simulation processes, CPU Intel XEON 2.7GHz)) and in scientific cloud (up to 100 parallel simulation processes, CPU Intel E5-2620 2GHz).

Experimental data for HumMod model were generated from single simulation of the model with specified parameter. In the case of the Rossi-Bernardi1967 model, we took the experimental data from the publication[15]. The values of parameters identified during the computation were comparable with the known values. However, we focus on the time of computation and possible speedup when the simulation was distributed into more parallel processes. The measured computation time and speedup is in the table I.

The single simulation of HumMod model takes about 30s of computation time. And we estimate the whole process of computation to 71 days. We didn't wait more than 2 months for this results, rather we estimate this after couple of hours from the number of the simulation done. When distributed into the local cluster up to the 4 CPU, we got the speedup about 3.5 times. When the computation was distributed into virtual infrastructure of 10 computers each contributing by 10 CPU with computation (totally 100CPU) we got the speedup about 96x and the estimation of 1 parameter was done in 18 hours.

The single simulation of the Rossi-Bernardini1967 model takes about 15miliseconds. When computation was distributed into 3 parallel nodes (3 CPU) the utilization of the service module was high and adding another computation node (totally 4 CPU) we did not get any other significant speedup. Distributing the computation to the scientific cloud we got even worse results influenced mainly by the network latency and increased communication overhead.

[1]http://www.cesnet.cz
[2]http://www.egi.eu

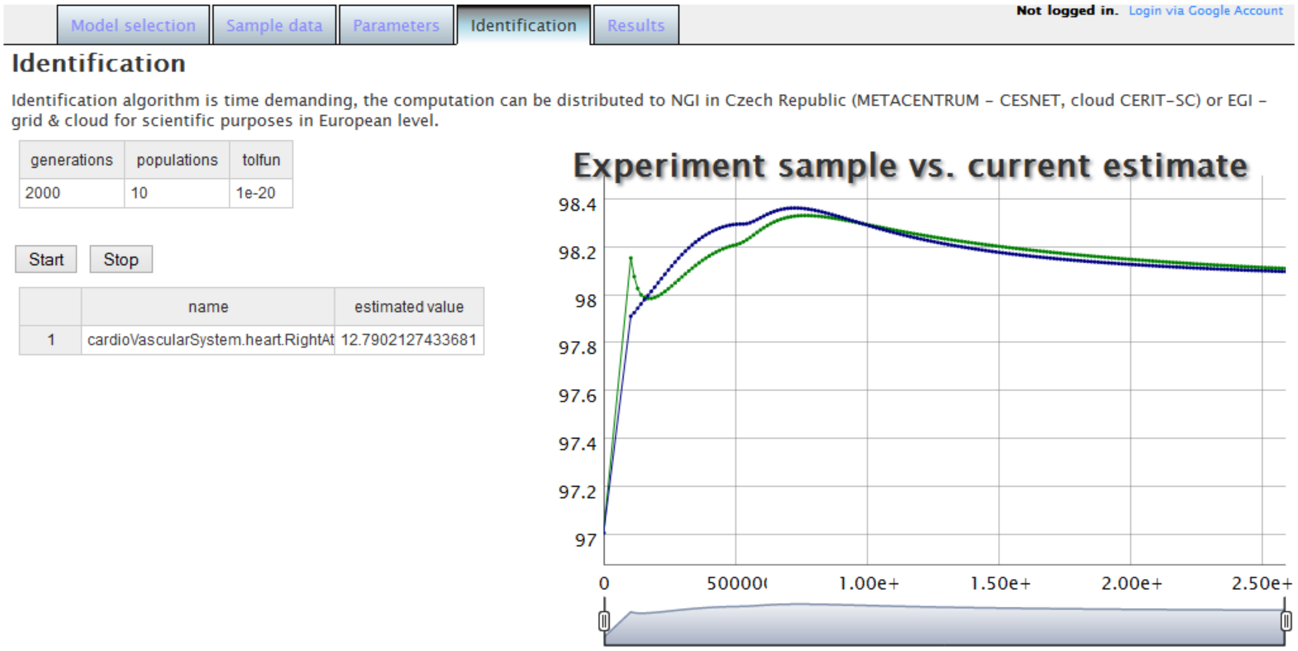


Fig. 2. Screenshot showing running estimation. Current best estimation of parameters values on the left table and curve of estimated model data (blue) vs. experimental data (green).

IV. DISCUSSION

The parameter identification of the complex model spends majority of it's time in parallel simulation and minority in communication and synchronization procedures. This type of tasks can achieve significant speedup if the computation is distributed into remote capacity e.g. within a cloud or grid. On the other hand, the parameter identification of simpler models converges to a highly parallel computation system where time spent in parallel simulation is moreless same as in communication and synchronization procedures.

We estimate that if the single simulation takes more than a second, then the identification task is worth to deploy into cloud computing environment and gain a speedup from it. However if the single simulation takes less for simpler models, than the identification task should stay on local cluster or should be computed in some supercomputer. Distributing them into cloud or grid using this system we do not get any significant speedup. However more exact distinction should be done in further studies.

The parameter estimation within this paper was provided by genetic algorithm, however, there are other identification algorithm (e.g. other evolutionary algorithms) which can gain significant speedup utilizing distributed computing environment.

The system was tested with models implemented in Modelica language, however, significant contribution to the knowledge of human physiology were done by other projects, e.g. VPH[18] or IUPS Physiome[19], which has a so called Physiome model repository[20] and the majority of models are in CellML modeling language or JSIM modeling language. There is an effort to develop translation tool among the modeling technologies to give researchers freedom of choosing the modeling technology e.g.[21]. The further development of the system for parameter estimation can be enhanced to support and simulate the models in the above mentioned modeling technologies.

The web application provides minimal set of functionality for system analysis. Further development of the web application and introduced system needs to do an usability survey and incorporate most useful functionality. For other methods and tasks related to system analysis in physiology use other specialized tools e.g. Design.Calibrate library available in Dymola[22], Optimization toolbox from MATLAB®etc.

V. CONCLUSION

We presented a system to support identification of physiological system in the phase of parameter identification. The loosely coupled part of the system might be deployed into remote distributed computational capacity and significant speedup was shown in the case of the large complex physiological model computed in cloud computing infrastructure.

The described system is accessible via a web application and allows user to focus on input experimental data, names of parameters, visual control of calibration process and hide unnecessary complexity of configuration of the remote computation.

The continued work is oriented to enhance the complex model of human physiology - HumMod and to integrate other modeling technologies.

ACKNOWLEDGMENT

This work was supported by project MPO FR-TI3/869 and by project FR CESNET number 431. The computation was

done on virtual resources of scientific cloud provided by the czech national grid infrastructure, the association CESNET.

References

[1] A. C. Guyton, T. G. Coleman, and H. J. Granger, "Circulation: overall regulation." *Annual review of physiology*, vol. 34, pp. 13–46, Jan. 1972. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/4334846

[2] R. L. Hester, A. J. Brown, L. Husband, R. Iliescu, D. Pruett, R. Summers, and T. G. Coleman, "Hummod: a modeling environment for the simulation of integrative human physiology," *Frontiers in physiology*, vol. 2, 2011.

[3] R. L. Hester, R. Iliescu, R. Summers, and T. G. Coleman, "Systems biology and integrative physiological modelling," *The Journal of physiology*, vol. 589, no. 5, pp. 1053–1060, 2011.

[4] J. Kofránek and J. Rusz, "Restoration of Guyton's diagram for regulation of the circulation as a basis for quantitative physiological model development." *Physiological research / Academia Scientiarum Bohemoslovaca*, vol. 59, no. 6, pp. 897–908, Jan. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20533860

[5] J. Kofranek, M. Matejak, and P. Privitzer, "Hummod-large scale physiological models in modelica," in *Proceedings 8th Modelica Conference, Dresden, Germany*, 2011, pp. 713–724.

[6] J. Kofránek, M. Mateják, P. Privitzer, M. Tribula, T. Kulhánek, J. Šilar, and R. Pecinovský, "Hummod-golem edition: large scale model of integrative physiology for virtual patient simulators," in *Proceedings of World Congress in Computer Science 2013 (WORLDCOMP'13), International Conference on Modeling, Simulation and Visualisation Methods (MSV'13)*, 2013, pp. 182–188.

[7] P. Fritzson and P. Bunus, "Modelica-a general object-oriented language for continuous and discrete-event system modeling and simulation," in *Simulation Symposium, 2002. Proceedings. 35th Annual*. IEEE, 2002, pp. 365–380.

[8] "Modelica and the Modelica Association," https://modelica.org, accessed: 2013-12-01.

[9] M. C. Khoo, *Physiological control systems*. IEEE press, 2000, ch. 7: Identification of Physiological Control Systems.

[10] S. Proßand B. Bachmann, "An Advanced Environment for Hybrid Modeling and Parameter Identification of Biological Systems," *Proceedings 8th Modelica Conference, Dresden, Germany*, pp. 557–571, Jun. 2011.

[11] T. Kulhanek, J. Silar, M. Matejak, P. Privitzer, J. Kofranek, and M. Tribula, "Distributed computation and parameter estimation in identification of physiological systems," *Proceedings of VPH conference*, 2010.

[12] S. Maffioletti, R. Murri, B. Jonen, and S. Scheuring, "Computational workflows with gc3pie," in *Proceedings of the EGI Community Forum*, 2012.

[13] M. Humphrey and N. Beekwilder, "Calibration of watershed models using cloud computing," *Proceedings of IEEE 8th International Conference on E-Science*, 2012.

[14] R. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, 2000.

[15] L. Rossi-Bernardi and F. Roughton, "The specific influence of carbon dioxide and carbamate compounds on the buffer power and Bohr effects in human haemoglobin solutions," *The Journal of Physiology*, pp. 1–29, 1967.

[16] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J. Peetz, S. Wolf, G. I. T. I. Gmbh, and D. L. R. Oberpfaffenhofen, "The Functional Mockup Interface for Tool independent Exchange of Simulation Models The Functional Mock-Up Interface," *Modelica*, pp. 105–114, 2011.

[17] "Servicestack," https://github.com/ServiceStack/ServiceStack, accessed: 2014-01-06.

[18] P. Kohl and D. Noble, "Systems biology and the virtual physiological human," *Molecular Systems Biology*, vol. 5, no. 1, 2009.

[19] P. Hunter, P. Robbins, and D. Noble, "The IUPS human physiome project," *European journal of physiology*, vol. 445, no. 1, pp. 1–9, 2002.

[20] T. Yu, C. M. Lloyd, D. P. Nickerson, M. T. Cooling, A. K. Miller, A. Garny, J. R. Terkildsen, J. Lawson, R. D. Britten, P. J. Hunter, and P. M. F. Nielsen, "The Physiome Model Repository 2." *Bioinformatics (Oxford, England)*, vol. 27, no. 5, pp. 743–4, Mar. 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21216774

[21] L. P. Smith, E. Butterworth, J. B. Bassingthwaighte, and H. M. Sauro, "SBML and CellML translation in Antimony and JSim." *Bioinformatics (Oxford, England)*, Nov. 2013.

[22] H. Elmqvist, H. Olsson, and S. Mattsson, "Optimization for design and parameter estimation," *Proceedings of the 4th International Modelica Conference, Hamburg*, 2005. [Online]. Available: http://elib.dlr.de/12296/

Appendix E

Modeling of short-term mechanism of arterial pressure control in the cardiovascular system: Object-oriented and acausal approach

The paper [5] published as

Tomáš Kulhánek, Jiří Kofránek, and Marek Mateják. Modeling of short-term mechanism of arterial pressure control in the cardiovascular system: Object oriented and acausal approach. *Computers in Biology and Medicine*, 54(1):137–144, 2014. doi:10.1016/j.compbiomed.2014.08.025

Available online at http://dx.doi.org/10.1016/j.compbiomed.2014.08.025

Computers in Biology and Medicine 54 (2014) 137–144

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/cbm

Letter to the Editor

Modeling of short-term mechanism of arterial pressure control in the cardiovascular system: Object-oriented and acausal approach

ARTICLE INFO

Keywords:
Acausal modeling
MODELICA programming language
OPENMODELICA modeling environment
DYMOLA modeling environment
Cardiovascular system

ABSTRACT

This letter introduces an alternative approach to modeling the cardiovascular system with a short-term control mechanism published in Computers in Biology and Medicine, Vol. 47 (2014), pp. 104–112. We recommend using abstract components on a distinct physical level, separating the model into hydraulic components, subsystems of the cardiovascular system and individual subsystems of the control mechanism and scenario. We recommend utilizing an acausal modeling feature of Modelica language, which allows model variables to be expressed declaratively. Furthermore, the Modelica tool identifies which are the dependent and independent variables upon compilation. An example of our approach is introduced on several elementary components representing the hydraulic resistance to fluid flow and the elastic response of the vessel, among others.

The introduced model implementation can be more reusable and understandable for the general scientific community.

1. Introduction

Fernandez de Canete et al. [1] described a closed loop cardiovascular model and short-term and long-term mechanisms of arterial pressure control in the Modelica language and DYMOLA tool. The model is decomposed into several distinct components, which express computation of output volume, pressure and flow, based on elastance input, input flow and pressure. The basic blocks from standard Modelica libraries are composed together to model the whole hemodynamics with the control mechanism [1]. The process of computing of similar flowratevolume, pressure, elastance and resistance is repeated in multiple blocks of pulmonary circulation, systemic circulation and heart circulation [1].

Redundant definition of similar relations and the mixing of more phenomenon in a single component were identified as antipatterns (bad practices) by Tiller [2], who recommends expressing single phenomenon as a general component model. This should be conducted by composing a subsystem model with instances of component models and utilizing the object-oriented features of the Modelica language.

Additionally the model [1] defines the flow of the computation of output values from input values. It is identified as causal or signal-oriented modeling. Modelica allows the expression of models not only in the causal (signal) manner, but also in an acausal manner. The acausal modeling technique is based on the fact that model variables are expressed declaratively, and the Modelica tool identifies which are the dependent and independent variables upon compilation. We have shown that an acausal description captures the essence of the modeled reality much better, and the simulation models are much more legible and, thus, also less prone to mistakes [3]. With the greatest respect to the authors of the above-mentioned publication, we would like to introduce an alternative implementation of the original model with recommendations to reduce redundancy and utilize acausal and object-oriented modeling techniques. An example of our approach is shown with respect to an elastic vessel component with unstressed volume and external pressure.

As a result of the acausal approach we believe that the alternative implementation method introduced can be more reusable by the scientific community for educational as well as for research purposes.

Additionally, Modelica tools allow rich sets of numerical solving methods for further simulation and analysis, as well as export to third party tools used by computational physiologists.

Supplementary materials contain a full source code of the alternative model implementation derived from the original publication in the Modelica language together with dependent libraries.

2. Methods

Model behavior (equations) can be expressed in Modelica in text form and in graphical form – diagram. Later in this paper we present text form as a source code listing and graphical form as detailed by the figures.

To introduce an alternative implementation of the original model, the following recommendations should be followed:

(a) Introduce an acausal connector for a hydraulic domain with “flow” variable q–flowrate and “non-flow” variable p–pressure. Introduce a single component for an elastic vessel, a hydraulic resistor and a cardiac valve with one or two acausal connectors and describe equations of volumetric flow, volume and pressure based on the parameters of elastance and resistance.

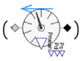
http://dx.doi.org/10.1016/j.compbiomed.2014.08.025
0010-4825/

(b) Separate a general model from a control model and a specific experiment. Utilize object-oriented features of the Modelica language to reuse the architecture model and replace the concrete implementation of the subsystem model with a derived model experiment.
(c) Prefer the text form of Modelica notation to define equations on a component level model. Prefer the diagram form of Modelica notation to express relations between components on a higher subsystem and system level model.

We have recently published a Modelica library, referred to as Physiolibrary [4], to support modeling in the physiological domain. The library contains several hydraulic components that can be directly used to model the cardiovascular system following the recommendation (a). Table 1 contains icon, description and equations characterizing the components. The equations of these components are defined in Modelica using text form following the recommendation (c) and can be seen in supplementary materials. The models of the pulmonary circulation and the systemic circulation are defined by diagrams in Figs. 1 and 2 utilizing the components from Physiolibrary (from Table 1). The models are almost equivalent to the pulmonary and systemic blocks from the original work [1], apart from the pulmonary and aortic valves, that we moved to the model of the heart subsystem described later.

Additionally the systemic circulation contains a block to measure blood properties in aorta, it extends the existing block from Physiolibrary that measures flow. Additionally it computes systolic, diastolic and mean pressure during a single cardiac cycle. The mean arterial pressure P_{mean} during the cardiac cycle is counted as the average of pressure going into the component ($q_{in_{pressure}}$) from the beginning of the cardiac cycle (T_0) during the heart period (HP) by formula:

$$P_{mean} = \frac{\int_{T_0}^{T_0+HP} q_{in_{pressure}}\,\mathrm{d}t}{HP} \tag{1}$$

The component is declared by icon () and by the statements and equations in the following Modelica listing (shortened):

```
model AortaFlowMeasurement "measures flow, diastolic, systolic and mean pressure"
...
  discrete Boolean b(start=false) "beat signal";
  Time T0(start=0) "start of cardiac cycle ";
  discrete Time HP(start=1) "length of cardiac cycle";
initial algorithm
  Ps := q_in.pressure;
  Pd := q_in.pressure;
equation
  Pmax = max(Pmax, q_in.pressure);
  Pmin = min(Pmin, q_in.pressure);
  b = der(q_in.pressure) > 0;
  when {b and not pre(b)} then
    T0 = time "initial time of current cardiac cycle";
    HP = if (pre(T0) > 0) then time - pre(T0) else 1;
    Pmean = SumPressure / pre(HP) "mean pressure";
    Ps = Pmax "systolic pressure = maximum pressure during cardiac cycle";
    Pd = Pmin "diastolic pressure=minimal pressure during cardiac cycle";
    reinit(SumPressure, 0) "reinitialisation of sum pressure";
    reinit(Pmax, q_in.pressure) "reinitialisation of maximal pressure";
    reinit(Pmin, q_in.pressure) "reinitialisation minimal pressure";
  end when;
  der(SumPressure) = q_in.pressure;
end AortaFlowMeasurement;
```

We separate out the variable elastance(compliance) generator from the heart subsystem in Fig. 3.

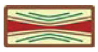
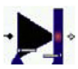
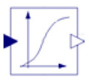
The block "pulsos" (identified by icon () generates the relative heart phase within the heart period during a simulation time based on the heart rate signal [1]. However, in contrast to the original implementation we define it in text form per recommendation (c), and with the changed behavior: the heart period HP is changed per the input signal heart rate only at the moment when the new cardiac cycle begins. The output signal "heartphase" modeled in the original work can be presented as the following equation:

$$heartphase = \frac{time - T_0}{HP} \tag{2}$$

Thus, the model behavior is defined by the following listing:

```
model pulsos  "relative position in heart period"
  discrete Physiolibrary.Types.Time HP(start = 0)
    "heart period - duration of cardiac cycle";
  Boolean b(start = false);
  discrete Physiolibrary.Types.RealIO.TimeOutput T0
    "start time of cardiac cycle";
  Physiolibrary.Types.RealIO.FrequencyInput HR;
  Modelica.Blocks.Interfaces.RealOutput heartphase;
equation
  b = time - pre(T0) >= pre(HP); //new cycle begins?
  when {initial(),b} then
    T0 = time;  //update start time of cardiac cycle
    HP = 1 / HR; //update heart period
  end when;
  heartphase = (time - pre(T0))/pre(HP);
end pulsos;
```

Table 1
Icon and description of hydraulic components used from Physiolibrary [4].

Icon	Description
	Hydraulic connectors – the MODELICA tool generates the following equations to keep "Kirchhoff law" analogy for all connected component's non-flow variables $p_1 \ldots p_n$ - pressure and "flow" variable $q1 \ldots q_n$ - flowrate: $p_1 = p_2 = \cdots = p_n$ (3) $\sum_{i=1}^{n} q_i = 0$ (4)
	Hydraulic Resistor – characterized by G–conductance parameter (reciprocal value of resistance $G = 1/R$) and defined by relation among quantities from both hydraulic connectors, q–flowrate and $(p_{out} - p_{in})$ –pressure gradient: $q = G * (p_{out} - p_{in})$ (5)
	Elastic compartment – characterized by C–compliance parameter(reciprocal value of elastance $C = 1/E$), V_0–unstressed volume parameter, p_0–external pressure parameter and by equation among p–pressure, V–volume and q–flowrate: $p - p_0 = \begin{cases} 0 & \text{if } V < V_0 \\ \frac{V - V_0}{C} & \text{otherwise} \end{cases}$ (6) $\frac{dV}{dt} = q$ (7)
	Valve is characterized by the direction where the flow is allowed, by inflow and backflow conductance
	2D natural cubic interpolation spline defined by x, y and slope points. Used to define curve determined by empirical data

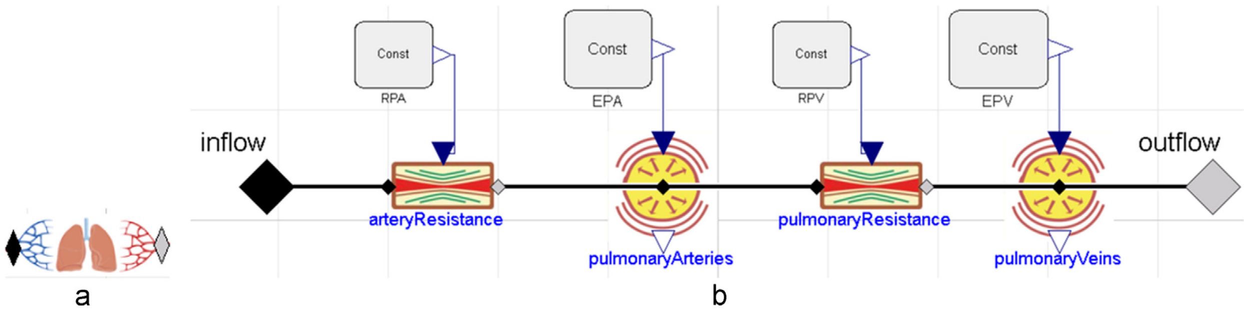


Fig. 1. Pulmonary circulation model icon (a) and its constitutive diagram (b). The elastic compartments and the resistors are connected via hydraulic connectors. The model parameters are presented as block Const with an appropriate type and value.

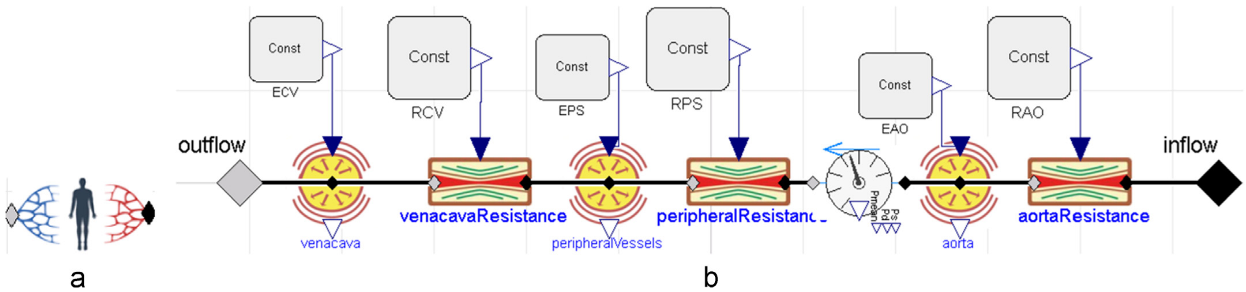


Fig. 2. Systemic circulation icon (a) and its constitutive diagram (b). The elastic compartments and the resistors are connected via hydraulic connectors. The model parameters are presented as block Const with an appropriate type and value.

The heart model in Fig. 4 consists of the left and right parts; they are driven separately by the appropriate elastance generator, but with the same heart rate signal.

Finally, the generic model of the hemodynamics without any control mechanism is shown in Fig. 5. Note that the diagram looks very similar to the usual conceptual decomposition of the cardiovascular system.

Following recommendation (b), the model defines heart, systemicCirculation and pulmonaryCirculation components as "replaceable" to allow replacement with some derived subtype

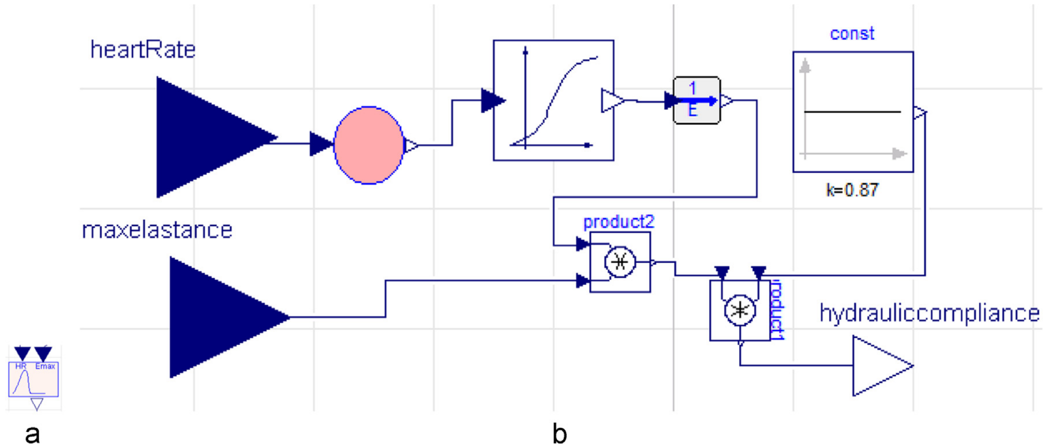


Fig. 3. Compliance generator icon (a) and its constitutive diagram (b). The relative elastance is generated from the relative time position in the heart period via empirical curves generated from points. As Physiolibrary uses a connector for compliance – reciprocal value of elastance – we convert elastance to compliance with a block $1/E$.

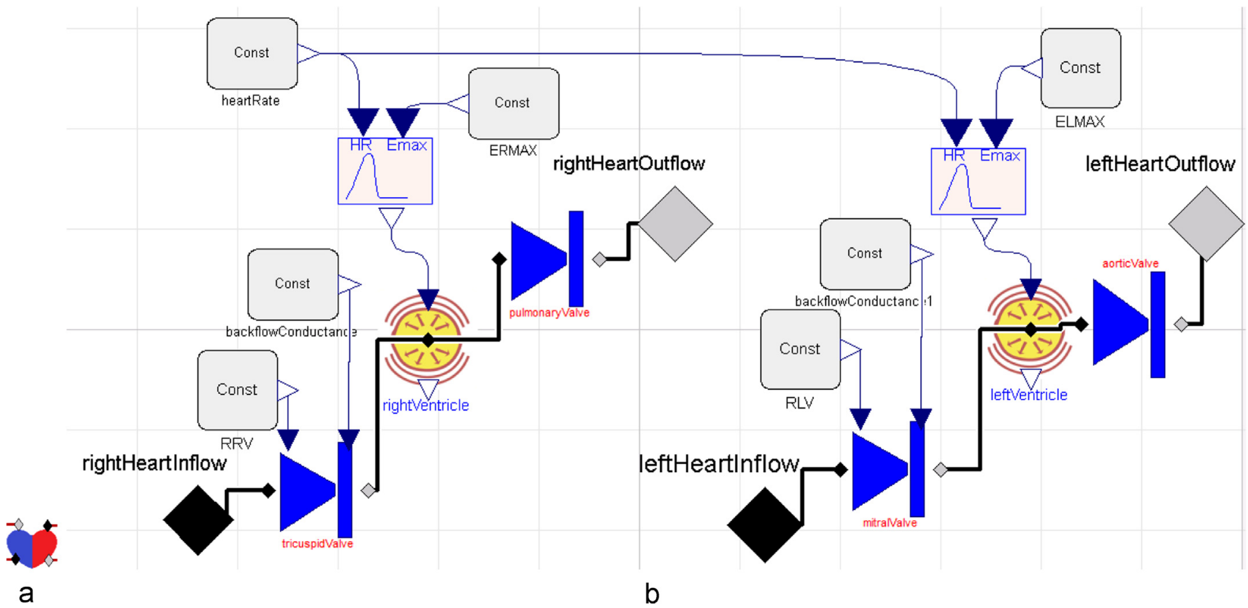


Fig. 4. Heart model icon (a) and its constitutive diagram (b). The valves are connected with the elastic compartment modeling the ventricle driven by the variable elastance component. Heart rate is shared but other parameters are different for each part.

of the model as seen from the text form of the model in the following listing:

```
model Hemodynamics_pure
  replaceable Parts.Heart heart;
  replaceable Parts.SystemicCirculation systemicCirculation;
  replaceable Parts.PulmonaryCirculation pulmonaryCirculation;
...
```

To simulate a particular scenario or control mechanism, selected parameters of the model need to be manipulated externally during simulation. We introduced derived model *SystemicCirculation_baro* and redeclared the constant block with a compatible control block as seen in the following listing which allows manipulation with the model parameter by an external component.

```
model SystemicCirculation_baro
  extends FernandezModel.Parts.SystemicCirculation(
    redeclare HydraulicConductanceControl RPS,
    redeclare HydraulicComplianceControl ECV);
...
```

The subsystem *SystemicCirculation_baro* contains additional input connectors connected with the previously redeclared control blocks as shown in Fig. 6.

A similar technique is used for the heart component in Fig. 7. The model of controllable hemodynamics is an extension of the generic model of hemodynamics. The model uses new implementation of the controllable heart and systemic circulation and introduces input and output connectors specific for the baroreceptor control system, as seen in the following listing and in Fig. 8.

```
model Hemodynamics_controllable
  extends Models.Hemodynamics_pure(
    redeclare Heart_baro heart,
    redeclare SystemicCirculation_baro systemicCirculation);
...
```

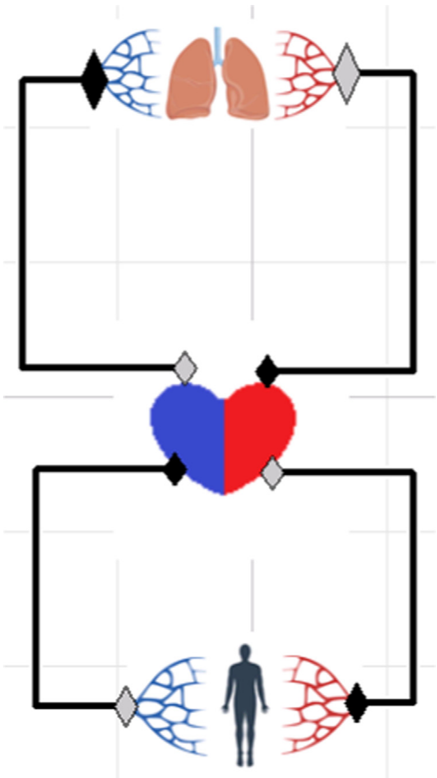


Fig. 5. Complete model of the cardiovascular system without any regulation.

The controllable model of hemodynamics is connected with baroreceptor module and with the module driving the simulation of stenosis, as shown in Fig. 9

3. Results

The implementation was changed in the case of block *pulsos* and block *AortaFlowMeasurement*. Other parts are equivalent to the original work and use the same initial values of state variables and parameters [1]. The model can be executed in a free OPENMODELICA tool developed by Open Source Modelica Consortium [5]. The dynamics of the aortic pressure during a sudden change of elastance of vena cava shown in Fig. 10 gives similar results as the original work. The evolution of heart rate and ventricle elastance is smoother because the mean arterial pressure and the heart period signals are changed only at the beginning of the next cardiac cycle.

4. Discussion and summary

We believe that the recommendation (a) applied to the model allows it to better capture the essence of the modeled reality, in contrast to the original "signal"-like approach where it might be hard to deduce the concept of the model for a user who is not familiar with this modeling technique. Additionally further modification or enhancements of the basic component, e.g., elastic vessel by adding non-linear compliance or active tone, will be propagated to the existing model using this component with no or minimal need for further modification of the model. Such modification will appear in the model within the modified components as modified behavior or new parameters which can be set.

For example, the original model expresses the pressure p, volume V and compliance C (reciprocal value of elastance) as equation $p = V/C$. In contrast to the original model, the alternative implementation modifies the basic element of elastic compartment with unstressed volume V_0 and external pressure p_0 as Eq. (6). V_0 and p_0 are set to 0 by default and can be changed in further model experiments.

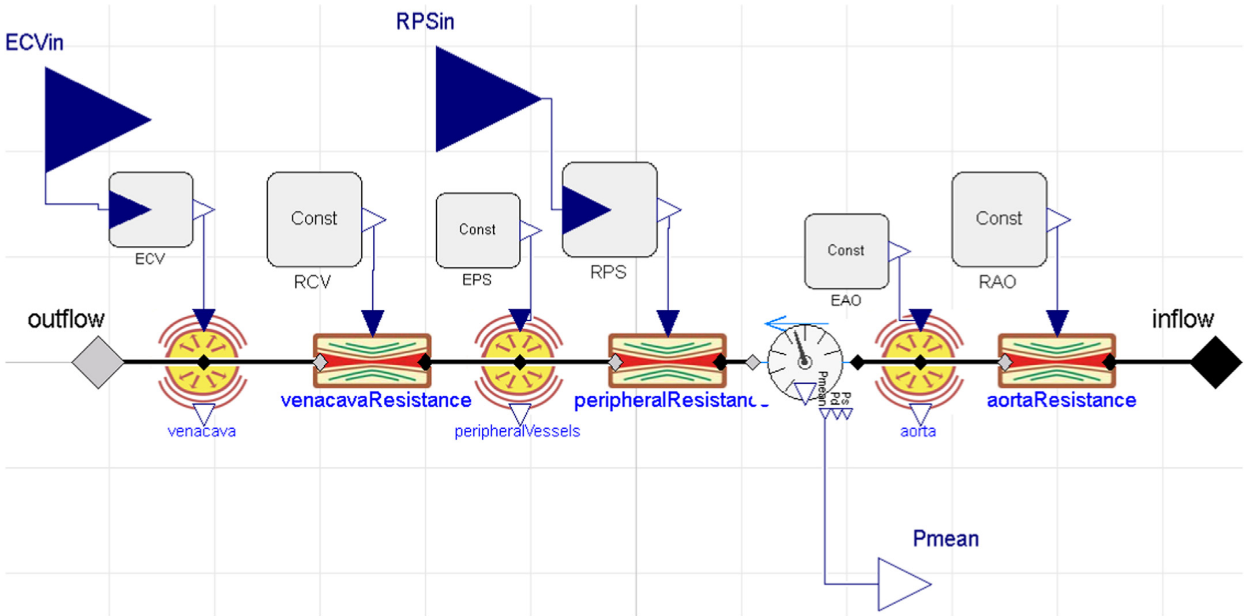


Fig. 6. SystemicCirculation prepared to be driven by elastance of vena cava and peripheral resistance coming from outside.

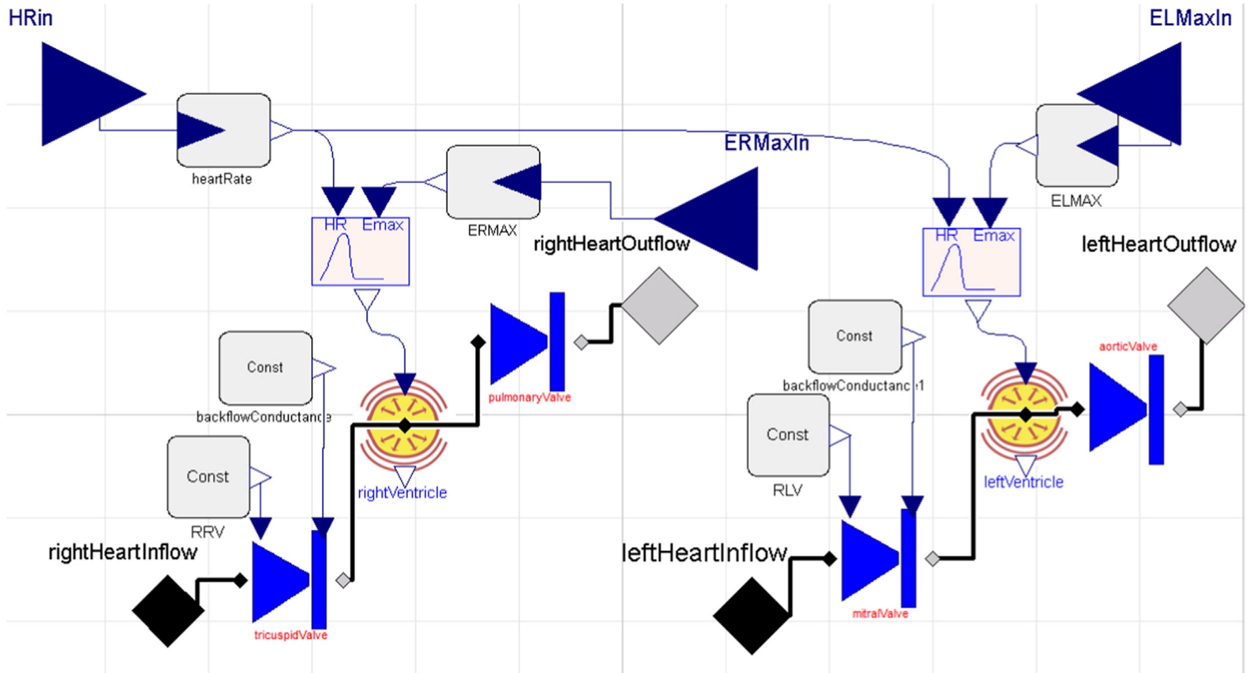


Fig. 7. The Heart model prepared to be driven by an elastance, and a change of the heart rate from outside control mechanism.

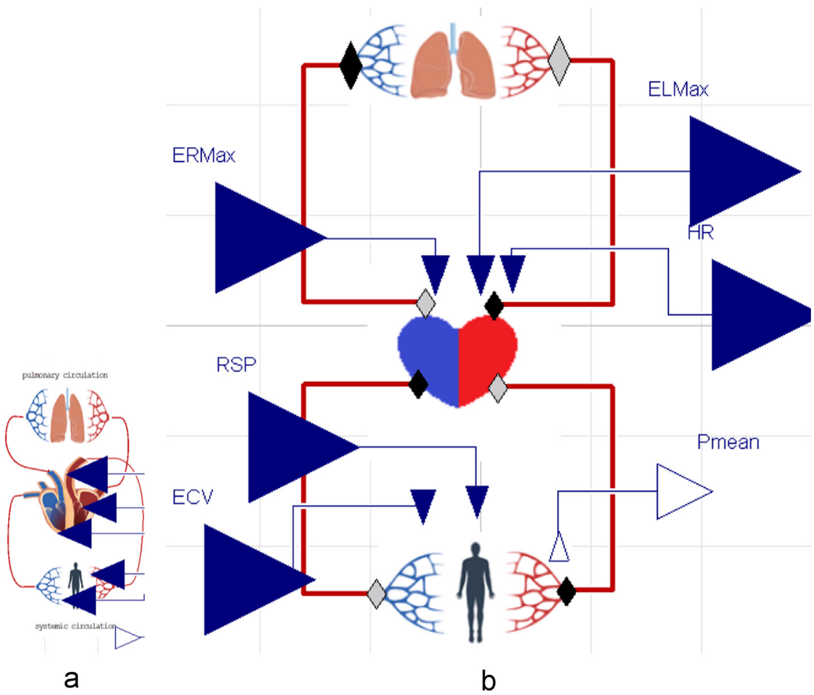


Fig. 8. Icon of the controllable model of hemodynamics (a) extended by input and output connectors and its constitutive diagram (b) connecting new connectors with the redeclared implementation of controllable heart and systemic circulation.

Such enhancement is propagated by inheritance to all elastic compartments used within the model. When the original model [1] needs to be enhanced with such features of unstressed volume or external pressure effect, then more than one component (compartment with valve, compartment without valve and systemic circulation) needs to be modified.

The recommendation (b) applied to the generic model brings the advantage of a major feature of object-oriented programming –

polymorphism. A model (e.g. Hemodynamics_pure) comprises a subsystem model of a certain type (e.g. Heart) will work correctly with another subsystem that is compatible with the certain type (e.g. Heart_baro). The subsystem models might be reimplemented and replaced in an existing model without touching the generic model. Therefore, it may be appropriate for specific in silico experiments.

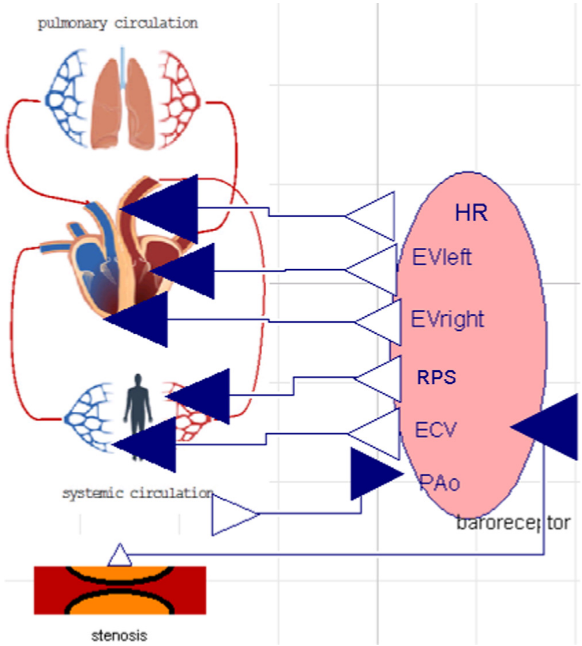


Fig. 9. The model of hemodynamics with baroreceptor control and a module manipulating the elastance during simulation – stenosis of vena cava simulation.

The Modelica standard library allows equations to be expressed using blocks in diagrams; however, it should not be overused. For example, the recommendation (c) applied to the component "pulsos" shows the Eq. (2) cleanly in the text form of Modelica notation (see the listing above). During the reimplementation of this block, we reformulated the method of changing the heart rate signal where heart period and pulse generation are updated at the beginning of the next cardiac cycle. This facilitates the computation. In the original work, the heart period and pulse generation are changed immediately. This difference is seen in simulating the baroreflex control, where the control mechanism is smoother, as seen in Fig. 10.

On the higher subsystem level, the recommendation (c) is optional. If there are more relations among components then a diagram form may be more understandable than a textual form. For example, the block VariableElasticityGenerator in Fig. 3 can be expressed in an equivalent text form as seen in the following model listing:

```
model VariableElasticityGenerator_text
...
equation
  pulsos.HR = heartRate;
  curve.u = pulsos.heartPhase;
  hydrauliccompliance = 0.87 * (maxelastance/curve.val);
end VariableElasticityGenerator_text
```

The original work contains a long-term pressure control mechanism, which was not implemented within this alternative implementation. However, it can be done following the above recommendations similar to the presented short-term "baroreflex" pressure control.

We believe that the introduced alternative approach to modeling the cardiovascular system will enhance the understandability

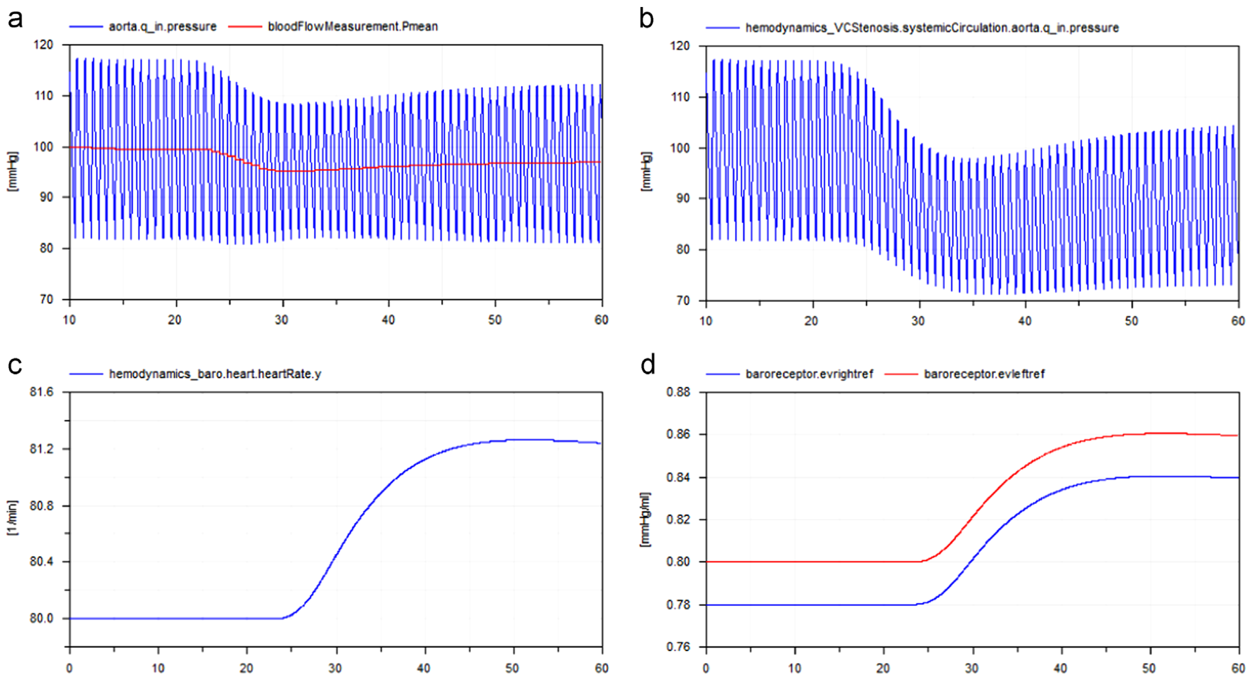


Fig. 10. Simulation of the aortic pressure response on vena cava elastance reduction at the simulation time 20 s for model with (a) and without (b) baroreceptor control showing the evolution of heart rate (c) and ventricle elastance (d).

and reusability of the excellent work done by the authors of the original model.

Conflict of interest statement

None declared.

Acknowledgments

This paper describes the outcome of research that has been accomplished as part of a research program funded by the Ministry of Industry and Trade of the Czech Republic via the grant FR-TI3/869; and by The Ministry of Education, Youth and Sports via the grant SVV-2014-260033.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.compbiomed.2014.08.025.

References

[1] J. Fernandez de Canete, J. Luque, J. Barbancho, V. Munoz, Modelling of long-term and short-term mechanisms of arterial pressure control in the cardiovascular system: an object-oriented approach, Comput. Biol. Med. 47 (2014) 104–112. http://dx.doi.org/10.1016/j.compbiomed.2014.01.006.

[2] M. Tiller, Patterns and anti-patterns in Modelica, in: Proceedings Modelica 2008 Conference, 2008, pp. 647–655.

[3] J. Kofránek, M. Mateják, P. Privitzer, M. Tribula, Causal or acausal modeling: labour for humans or labour for machines, Technical Computing Prague, 2008, pp. 1–16.

[4] M. Mateják, T. Kulhánek, J. Šilar, P. Privitzer, F. Ježek, J. Kofránek, Physiolibrary—modelica library for physiology, in: Proceedings of the Tenth Modelica Conference, March 10–12, 2014, Lund, Sweden, 2014, pp. 499–505. URL ⟨http://dx.doi.org/10.3384/ecp14096499⟩.

[5] P. Fritzson, P. Aronsson, H. Lundvall, K. Nyström, A. Pop, L. Saldamli, D. Broman, The OpenModelica modeling, simulation, and software development environment, Simul. News Eur. 44 (45) (2005).

Tomáš Kulhánek, Jiří Kofránek*, Marek Mateják
Institute of Pathological Physiology, Charles University in Prague, U Nemocnice 5, 12853 Praha 2, Czech Republic
E-mail address: kofranek@gmail.com (J. Kofránek)

Received 15 May 2014; accepted 22 August 2014

* Corresponding author.

Appendix F

Simple Models of the Cardiovascular System for Educational and Research Purposes

The paper [6] published as

Tomáš Kulhánek, Martin Tribula, Jiří Kofránek, and Marek Mateják. Simple models of the cardiovascular system for educational and research purposes. *MEFANET Journal*, 2(2):56–63, 2014

Available online at `http://mj.mefanet.cz/mj-04140914`

SIMPLE MODELS OF THE CARDIOVASCULAR SYSTEM FOR EDUCATIONAL AND RESEARCH PURPOSES

Tomáš Kulhánek*, Martin Tribula, Jiří Kofránek, Marek Mateják
Institute of Pathological Physiology, 1st Faculty of Medicine, Charles University in Prague, Prague, Czech Republic
** Corresponding author: tomaton@centrum.cz*

ARTICLE HISTORY
Received 14 September 2014
Revised 30 October 2014
Accepted 4 November 2014
Available online 24 November 2014

KEYWORDS
Modelica
modeling physiology
cardiovascular system
Physiolibrary

ABSTRACT — *Modeling the cardiovascular system as an analogy of an electrical circuit composed of resistors, capacitors and inductors is introduced in many research papers. This contribution uses an object oriented and acausal approach, which was recently introduced by several other authors, for educational and research purpose. Examples of several hydraulic systems and whole system modeling hemodynamics of a pulsatile cardiovascular system are presented in Modelica language using Physiolibrary.*

INTRODUCTION

The mathematical formalization of the cardiovascular system (CVS), i.e. the models, can be divided into two main approaches. The first approach builds 3D models with geometrical, mechanical properties and the time-dependence of the compartments of the CVS. The complexity of such models implicates high demand on computational power to simulate them.

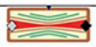
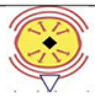
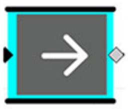
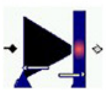
The second approach, lumped parameter approach, uses a high degree of simplification and different complex regions are generalized as single compartments characterized by lumped parameter. Using this approach, the cardiovascular system is modeled as an analogy of an electrical circuit with a set of resistors, capacitors (elastic vessels), diodes (valves) and inductance (inertial elements) in a closed loop. The relationship between pressure and volume quantities is studied throughout the modeled system. This type of model is used for the improvement of reliability in patient-specific diagnosis, for example [1,2]; or for educational purposes and training [3–9]. Technology used for modeling is either proprietary, e.g. MATLAB–Simulink or MATLAB-Simscape in the case of the CVS model introduced by Fernandez et al.[7], home grown open technology introduced by Hester et al.[3], standard grown technology from the scientific community, such as SBML [10], JSIM [11] or CellML [1,12] or industrial standard technology implemented by several vendors, such as the Modelica language [8,9].

This article introduces a modeling method which follows the second approach for modeling CVS as pressure-volume relations and introduces example models implemented in the Modelica language. We believe that one of the properties of the Modelica language — acausal modeling — seems to capture the modeled reality much better and allows connecting simpler models into quite complex systems of differential equations in an understandable form [4,13,14]. Additionally, we use a library for modeling physiology in Modelica — Physiolibrary — as the model diagrams based on the Physiolibrary components are self-descriptive in most cases [15,16].

METHODS

Modelica is an object-oriented language, thus, in further text when we refer to components, models and systems they are classes, and the one's with concrete values of parameters are instances as is usual in standard object-oriented programming. Models (classes) can inherit behavior or can be composed from other

TABLE 1 Selected components from Physiolibrary with description

Icon	Description
	**Hydraulic connectors** – declared as acausal, without prejudicing any kind of computational order. The MODELICA tool generates the following equations to keep the analogy of "Kirchhoff laws" for all connected components "non-flow" variables *p1..pn* determining the pressure and "flow" variable *q1..qn* determining flowrate: $p_1=p_2=...=p_n$ (1) $\sum_{i=1}^{n} q_i=0$ (2)
	**Hydraulic conductor** or (hydraulic resistor) is characterized by the parameters *G*-conductance (reciprocal value of resistance $G=\frac{1}{R}$) and models the relation between pressures p_{in}, p_{out} and flowrates q_{in}, q_{out} from the two connectors by the following formulas: $q_{out}=-q_{in}$ (3) $q_{in}=G*(p_{out}-p_{in})$ (4)
	**Elastic vessel** is characterized by these parameters: *C*-compliance (reciprocal value of elastance $C=\frac{1}{E}$), *V0* – unstressed volume, *p0* – external pressure and by equation among the variable *V*-volume and variables of the hydraulic connector *q*-flowrate and *p*-pressure: $p-p_0=\begin{cases}0 & \text{if } V<V_0 \\ (V-V_0)/C & \text{otherwise}\end{cases}$ (5) $\frac{dV}{dt}=q$ (6)
	**UnlimitedPump** is defined as the generator of the flowrate q_{out} based on the value "solutionFlow". $q_{out}=-solutionFlow$ (7)
	**Inertia element** is characterized by the *I* - inertance parameter and the relationship between pressure and solution flow from the two connectors by the following formula: $q_{out}=-q_{in}$ (8) $\frac{dq_{in}}{dt}=\frac{p_{in}-p_{out}}{I}$ (9)
	**Hydraulic valve** is characterized by g_{on} outflow and g_{off} backflow conductances, described by parametric equations: $dp=\begin{cases} pass/g_{on}+P_{knee} & \text{for } pass>0 \\ pass+P_{knee} & \text{otherwise}\end{cases}$ (10) $q=\begin{cases} pass+P_{knee}*g_{off} & \text{for } pass>0 \\ pass*g_{off}+P_{knee}*g_{off} & \text{otherwise}\end{cases}$ (11)

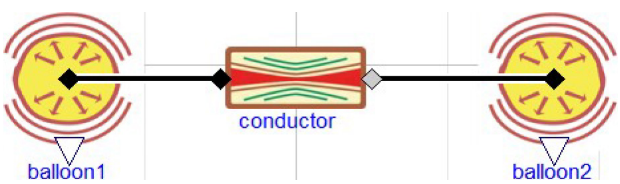


FIGURE 1 Two balloon example model diagram. It connects two balloons characterized with compliance (elastance) via a conductor (resistor).

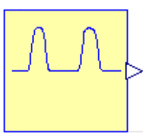


FIGURE 2 Icon of component "pulses"

models (classes). For the composition a special purpose class — connector — can be used to define variables of the model shared with other models.

The following models in the Modelica language are usually declared by an icon and defined either in diagram view or textual notation. The icon declares an interface of how the particular model instance can appear in other higher level model diagrams. The diagram and textual view declares the internal structure, which are in fact algebraic and differential equations among model parameters and variables.

As was mentioned earlier, Modelica models can be acausal, the equations can be expressed declaratively and the Modelica tool will decide which variables are dependent and independent based on the context upon compilation. This is extremely useful when several existing components are composed into a larger system model and the complexity is kept in a maintainable set of diagrams.

In further examples we use the following components from Physiolibrary (Table 1).

Two connected balloons

The model diagram (Figure 1) expresses the situation of two elastic balloons filled with liquid connected via an inelastic tube with characteristic resistance. Such a model diagram connects the system of equations of the two elastic vessels and one resistor and Modelica will figure out which variables will be independent and which will be dependent. In this example the pressure will be computed first from the initial volume of the two balloons by equation (5). This determines the pressure gradient, which directs the flow from the first balloon to the second one in equation (4). The flowrate affects the volume change in each elastic vessel in equation (6) and again causes the change of pressure from equation (5).

Windkessel models

The Windkessel models simplify the view on CVS as a series of compartments with capacity (compliance or elasticity of an elastic compartment), resistance (conductance or resistance of a conductor/resistor), impedance and inertance properties. They study the Windkessel effect, which maintains a relatively stable flowrate from the system, although the pulsatile flowrate comes from the pump. Several derivatives and improvements of Windkessel models were introduced [17].

Figures 3, 4 and 5 show diagrams of the Windkessel model of 2 elements, 3 elements and 4 elements in Modelica using Physiolibrary components. Additionally, the model diagrams contain component 'pulses' declared by the icon in Figure 2 and defined by following Modelica listing:

```
model pulses
  import Physiolibrary.Types.*;
  Physiolibrary.Types.RealIO.VolumeFlowRateOutput
volumeflowrate;
  discrete Time T0 “beginning of cardiac cycle”;
  Boolean b(start=false);
  discrete Time HP “duration of cardiac cycle”;
  parameter Frequency HR = 1.2;
  Time tc “relative time in cardiac cycle”;
  parameter Time TD1=0.07 “relative time of start of
systole”;
  discrete Time TD2 “relative time of end of systole”;
  parameter VolumeFlowRate QP = 0.000424 “peak volume
flowrate”;
equation
  b = time - pre(T0) >=pre(HP) “true if new cardiac cycle
begins”;
  when {initial(),b} then
    T0 = time “set beginning of cardiac cycle”;
    HP=1/HR “update length of cardiac cycle”;
    TD2 = TD1+(2/5)*HP “compute end time of systole”;
  end when;
  tc = time-T0 “relative time in cardiac cycle”;
  volumeflowrate =
    if tc<TD1 then 0 else
    if tc<TD2 then sin((tc-TD1)/(TD2-TD1)*Modelica.
Constants.pi)*QP else
    0 “zero before and after systole, otherwise sin up to
peak flow”;
end pulses;
```

This component generates regular pulses of flowrate during the systole period, which are approximated by the sinus function increasing from zero up to the peak flowrate and back to zero during 2/5 of a cardiac cycle, while at other times it generates a zero signal. The keyword ‘discrete’ notes the Modelica tool, wherein such variables will be changed only in discrete events, in our case at the beginning of the cardiac cycle.

A simple model of the cardiovascular system

We divide the CVS system into systemic circulation and pulmonary circulation; each circulation is characterized by a modified 2-element Windkessel model with an additional compliance component expressing the systemic and pulmonary veins (Figure 6).

The ‘rightHeart’ and ‘leftHeart’ components are instances of the model ‘HeartPump’. The flowrate of the model “HeartPump” is determined by the filling pressure and by the slope of the Starling curve. The equations are defined in the following Modelica text notation:

```
model HeartPump
  Physiolibrary.Hydraulic.Interfaces.HydraulicPort_a
inflow “inflow”;
  Physiolibrary.Hydraulic.Interfaces.HydraulicPort_b
outflow “outflow”;
  parameter Physiolibrary.Types.HydraulicConductance
StarlingSlope;
equation
  inflow.q + outflow.q =0;
  inflow.q = StarlingSlope * inflow.pressure;
end HeartPump;
```

The non-pulsatile model shows the mean values of pressure and flow throughout CVS. To enhance the above mentioned model, we first define the pulsatile model PulsatileHeartPump by diagram (Figure 7).

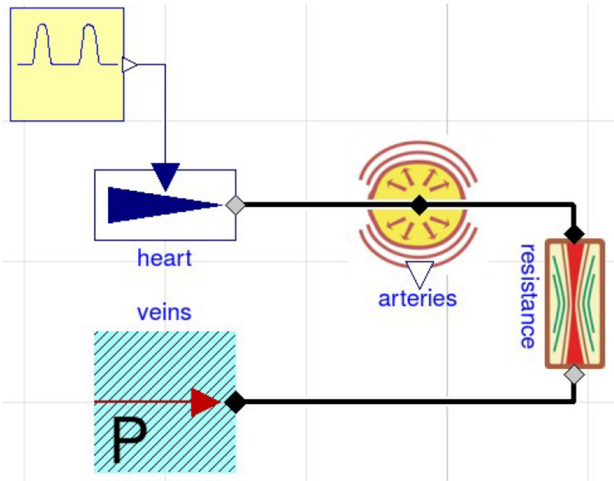


FIGURE 3 2-element Windkessel model characterized by compliance component (arteries) C = 1.4 ml/mmHg) and resistance with conductance G = 1.08 ml/(mmHg.s)

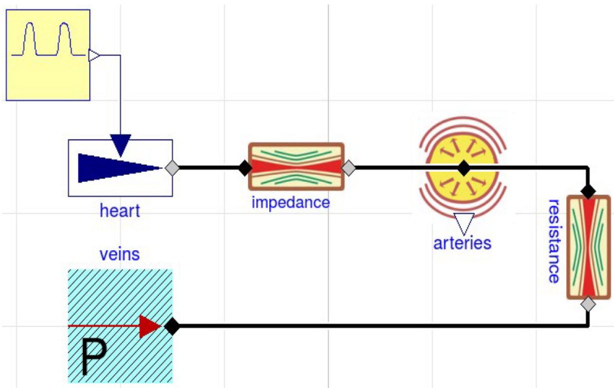


FIGURE 4 3-element Windkessel model characterized by impedance (here approximated by conductance = 200ml/(mmHg.s)), compliance component (arteries) C = 1.4 ml/mmHg) and resistance with conductance G = 1.08 ml/(mmHg.s)

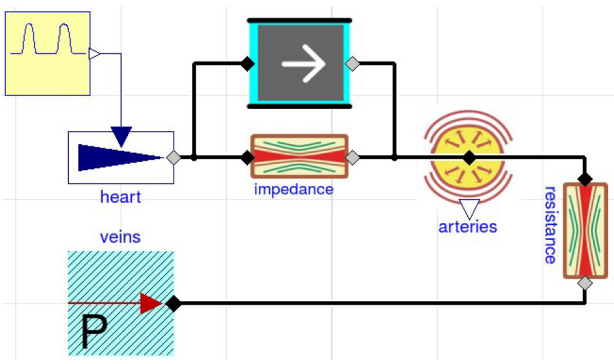


FIGURE 5 4-element Windkessel model characterized by impedance (here approximate by conductance = 200ml/(mmHg.s)), inertia I = 0.005 mmHg.s^2/ml, compliance component (arteries) C = 1.4 ml/mmHg) and resistance with conductance G = 1.08 ml/(mmHg.s)

We utilize the object-oriented features in Modelica to extend the non-pulsatile circulation by only replacing the heartPump instances by pulsatile heart pump instances as seen in the following listing. Note the redefinition of values (in SI units) of parameter QP, k, volume_start:

```
model PulsatileCirculation
  extends NonPulsatileCirculation(
    redeclare Parts.PulsatileHeartPump
rightHeart(pulses(QP = 0.000338)),
    redeclare Parts.PulsatileHeartPump
leftHeart(pulses(QP=0.000338)),
    CAS(k=7.2755972857029e-09),
    SystemicArteries(volume_start=0.000603),
    SystemicVeins(volume_start=0.003991));
end PulsatileCirculation;
```

REFERENCE MODEL WITH VENTRICLES

For further comparison purposes, we have chosen model of CVS published by Fernandez de Canete et al. implemented originally in MATLAB-Simscape [7] . This model is presented in Modelica using Physiolibrary components in one diagram (Figure 8). Systemic and pulmonary circulation are presented as modified Windkessel model with inertia element connected via heart subsystem. Additionally, each side of heart is composed by two valves and a ventricle. The ventricle is modeled as an elastic compartment driven by variable elasticity generator defined by the following Modelica listing:

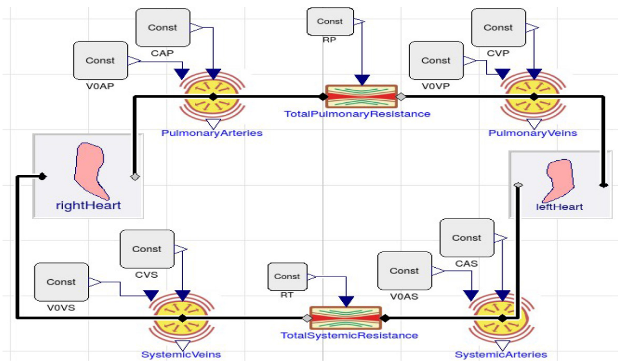


FIGURE 6 Non-pulsatile circulation model diagram. Pulmonary circulation on top of the diagram is connected via side of the heart with systemic circulation at the bottom of the diagram

```
model TimeVaryingElastance
  parameter Physiolibrary.Types.HydraulicElastance Ed
"e. of diastole";
  parameter Physiolibrary.Types.HydraulicElastance Es
"e.of systole";
  parameter Physiolibrary.Types.Pressure Pi0 "peak
isovolumic pressure";
  Physiolibrary.Types.Time tm
    "relative time from the beginning of cardiac cycle";
  discrete Physiolibrary.Types.Time HP "heart period";
  discrete Physiolibrary.Types.Time t0 "start time of the
cardiac cycle";
  discrete Physiolibrary.Types.Time ts "duration of
systole";
  Real a;
  Physiolibrary.Types.RealIO.HydraulicComplianceOutput C;
  Physiolibrary.Types.HydraulicElastance E;
  Physiolibrary.Types.RealIO.PressureOutput Pi;
  Physiolibrary.Types.RealIO.FrequencyInput HR "heart
rate";
equation
  tm = time - pre(t0);
  if (tm<pre(ts)) then
    a= (1-cos(2*Modelica.Constants.pi*tm/pre(ts)))/2;
  else
    a=0;
  end if;
  E=Ed+Es*a;
  C=1/E;
  Pi = Pi0*a;
  when {initial(), tm >= pre(HP)} then
    HP = 1/HR;
    t0= time;
    ts = 0.16+0.3*HP;
  end when;
end TimeVaryingElastance;
```

RESULTS

The two balloons model (Figure 1) is used in simulators to demonstrate that a liquid (e.g. blood) flows from the part with higher pressure to the part with lower pressure. If there is no other active force, the system converges to an equilibrium, where the pressures will be equal and no other flow occurs between the balloons (Figure 9).

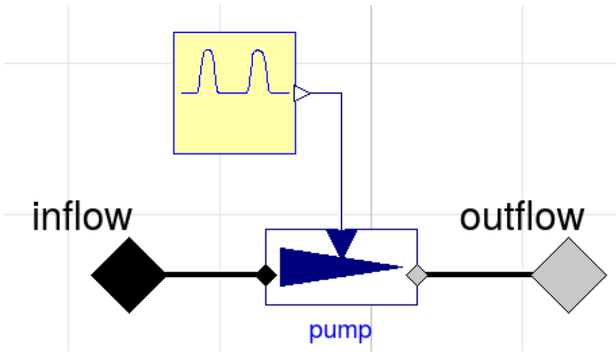


FIGURE 7 Pulsatile heart pump model

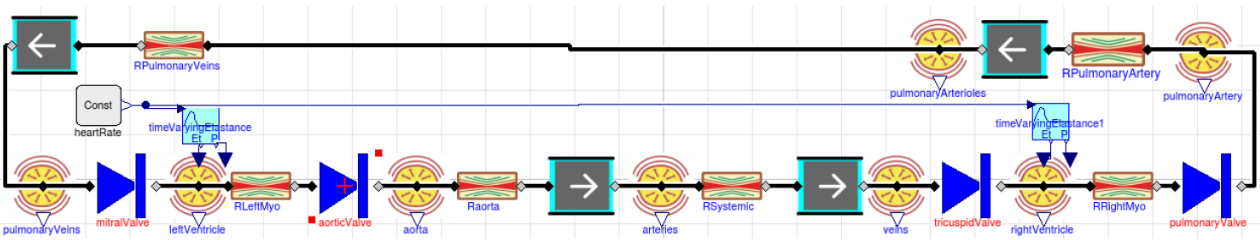


FIGURE 8 Reference model by Fernandez de Canete et al. [7] in Modelica using Physiolibrary components

This model and simulation is used to demonstrate that the blood has a tendency to flow from the arteries with lower compliance and higher pressure to the veins. Without any external force, the majority of blood accumulates in veins rather than in arteries.

Simulation of the Windkessel models imitate the heart pulses by the generator controlled with the 'pulses' component (Figure 10).

The simulation of Windkessel model shows the Windkessel effect, which reduces a high variation of flowrate coming from the heart to a relatively stable flowrate in the systemic peripheral vessels due to the compliance of the elastic compartment.

The simulation of newly introduced extended model with pulsatile circulation (Figure 11) shows approximate dynamic pressure in aorta and pulmonary artery compared to the same variables of non-pulsatile model showing rather a mean values. The values of parameters and initial values of state variables are in Table 2 and 3 using normal physiological units as well as SI units.

The simulation of reference model by Fernandez de Canete et al. (Figure 8) was performed using original values of parameters (Table 4). The pressure dynamics in aorta and pulmonary arteries (Figure 12) shows quite realistic dicrotic notch after closure of aortic valve. Detailed pressure dynamics of aorta and left ventricle during one cardiac cycle is in Figure 13.

DISCUSSION

Our non-pulsatile and pulsatile models generate a raw approximation of outgoing flowrate. However, other models, such as the model developed by Fernandez de Cante et al. [7,8], or that developed by Meurs et al. [5,6], introduce a heart with elastic vessels driven by variable compliance and valves rather than a pump with variable outgoing flowrate. The shapes of the pressure curves in these models are closer to the real experimental measurements.

In order to study more specific phenomena, additional components can be added to the model. For instance, our implementation does not use the inertia element and valves. This seems to be useful for examining the effect of a dicrotic notch in the arterial pressure when the aortic valve closes as seen, for instance, in the model and simulation developed by Fernandez de Canete et al. [7]. Figure 14 shows comparison with simple pulsatile model.

Our implementation uses one resistance and two elastic vessel components for the pulmonary and systemic circulation with total of 16 parameters and initial values of state variables (Table 2). The pulsatile model adds new 4 parameters and redefine values of three existing (Table 4). This seems satisfactory to explain basic physiological and pathophysiological phenomena, e.g. influence of congestive left or right heart failure to volumes of blood in systemic or pulmonary

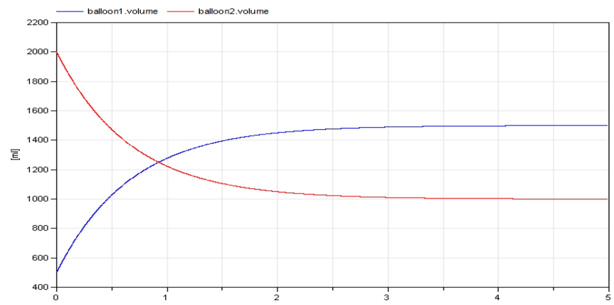


FIGURE 9 Simulation of volume dynamics from 500 ml to 1500 ml for the first balloon and from 2000 ml to 1000 ml for the second balloon. Unstressed volumes of 500 ml for both balloons and compliances of 2 ml/mmHg and 1 ml/mmHg and conductance 1 ml/(mmHg.s). After five seconds the system is almost at equilibrium

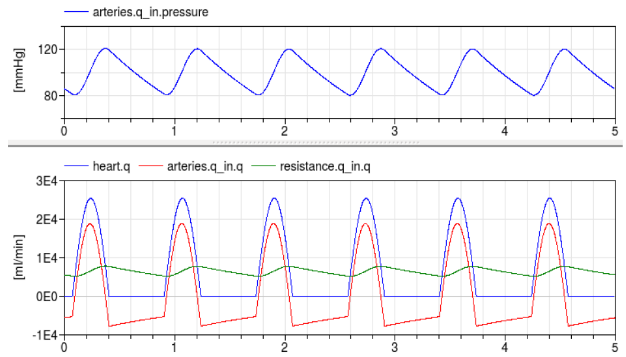


FIGURE 10 Simulation of 4-element Windkessel model characterized by impedance (approximate by conductance = 200 ml/(mmHg.s)), inertia I = 0.005 mmHg.s^2/ml, compliance component (arteries) C = 1.4 ml/mmHg) and resistance with conductance G = 1.08 ml/(mmHg.s) and initial volume of arteries at 0.97 l and unstressed volume of arteries at 0.85 l. The pressure during six beats is kept between 120/80 and the flowrate going from the heart ranging from 0 to 25 l per minute (blue) is compensated by the compliance compartment flowrate going from −5 to +18 l per minute (red) to the resulting average flowrate going from peripheral arteries 5 to 7 l per minute (green)

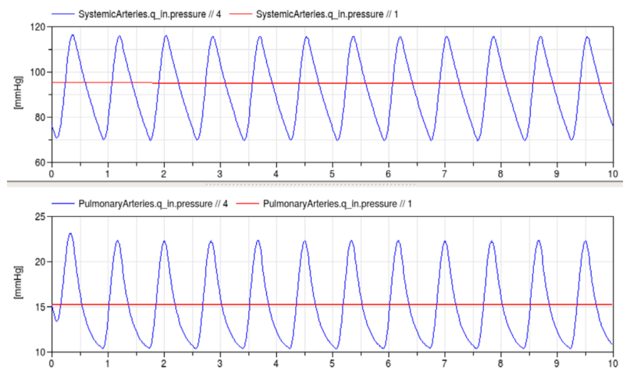


FIGURE 11 Pressure dynamics in systemic and pulmonary arteries simulated by both non-pulsatile (red) and pulsatile CVS models (red)

TABLE 2 Initial values of state variables and parameters of the non-pulsatile model of CVS in Modelica. Asterisk indicates initial volume of a component computed from steady state simulation

Parameter	Value	Value in SI	Ref.
V0CAS.k (unstressed volume of systemic arteries)	529 ml	5.29×10^{-4} m³	
CAS.k (compliance of systemic arteries)	1.5 ml/mmHg	1.13×10^{-8} m³/Pa	
systemicArteries.volume_start	672 ml	6.72×10^{-4} m³	*
V0VS.k (unstressed volume of systemic veins)	2845 ml	2.845×10^{-3} m³	
CVS.k (compliance of systemic veins)	200 ml/mmHg	1.50×10^{-6} m³/Pa	
systemicVeins.volume_start	3922 ml	3.922×10^{-3} m³	*
V0AP.k (unstressed volume of pulmonary arteries)	327 ml	3.27×10^{-4} m³	
CAP.k (compliance of pulmonary arteries)	3.01 ml/mmHg	2.26×10^{-8} m³/Pa	
pulmonaryArteries.volume_start	373 ml	3.73×10^{-4} m³	*
V0VP.k (unstressed volume of pulmonary veins)	435 ml	4.35×10^{-4} m³	
CVP.k (compliance of pulmonary veins)	30 ml/mmHg	2.25×10^{-7} m³/Pa	
pulmonaryVeins.volume_start	704 ml	7.04×10^{-4} m³	*
RT.k (total systemic resistance)	1 (mmHg.s)/ml	1.33×10^{8} (Pa.s)/m³	
RP.k (total pulmonary resistance)	0.07 (mmHg.s)/ml	9.33×10^{6} (Pa.s)/m³	
rightHeart.StarlingSlope	16.67 ml/(mmHg.s)	1.25×10^{-7} m³/(Pa.s)	
leftHeart.StarlingSlope	10 ml/(mmHg.s)	7.50×10^{-8} m³/(Pa.s)	

TABLE 3 Additional initial values of state variables and parameters of the pulsatile model of CVS in Modelica. Decreased value of CAS and changed initial volumes of systemic arteries and veins to compare with reference model. Other values inherited from non-pulsatile model (Table 2)

Parameter	Value	Value in SI	Ref.
CAS.k (compliance of systemic arteries)	0.97 ml/mmHg	7.28×10^{-9} m³/Pa	
systemicArteries.volume_start	603 ml	6.03×10^{-4} m³	*
systemicVeins.volume_start	3991 ml	3.991×10^{-3} m³	*
rightHeart.pulses.QP (peak flow of right heart)	20.28 l/min	3.38×10^{-4} m³/s	
leftHeart.pulses.QP (peak flow of left heart)	20.28 l/min	3.38×10^{-4} m³/s	
pulses.HR (default heart rate)	72 beats per min	1.2 Hz	
Pulses.TD1 (relative time of systole start)	0.07 s	0.07 s	

TABLE 4 Initial values of state variables and parameters of the reference model by Fernandez de Canete et al. [7] in Modelica. Values with reference were taken from the original publication, asterisk indicates initial values computed from steady state simulation, other values were estimated. "ZeroPressureVolume" is unstressed volume of a component, "volume_start" is initial volume of a component. "_Gon" is conductance of a valve in opened state. Compliance and conductance values were counted as reciprocal (1/×) of original elastance and resistance values

Parameter	Value	Value in SI	Ref.
aorta.ZeroPressureVolume	30 ml	3.00×10^{-5} m³	
aorta.Compliance	0.22 ml/mmHg	1.65×10^{-9} m³/Pa	[7]
aorta.volume_start	46 ml	4.60×10^{-5} m³	*
arteries.ZeroPressureVolume	700 ml	7.00×10^{-4} m³	
arteries.Compliance	1.46 ml/mmHg	1.10×10^{-8} m³/Pa	[7]
arteries.volume_start	805 ml	8.05×10^{-4} m³	*
veins.ZeroPressureVolume	2370 ml	2.37×10^{-3} m³	
veins.Compliance	20 ml/mmHg	1.50×10^{-7} m³/Pa	[7]
veins.volume_start	2443 ml	2.44×10^{-3} m³	*
pulmonaryArtery.ZeroPressureVolume	20 ml	2.00×10^{-5} m³	
pulmonaryArtery.Compliance	0.09 ml/mmHg	6.75×10^{-10} m³/Pa	[7]
pulmonaryArtery.volume_start	21 ml	2.10×10^{-5} m³	*
pulmonaryArterioles.ZeroPressureVolume	600 ml	6.00×10^{-4} m³	
pulmonaryArterioles.Compliance	2.67 ml/mmHg	2.00×10^{-8} m³/Pa	[7]

Parameter	Value	Value in SI	Ref.
pulmonaryArterioles.volume_start	637 ml	6.37×10^{-4} m³	*
pulmonaryVeins.ZeroPressureVolume	100 ml	1.00×10^{-4} m³	
pulmonaryVeins.Compliance	46.7 ml/mmHg	3.50×10^{-7} m³/Pa	[7]
pulmonaryVeins.volume_start	659.7 ml	6.597×10^{-4} m³	*
heartRate.k	72 1/min	1.2 Hz	
leftVentricle.ZeroPressureVolume	90 ml	9.00×10^{-5} m³	
leftVentricle.volume_start	209.7 ml	2.097×10^{-4} m³	*
timeVaryingElastanceLeft.Ed	0.1 mmHg/ml	1.33×10^{7} Pa/m³	[7]
timeVaryingElastanceLeft.Es	1.375 mmHg/ml	1.83×10^{8} Pa/m³	[7]
timeVaryingElastanceLeft.Pi0	50 mmHg	6.66×10^{3} Pa	[7]
rightVentricle.ZeroPressureVolume	70 ml	7.00×10^{-5} m³	
rightVentricle.volume_start	180 ml	1.80×10^{-4} m³	
timeVaryingElastanceRight.Ed	0.03 mmHg/ml	4.00×10^{6} Pa/m³	[7]
timeVaryingElastanceRight.Es	0.328 mmHg/ml	4.37×10^{7} Pa/m³	[7]
timeVaryingElastanceRight.Pi0	24 mmHg	3.20×10^{3} Pa	[7]
mitralValve._Gon	266.6 ml/(mmHg.s)	2.00×10^{-6} m³/(Pa.s)	[7]
RleftMyo.Conductance	12.5 ml/(mmHg.s)	9.37×10^{-8} m³/(Pa.s)	[7]
aorticValve._Gon	266.6 ml/(mmHg.s)	2.00×10^{-6} m³/(Pa.s)	[7]
Raorta.Conductance	14.81 ml/(mmHg.s)	1.11×10^{-7} m³/(Pa.s)	[7]
Rsystemic.Conductance	1 ml/(mmHg.s)	7.50×10^{-9} m³/(Pa.s)	[7]
tricuspidValve._Gon	266.6 ml/(mmHg.s)	2.00×10^{-6} m³/(Pa.s)	[7]
RRightMyo.Conductance	57.14 ml/(mmHg.s)	4.29×10^{-7} m³/(Pa.s)	[7]
pulmonaryValve._Gon	266.6 ml/(mmHg.s)	2.00×10^{-6} m³/(Pa.s)	[7]
RPulmonaryArtery.Conductance	29.62 ml/(mmHg.s)	2.22×10^{-7} m³/(Pa.s)	[7]
RPulmonaryVeins.Conductance	9.9 ml/(mmHg.s)	7.43×10^{-8} m³/(Pa.s)	[7]
aorticInertia.I	8.25×10^{-4} mmHg.s²/ml	1.10×10^{5} Pa.s²/m³	[7]
aorticInertia.volumeFlow_start	623.1 ml/min	1.04×10^{-5} m³/s	*
systemicInertia.I	3.6×10^{-3} mmHg.s²/ml	4.80×10^{5} Pa.s²/m³	[7]
systemicInertia.volumeFlow_start	4761 ml/min	7.94×10^{-5} m³/s	*
pulmonaryArterialInertia.I	7.5×10^{-4} mmHg.s²/ml	1.00×10^{5} Pa.s²/m³	[7]
pulmonaryArterialInertia.volumeFlow_start	43.94 ml/min	7.32×10^{-7} m³/s	*
pulmonaryVeinsInertia.I	3.08×10^{-3} mmHg.s²/ml	4.11×10^{5} Pa.s²/m³3	[7]
pulmonaryVeinsInertia.volumeFlow_start	1335 ml/min	2.23×10^{-5} m³/s	*

veins. The more realistic models with additional components introduce also additional parameters and initial values for state variables. The CVS model by Fernandez de Canete et al. [7] has 48 parameters (Table 4) and this may cause some confusion when teaching only the effects described above. But these more complex models are, however, more suitable to study e.g. the phases within cardiac cycle as seen in Figure 12.

The non-pulsatile model (Figure 6) is a base of an existing educational application simple circulation available in the Atlas of Physiology and Pathophysiology (www.physiome.cz/atlas) [18] and is used in teaching physiology and pathophysiology of cardiovascular system of students of medicine and biomedical engineering[19,20]. The new pulsatile model is used in further development of the simulators to describe additionally the Windkessel effect. The simulators are introduced in lectures of pathophysiology for students of medicine and several basic scenarios are demonstrated. Students, according to surveys, use the simulators again at home when they prepare for tests and exams. This has been acknowledged as an improvement of understanding complex relation and mechanism in CVS compared to static text or non-interactive e-learning materials.

The models introduced above are used in teaching modeling and simulation of students of biomedical engineering. Because the physical system can be decomposed into basic components and modeled in an understandable form, the students focus much more on system analysis rather than on implementation issues [21].

There are some general limitations to the models based on the Windkessel phenomenon and modeling pressure volume relationship, e.g. wave transmission and wave travel cannot be studied.

CONCLUSION

Modeling technique and example models were introduced to demonstrate some effects that are important in understanding the physiology of the cardiovascular system, e.g. flow is determined by pressure gradient and Windkessel effect stabilizes the flowrate changes going from the heart to the peripheral vessels.

The implementation in Modelica language, utilizing the open source Physiolibrary, allows the expression of a complex system of differential and algebraic equations in self-descriptive model diagrams. The mathematics is hidden in the low level component model definition.

The newly introduced pulsatile model of CVS is used in further development of educational simulators.

A full source code of presented models is attached as supplementary materials to this paper and can be tried in the commercial tool Dymola or in the open source tool OpenModelica (www.openmodelica.org) using Physiolibrary (www.physiolibrary.org).

Mgr. Tomáš Kulhánek

ACKNOWLEDGMENTS

This paper describes the outcome of research that has been accomplished as part of a research program funded by the Ministry of Industry and Trade of the Czech Republic via grant FR-TI3/869; and by the Ministry of Education, Youth and Sports via the grant SVV-2014-260033.

REFERENCES

[1] Smith BW, Chase JG, Nokes RI, Shaw GM, Wake G. Minimal Haemodynamic System Model Including Ventricular Interaction and Valve Dynamics. Med Eng Phys 2004; 26(2): 131–139.
[2] Revie JA, Stevenson DJ, Chase JG, Hann CE, Lambermont BC, Ghuysen A, Kolh P, Shaw GM, Heldmann S, Desaive T. Validation of Subject — Specific Cardiovascular System Models from Porcine Measurements. Comput Methods Programs Biomed 2013; 109(2): 197–210.
[3] Hester RL, Brown AJ, Husband L, Iliescu R, Pruett D, Summers R, Coleman TG. HumMod: A Modeling Environment for the Simulation of Integrative Human Physiology. Front Physiol 2011; 2: 12.
[4] Kofránek J, Mateják M, Privitzer P. Complex Model of Integrated Physiological System — A Theoretical Basis For Medical Training Simulators. Mefanet Report 04 2011: 32–59.
[5] Van Meurs W. Modeling and Simulation in Biomedical Engineering: Applications in Cardiorespiratory Physiology. McGraw-Hill Professional 2011.
[6] Sá Couto CD, van Meurs WL, Goodwin JA, Andriessen P. A Model for Educational Simulation of Neonatal Cardiovascular Pathophysiology. Simul Healthc 2006; 1(1): 4–12.
[7] Fernandez de Canete J, del Saz-Orozco P, Moreno-Boza D, Duran-Venegas E. Object-Oriented Modeling and Simulation of the Closed Loop Cardiovascular System by Using SIMSCAPE. Comput Biol Med 2013; 43(4): 323–33.
[8] Fernandez de Canete J, Luque J, Barbancho J, Munoz V. Modelling of Long-Term and Short-Term Mechanisms of Arterial Pressure Control in the Cardiovascular System: An Object-Oriented Approach. Comput Biol Med 2014; 47: 104–112.
[9] Kofránek J. Modelica. In: Sborník Příspěvků MEDSOFT. 2013; 64–114.
[10] Hucka M, Finney A, Bornstein BJ, Keating SM, Shapiro BE, Matthews J, Kovitz BL, Schilstra MJ, Funahashi A, Doyle JC. Evolving a Lingua Franca and Associated Software Infrastructure for Computational Systems Biology: The Systems Biology Markup Language (SBML) Project. Syst Biol (Stevenage) 2004; 1(1): 41–53.
[11] Butterworth E, Jardine BE, Raymond GM, Neal ML, Bassingthwaighte JB. JSim, an Open-Source Modeling System for Data Analysis. F1000Research 2014; 2: 288.
[12] Cooling M, Hunter P, Crampin E. Modelling Biological Modularity with CellML. IET Syst Biol 2008; 2(2): 73–79.
[13] Kofránek J, Mateják M, Privitzer P, Tribula M. Causal or Acausal Modeling: Labour for Humans or Labour for Machines. Tech Comput Prague 2008: 1–16.
[14] Kulhánek T, Kofránek J, Mateják M. Modeling of Short-Term Mechanism of Arterial Pressure Control in the Cardiovascular System: Object Oriented and Acausal Approach. Comput Biol Med 2014; 54(1): 137–144.
[15] Mateják M. Physiology in Modelica. Mefanet J 2014; 2(1): 10–14.
[16] Mateják M, Kulhánek T, Šilar J, Privitzer P, Ježek F, Kofránek J. Physiolibrary — Modelica Library for Physiology. In: Proc. 10th Int. Model. Conf. March 10–12, 2014, Lund, Sweden. Linköping University Electronic Press 2014: 499–505.
[17] Westerhof N, Lankhaar J-W, Westerhof BE. The Arterial Windkessel. Med Biol Eng Comput 2009; 47(2): 131–141.
[18] Kofranek J, Matousek S, Rusz J, Stodulka P, Privitzer P, Matejak M, Tribula M. The Atlas of Physiology and Pathophysiology: Web-Based Multimedia Enabled Interactive Simulations. Comput Methods Programs Biomed 2011; 104(2): 143–53.
[19] Kofránek J, Mateják M, Ježek F, Privitzer P, Šilar J. Výukový webový simulator krevního oběhu. In: Sborník Příspěvků MEDSOFT 2011: 106–121.
[20] Tribula M, Ježek F, Privitzer P, Kofránek J, Kolman J. Webový výukový simulátor krevního oběhu. In: Sborník Příspěvků MEDSOFT 2013: 197–204.
[21] Ježek F, Kroček T, Mateják M, Kofránek J. Zkušenosti Z Inovace Výuky Modelování a Simulace Na FEL ČVUT. In: Sborník Příspěvků MEDSOFT 2012: 139–146.

Appendix G

Adair-Based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity*

The paper [7] published as

Marek Mateják, Tomáš Kulhánek, and Stanislav Matoušek. Adair-Based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity. *Scandinavian journal of clinical and laboratory investigation*, 2014. doi:10.3109/00365513.2014.984320

Scandinavian Journal of Clinical & Laboratory Investigation, 2015; Early Online: 1–8

informa healthcare

ORIGINAL ARTICLE

Adair-based hemoglobin equilibrium with oxygen, carbon dioxide and hydrogen ion activity

MAREK MATEJÁK, TOMÁŠ KULHÁNEK & STANISLAV MATOUŠEK

The Institute of Pathological Physiology, First Faculty of Medicine, Charles University in Prague, Czech Republic

Abstract

As has been known for over a century, oxygen binding onto hemoglobin is influenced by the activity of hydrogen ions (H^+), as well as the concentration of carbon dioxide (CO_2). As is also known, the binding of both CO_2 and H^+ on terminal valine-1 residues is competitive. One-parametric situations of these hemoglobin equilibria at specific levels of H^+, O_2 or CO_2 are also well described. However, we think interpolating or extrapolating this knowledge into an 'empirical' function of three independent variables has not yet been completely satisfactory. We present a model that integrates three orthogonal views of hemoglobin oxygenation, titration, and carbamination at different temperatures. The model is based only on chemical principles, Adair's oxygenation steps and Van't Hoff equation of temperature dependences. Our model fits the measurements of the Haldane coefficient and CO_2 hemoglobin saturation. It also fits the oxygen dissociation curve influenced by simultaneous changes in H^+, CO_2 and O_2, which makes it a strong candidate for integration into more complex models of blood acid-base with gas transport, where any combination of mentioned substances can appear.

Key Words: *Acid-base equilibrium, blood gas analysis, carboxyhemoglobin, hemoglobin A, oxyhemoglobins*

Abbreviations: ODC, hemoglobin oxygen dissociation curve; 2,3-DPG, 2,3-diphosphoglycerate; [X], molar concentration of X in $mol.m^{-3}$; aH^+, activity of hydrogen ions, where $pH = -\log_{10}(aH^+)$; αO_2, O_2 solubility in $mol.m^{-3}.Pa^{-1}$; αCO_2, CO_2 solubility in $mol.m^{-3}.Pa^{-1}$; pO_2, partial pressure of O_2 in Pa; pCO_2, partial pressure of CO_2 in Pa; Hb_u, hemoglobin alpha or beta subunit; Hb_{uD}, deoxygenated Hb_u; Hb_{uO}, oxygenated Hb_u; $Hb_uNH_3^+$, Hb_u with protonated Nterminus; $Hb_{uD}NH_3^+$, Hb_{uD} with protonated Nterminus; $Hb_{uO}NH_3^+$, Hb_{uO} with protonated Nterminus; Hb_uNH_2, Hb_u with $-NH_2$ form of Nterminus; $Hb_{uD}NH_2$, Hb_{uD} with $-NH_2$ form of Nterminus; $Hb_{uO}NH_2$, Hb_{uO} with $-NH_2$ form of Nterminus; Hb_uCOO^-, Hb_u with carboxylated Nterminus; $Hb_{uD}COO^-$, Hb_{uD} with carboxylated Nterminus; $Hb_{uO}COO^-$, Hb_{uO} with carboxylated Nterminus; Hb_uAH, Hb_u with protonated side-chains; $Hb_{uD}AH$, Hb_{uD} with protonated side-chains; $Hb_{uO}AH$, Hb_{uO} with protonated side-chains; Hb_uA^-, Hb_u with deprotonated side-chains; $Hb_{uD}A^-$, Hb_{uD} with deprotonated side-chains; $Hb_{uO}A^-$, Hb_{uO} with deprotonated side-chains; $Hb_uA^-NH_2$, selected normalized form of Hb_u with deprotonated side-chains and NH_2 form of Nterminus; $Hb_{uD}A^-NH_2$, deoxygenated form of $Hb_uA^-NH_2$; $Hb_{uO}A^-NH_2$, oxygenated form of $Hb_uA^-NH_2$; f_{nD}, fraction of $Hb_{uD}A^-NH_2$ from Hb_{uD}; f_{nO}, fraction of $Hb_{uO}A^-NH_2$ from Hb_{uO}; f_{zcD}, fraction of $Hb_{uD}NH_2$ form Hb_{uD}; f_{zcO}, fraction of $Hb_{uO}NH_2$ form Hb_{uO}; f_{hD}, fraction of $Hb_{uD}A^-$ form Hb_{uD}; f_{hO}, fraction of $Hb_{uO}A^-$ form Hb_{uO}; $\Delta H+_h$, change of valence (charge) on side-chains during deoxygenation per one Hb_u; $\Delta H+_z$, protonation of $-NH_2$ form of Nterminus during deoxygenation per one Hb_u; $\Delta H+_c$, decarboxylation of carboxylated Nterminus during deoxygenation per one Hb_u; ΔH^+, Haldane coefficient per hemoglobin subunit; Hb_t, hemoglobin tetramer without bound O_2 molecules; $(O_2)_iHb_t$, hemoglobin tetramer with the number of i bound O_2 molecules; $(O_2)_iHb_{tn}$, hemoglobin tetramer composed only of $Hb_uA^-NH_2$ subunit forms with the number of i bound O_2 molecules; sO_2, O_2 saturation of hemoglobin; sCO_{2D}, CO_2 saturation of deoxyhemoglobin; sCO_{2O}, CO_2 saturation of oxyhemoglobin; sCO_2, CO_2 saturation of hemoglobin; tCO_2, total concentration of CO_2 = free dissolved $CO_{2\,+}$ $+HCO_3^-$ $+CO_3^{2-}$ $+Hb_uCOO^-$; dTH, shift of titration curve with the change of both pO_2 and pCO_2 to zero, which equals to how many moles of strong acid must be added to one mole of Hb_u to reach the pH of the chosen standard titration curve (O_2 and CO_2 free); pH, acidity/basicity of hemoglobin solution (e.g. inside erythrocytes); pH_p, pH in plasma (e.g. acidity/basicity of blood).

Correspondence: Marek Mateják, U Nemocnice 5, Prague 2, 128 53, Czech Republic. Tel: +420 776 301 395. Fax: +420 224 965 916. E-mail: marek.matejak@lf1.cuni.cz

(Received 20 June 2014; accepted 2 November 2014)

ISSN 0036-5513 print/ISSN 1502-7686 online © 2015 Informa Healthcare
DOI: 10.3109/00365513.2014.984320

Introduction

Human hemoglobin A is one of the most extensively studied protein macromolecules. The composition and 3D conformation of both α and β chains is known [1,2] and the binding of O_2, H^+ and 2,3-DPG has been described [2,3]. Since the 1930s it has generally been believed that CO_2 binding to Hb occurs by carbamination of the amino-terminus, i.e. forming a carboxylate compound [4–6]. This hemoglobin carbamination was verified by Morrow et al. [7], who used nuclear magnetic resonance of $^{13}CO_2$ to find its exact binding sites at valine-1. The shift of titration between oxygenated and deoxygenated forms is also well known. Called the Bohr or Haldane effect [8–10], it is caused by the same valine-1 side and by more than 10 other acid-base residues [2,11].

The most common descriptions of the hemoglobin oxygen dissociation curve (ODC) are the allosteric models [12,13], the model based on Hill equation [14,15], and Adair's four-step model [16]. Some of the ODC models [15,17,18] also include the effects of varying pH and CO_2 concentration. However, these models operate only with interpolation or extrapolation of ODC from normal pH or normal pCO_2 and do not take into account the chemical dependences between titration [8] and carbamination [19]. They fail when both pH and CO_2 are not at normal values, especially in the alkaline pH range.

Based on these findings, we propose a hemoglobin binding equilibrium model that starts with a description of hemoglobin-oxygen dissociation using a slightly modified version of Adair's approach [16]. We continue by describing the relationship between CO_2 and H^+ as competitive inhibition in the amino group of terminal valine-1 residue on each chain (suffixes *z* and *c*), which is in accordance with the known facts [4,6,20]. Finally, we lump all other acid-base side chains residues in a hemoglobin subunit into one Bohr proton-binding site of the side chain residues. These are denoted by the suffix *h* in the article.

Methods

The model is built in Mathematica 9.0 (Wolfram, Champaign, IL, USA) and also in Dymola FD01 2014 (Dassault Systemes, Paris, France) as an example of chemical package in open-source Modelica library Physiolibrary 2.2.0 (http://www.physiolibrary.org) [21,22] according to the model structure defined in the following section. The model contains only physiological parameters such as gas solubilities and dissociation coefficients of defined reactions. The unknown parameters are fitted to the data of Siggaard-Andersen [23], Bauer and Schröder [24], Severinghaus [18], Matthew et al. [1] and Reeves [25] using Mathematica function FindFit and also using parameter estimation method suggested by Kulhanek et al. [26].

Free dissolved concentrations of $[O_2]$ and $[CO_2]$ in red blood cells are calculated from gas partial pressures using Henry's law ($[O_2] = \alpha O_2 * pO_2$ and $[CO_2] = \alpha CO_2 * pCO_2$) with solubility coefficients $\alpha O_2 = 1.005*10^5$ mol.m^{-3}.Pa^{-1} measured at 38°C by Sendroy et al. [27], and $\alpha CO_2 = 2.3*10^4$ mol.m^{-3}.Pa^{-1} at 37°C, as proposed by Maas et al. [28]. These solubilities are slightly different from solubilities in pure water or in plasma because of the effects of salts and proteins inside erythrocyte [27].

Model structure

The model is built upon several simplifying assumptions. Firstly, each of four hemoglobin tetramer subunits is treated as identical, even though (slight) differences are known to exist between α and β subunits. Secondly, CO_2 or H^+ binding is supposed not to affect the CO_2 or H^+ affinities in the other three subunits. Thus, the interaction between the subunits is modeled purely through the varying affinities of oxygen in each oxygenation step. The third simplifying assumption has already been mentioned; it is lumping all Bohr proton binding sites of subunit into two (first for the valine-1 amino-terminus and second for the side chains residues), a simplification first suggested by Antonini [29].

Model structure of hemoglobin subunit

The three reactions that participate in the Bohr and Haldane effect of each subunit are as follows:

Reaction z: $Hb_uNH_3{}^+ \leftrightarrow Hb_uNH_2 + H^+$
Reaction c: $Hb_uNH_2 + CO_2 \leftrightarrow Hb_uCOO^- + H^+$
Reaction h: $Hb_uAH \leftrightarrow Hb_uA^- + H^+$

where the reactions z and c are competitive on the valine-1 amino-terminus, and the reaction h is independent of z or c.

The chemical equilibrium equations of these three reactions are Equations 1–3, where K_x is the equilibrium dissociation coefficients of the reaction *x* (i.e. *z*, *c* or *h*).

$$\text{Reaction z: } K_z = \frac{[Hb_uNH_2]*aH^+}{[Hb_uNH_3^+]} \qquad (1)$$

$$\text{Reaction c: } K_c = \frac{[Hb_uCOO^-]*aH^+}{[CO_2][Hb_uNH_2]} \qquad (2)$$

$$\text{Reaction h: } K_h = \frac{[Hb_uA^-]*aH^+}{[Hb_uAH]} \qquad (3)$$

These dissociation coefficients are different between oxy and deoxy subunits, which are distinguished by the subscripts O and D in the following text. They can be also written in their logarithmic form, where $\mathbf{pK_x}$ means $-\log_{10}(K_x)$. Thus, for instance, pK_{zO} denotes the equilibrium coefficient of the reaction z for the oxy form of the hemoglobin subunit. Similar notation is used for describing the activity of hydrogen ions (acidity), where $pH = -\log_{10}(aH^+)$.

Using Equations 1–3 it is possible to express fractions of chosen species for deoxy and oxy subunits. We label these fractions as follows: Hb_uNH_2 fractions are called $\mathbf{f_{zcD}}$ ($\mathbf{f_{zcO}}$), Hb_uA^- fractions $\mathbf{f_{hD}}$ ($\mathbf{f_{hO}}$), $Hb_uA^-NH_2$ fractions $\mathbf{fn_D}$ ($\mathbf{fn_O}$) and Hb_uCOO^- fractions $\mathbf{sCO_{2D}}$ ($\mathbf{sCO_{2O}}$) as in Equation 4–7. The selection of form $Hb_uA^-NH_2$ from the orthogonal division into Hb_uNH_2 and Hb_uA^- is also illustrated in Figure 1A, B.

$$f_{zcD} = \frac{[Hb_{uD}NH_2]}{[Hb_{uD}]} = \frac{1}{1+10^{pKzD-pH}+[CO_2]10^{pH-pKcD}} \quad (4)$$

$$f_{hD} = \frac{[Hb_{uD}A^-]}{[Hb_{uD}]} = \frac{1}{10^{pKhD-pH}+1} \quad (5)$$

$$f_{nD} = \frac{[Hb_{uD}A^-NH_2]}{[Hb_{uD}]} = f_{zcD} \times f_{hD} \quad (6)$$

$$sCO_{2D} = \frac{[Hb_{uD}COO^-]}{[Hb_{uD}]} = 10^{pH-pKcD} f_{zcD}[CO_2] \quad (7)$$

We define a titration shift as the amount of acid that must be added to achieve the same pH after full deoxygenation of the hemoglobin subunit. This change of subunit charge during deoxygenation (by the Bohr protons) is also called Haldane coefficient ΔH^+. The coefficient can be divided into contributions of the previously mentioned reactions ΔH^+_h, ΔH^+_z and ΔH^+_c, as is algebraically expressed by Equations 8–11.

$$\Delta H_h^+ = -\frac{[Hb_{uD}A^-]-[Hb_{uO}A^-]}{[Hb_u]} = -(f_{hD}-f_{ho}) \quad (8)$$

$$\Delta H_z^+ = \frac{[Hb_{uD}NH_3^+]-[Hb_{uo}NH_3^+]}{[Hb_u]} = \left(\frac{10^{pKzD}}{10^{pH}} f_{zcD} - \frac{10^{pKzO}}{10^{pH}} f_{zcO}\right) \quad (9)$$

$$\Delta H_c^+ = -\frac{[Hb_{uD}COO^-]-[Hb_{uO}COO^-]}{[Hb_u]} = -(sCO_{2D}-sCO_{2O}) \quad (10)$$

$$\Delta H^+ = \Delta H_z^+ + \Delta H_c^+ + \Delta H_h^+ \quad (11)$$

Model structure of hemoglobin tetramer

Chemical speciation of the hemoglobin tetramer molecule can be considered at various levels of detail; the one chosen as appropriate in our approach is indicated in Figure 1C. The possible forms include different combinations of oxygenated and deoxygenated

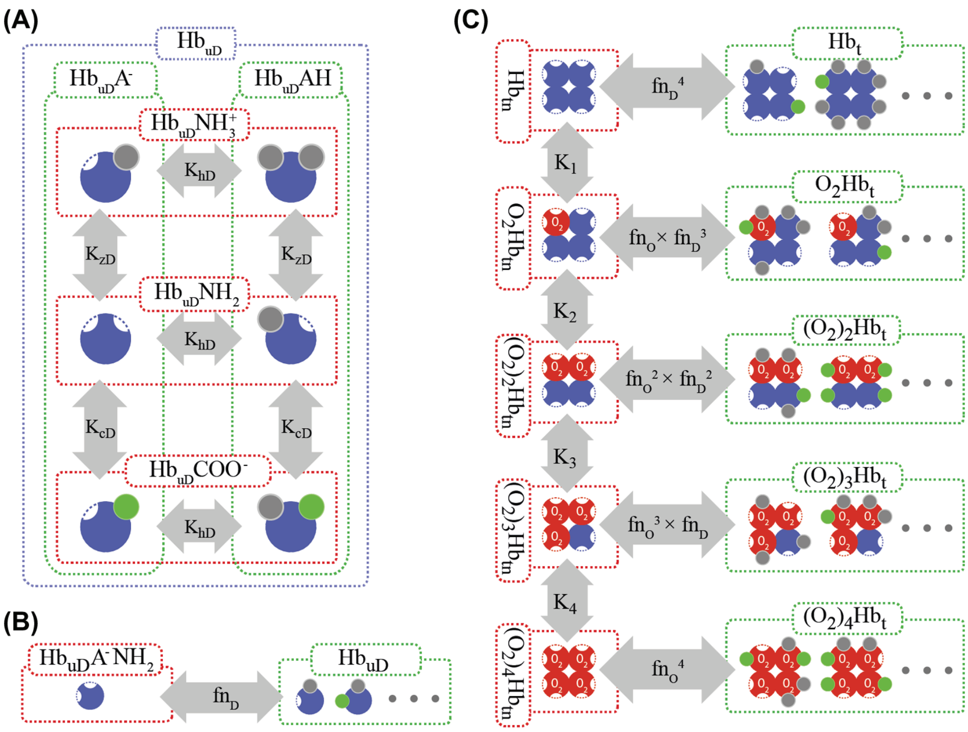


Figure 1. Schema of hemoglobin calculation. (A) Possible forms of deoxyhemoglobin subunit (the blue circles). The gray circles represent hydrogen ions, and the green circles represent carbon dioxide. The arrows with dissociation coefficients represent the reactions z, c and h. (B) Schema of chemical speciation of deoxyhemoglobin subunit. (C) Schema of chemical speciation and Adair's oxygenation steps of hemoglobin tetramer. This Figure is reproduced in color in the online version of *The Scandinavian Journal of Clinical & Laboratory Investigation.*

hemes, protonated and deprotonated oxygen-linked acid groups, and free or carboxylated amino endings of each chain. Yet because of the law of detailed balance in equilibrium [30], it is not necessary to calculate all reactions between these forms. As a result, only four oxygenation reactions need be selected for the calculation. This is done in accordance with Figure 1C, by selecting a tetramer form $(O_2)_iHb_{tn}$ composed only of four subunits in the form $Hb_uA^-NH_2$ (H^+ and CO_2 free), and modeling oxygen binding to these forms with Adair-type coefficients of the following reactions:

$$O_2Hb_{tn} \leftrightarrow Hb_{tn} + O_2$$
$$(O_2)_2Hb_{tn} \leftrightarrow O_2Hb_{tn} + O_2$$
$$(O_2)_3Hb_{tn} \leftrightarrow (O_2)_2Hb_{tn} + O_2$$
$$(O_2)_4Hb_{tn} \leftrightarrow (O_2)_3Hb_{tn} + O_2$$

where dissociation coefficients are defined by Equation 12, which are also represented by vertical arrows at Figure 1C.

$$K_i = \frac{[(O_2)_{i-1}Hb_{tn}]\cdot[O_2]}{[(O_2)_iHb_{tn}]} \quad (12)$$

For the next calculation, all forms can be expressed as a fraction of the deoxy-tetramer form $(O_2)_0Hb_{tn}$, as shown in Equation 13.

$$[(O_2)_iHb_{tn}] = \frac{[(O_2)_0Hb_{tn}]\cdot[O_2]^i}{\prod_{j=1}^{i}K_i} \quad (13)$$

Let us move the attention from specific forms of Hb_{tn} to the description of the equilibrium within the whole group of Hb_t. Looking at Figure 1C, one can see that for each oxygenation step we can calculate the equilibrium in each horizontal line using Equation 14.

$$[(O_2)_iHb_t]\cdot fn_D^{4-i}\cdot fn_O^{i} = [(O_2)_iHb_{tn}] \quad (14)$$

The CO_2- and pH-dependent oxygen saturation equation in the Adair style (Equation 15) is algebraically derived from Equation 13–14, where $a_i = \frac{1}{\left(\prod_{j=1}^{i}K_j\right)}$ $x = (fn_D(pH,[CO_2])/fn_O(pH,[CO_2]))*[O_2]$.

$$sO_2 = \frac{a_1x + 2a_2x^2 + 3a_3x^3 + 4a_4x^4}{4 + 4a_1x + 4a_2x^2 + 4a_3x^3 + 4a_4x^4} \quad (15)$$

Hemoglobin saturation with CO_2 (sCO_2) is calculated separately in oxygenated and deoxygenated subunits forms (sCO_{2O} and sCO_{2D}) using Equation 16.

$$sCO_2 = sO_2 \cdot sCO_{2O} + (1 - sO_2)\cdot sCO_{2D} \quad (16)$$

Finally, the shift of titration after deoxygenation and decarbamination of the hemoglobin subunit can be expressed by Equation 17.

$$dTH = sO_2 \cdot \Delta H^+ + sCO_{2D} + \frac{sCO_{2D}}{1+10^{pH-pKzD}} \quad (17)$$

Temperature dependences

The temperature dependences are integrated using the Van't Hoff equation (Equation 18), where $R = 8.314\ J.K^{-1}.mol^{-1}$ is gas constant, ΔH^θ is the standard enthalpy change, i.e. an amount of heat consumed by a reaction changing one mole of substrates to products, and K_2 and K_1 are Henry's coefficients of solution or the dissociation coefficient of the chemical reaction at temperature T_2 and T_1 (expressed in Kelvin).

$$\ln\left(\frac{K_2}{K_1}\right) = \frac{-\Delta H^\theta}{R}\left(\frac{1}{T_2} - \frac{1}{T_1}\right) \quad (18)$$

Results

The Adair's coefficients are fitted to the collection of ODC measurements by Severinghaus [18] at $pCO_2 = 0$ Pa and $pH_p = 7.4$, see Figure 2A and Table I. The dissociation coefficients of carboxylation are determined in close agreement with Bauer and Schröder using their data [24]; the resulting fit can be seen in Figure 2B, the coefficients are in Table II. The lumped acid dissociation coefficients for the side-chains pK_{hD} and pK_{hO} are estimated by optimization of Siggaard-Andersen's data [23] at DPG/Hbt = 0.84, pH = 6.5–8.0 and 37°C; the resulting fit can be seen in Figure 2C, and the coefficients complete Table II.

We conclude that the model also describes the ODC shifts by comparing with Naerea et al.'s [31] oxygen saturation measurements at different plasma pH and CO_2 levels, see Figure 2D. The recalculation of data from plasma pH_p to intracellular erythrocyte pH uses the equation $pH = 7.2464 + 0.796(pH_p - 7.4)$, as presented by Siggaard-Andersen and Salling [32].

The gas solubility as Henry's coefficient at different temperatures can be recalculated using the enthalpy of gas solution (− 14 $kJ.mol^{-1}$ for O_2, and − 20 $kJ.mol^{-1}$ for CO_2 [33]). If we assume that the examined hemoglobin of Matthew et al. [1] is at compatible conditions, but at a temperature of 30°C with $pK_{zD} = 7.53$, $pK_{zO} = 7.28$, $pK_{cD} = 4.77$ and $pK_{cO} = 5.20$ as plotted in Figure 2E, then the enthalpies of these reactions are − 51, 8, 59 and − 41 kJ.mol^{-1}.

Atha and Ackers measured the heat of hemoglobin oxygenation under conditions independent of carbon dioxide and Bohr protons and not including the oxygen heat of solution, coming to the value of 59 $kJ.mol^{-1}$ [34]. Using this value for each Adair oxygenation step, one can optimize other enthalpies to fit the Reeves data [25] measured at 19–43°C (Figure 2F). Resulting enthalpies are 59 $kJ.mol^{-1}$ for the deoxy and 127 $kJ.mol^{-1}$ for the oxy version of reaction h.

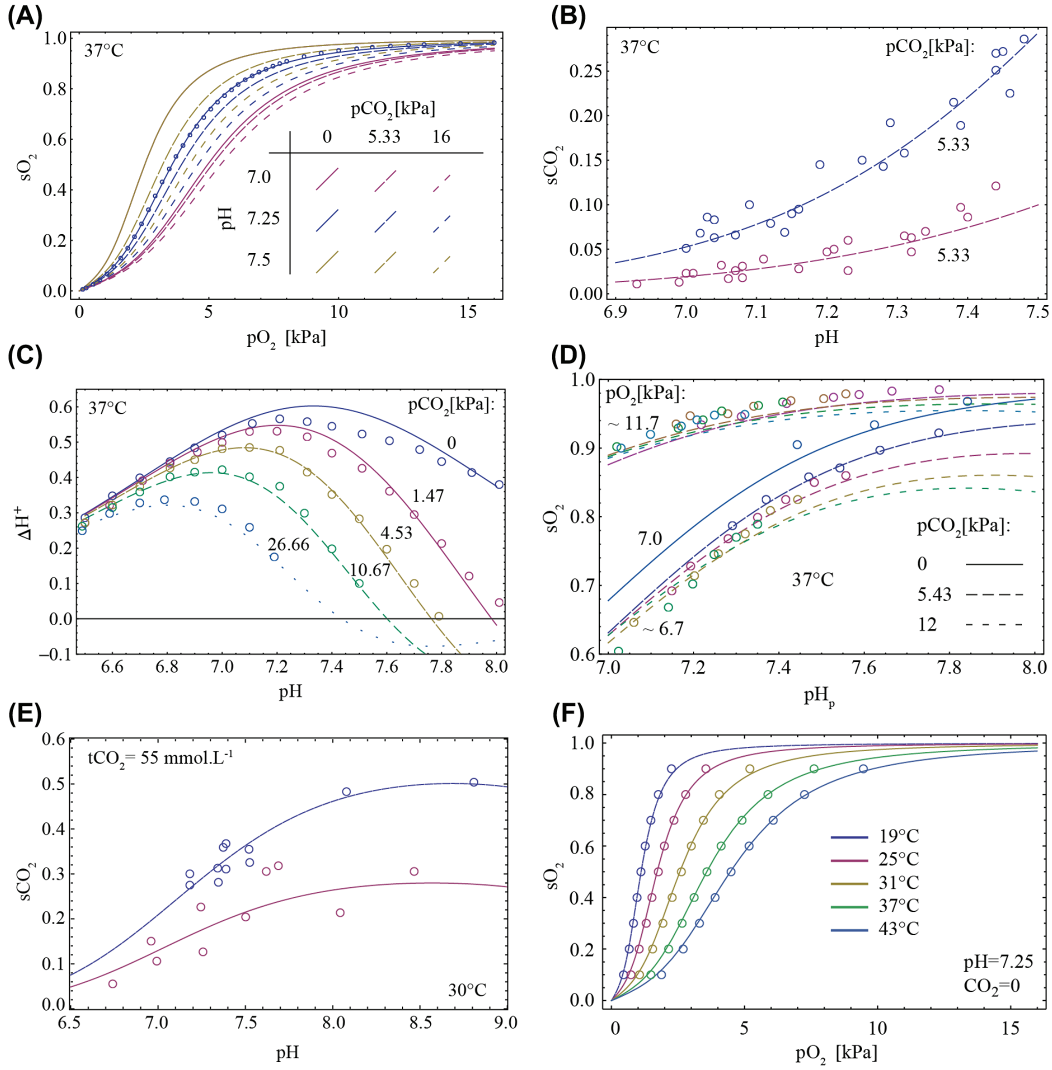


Figure 2. Model curves and measured data points. (A) Points are from Severinghaus's oxygen dissociation curve collection [18] at $pCO_2 = 0$ Pa, erythrocyte intracellular pH = 7.25 and temperature 37°C. Lines are sO_2 defined by Equation 15. (B) Carboxylation of hemoglobin measured and estimated by Bauer and Schröder [24]. The lines (Equation 7) are sCO_{2D} and sCO_{2O} at $pCO_2 = 5.33$ kPa. (C) Bohr protons released during oxygenation of one hemoglobin subunit. Dots are data measured by Siggaard-Andersen [23] in erythrolysate at 37°C, $DPG/Hb_t = 0.84$. Lines are calculated from Equation 11 using coefficients from Table II. From top to bottom data are plotted for different $pCO_2 = 0$ kPa, 1.47 kPa, 4.53 kPa, 10.67 kPa and 26.66 kPa. (D) Naeraa et al.'s [31] oxygen saturation data comparison. The two groups of lines (Equation 15) are determined mainly by pO_2 around 6.7 kPa and 11.7 kPa. The small nuances in each group are caused by different pCO_2 changed from 0–12 kPa. (E) Matthew et al.'s [1] carbamination data measured at 30°C with constant content of all carbonates 55 $mmol.L^{-1}$. Fitted dissociation constants, which determine the lines, are used to estimate enthalpies of their reactions to be consistent with Bauer and Schröder [24]. (F) Reeves [25] oxygen saturation data at different temperatures. Enthalpies of reaction h and specific Adair oxygenation step enthalpy is estimated (lines) to fit the data (circles). This Figure is reproduced in color in the online version of *The Scandinavian Journal of Clinical & Laboratory Investigation.*

Discussion

Having a precise quantitative description of hemoglobin behavior is a crucial aspect in describing the behavior of whole blood, which has numerous uses in medicine today. It is important in areas such as blood gas analysis [15,34], the building of complex models [35], simulation for teaching purposes [36], or rebreathing-based methods for cardiac output estimation, which represent a more specific use [37]. For instance, the precision of the latter methods is crucially dependent on the exact calculation of the total amount of carbon dioxide in blood for various

Table I. Estimated form-specific Adair's coefficients [$mol.m^{-3}$] at 37°C.

K_1	K_2	K_3	K_4
0.0121	0.0117	0.0871	0.000386

Table II. Estimated acid dissociation coefficients at 37°C.

Reaction z	Reaction c	Reaction h
$pK_{zD} = 7.73$	$pK_{cD} = 7.54$	$pK_{hD} = 7.52$
$pK_{zO} = 7.25$	$pK_{cO} = 8.35$	$pK_{hO} = 6.89$

(abnormal) patient conditions, as has been recently pointed out [38].

This model offers a precise description of various phenomena that take place with hemoglobin, based on relatively simple starting points, such as competitive binding of H+ and CO_2 at valine-1 amino terminus or the law of detailed balance [30]. Even with a simple structure, it offers a remarkably good fit to the data of hemoglobin oxygenation, titration, and carbamination at different temperatures, as shown in Figure 2A–E. These can be calculated with any combination of oxygen, carbon dioxide and hydrogen ions defined as open system (lungs), where the partial pressures are equilibrated, and also in a closed system (tissues), where mass conservation laws and the total amount of substances take place, as was also modeled by Rees and Andreassen [39], among others.

The results of our model (Figure 2A–E) show strong nonlinear dependences between variables, which is in agreement with the known data [1,23–25,40]. Looking at Figure A, one can compare sets of ODCs, where each color represents a different pH. Various curves of each color represent ODCs for various levels of pCO_2 for a given pH. As can be seen, the effect of pCO_2 on the ODC is stronger at high (alkaline) pH, which is in agreement with the data [23]. Similarly, one can compare the curves within the sets of solid or dashed lines, where each set represents dissociation curves for various values of pH at a given level of pCO_2. The variation between the curves of each line type represents the Bohr effect for the given level of CO_2, this effect can also be appreciated from data of Figure 2C, which shows the average amount of released H^+ upon oxygenation of one hemoglobin subunit.

The model uses enthalpies to calculate temperature dependences of hemoglobin behavior (Figure 2F, E), which allows examination of the heat transfers during single chemical processes. For instance, binding of aqueous oxygen onto hemoglobin produces 30–40 kJ/mol of heat [41–44], and the same amount of heat is consumed by deoxygenation process in metabolically active tissues, thus helping to cool them down. When the hemoglobin model is used in the standard conditions of a large-scale HumMod model [35], the resulting heat transfer due to the exothermy of the hemoglobin oxygen reaction is 4–7% of the total heat produced by muscle, which is in agreement with experimental results [41,43,44]. It is interesting to note that this heat transfer occurs without any increase in blood temperature.

The integration of chemical processes in a macromolecule requires a precise view into their underlying principles. Some physiologists use the elementary chemical equations [39], but do not implement the principle of detailed balance [30]. Other physiologists make empirically-based equations with a raw linear gradient approximation of possible combinations of model values [15]. We feel that it is almost impossible to see the problem as this type of black-box function with more than two inputs. Instead, it is better to have an integrated model of oxygenation, titration and carbamination.

Today, allosteric hemoglobin oxygenation models do exist that seem more in agreement with the structural knowledge of hemoglobin [12,45–47]. These models take into account two or more structurally different forms: relaxed and tensed. However, these models have so far been limited to hemoglobin oxygenation only. Our Adair-based model can explain not only oxygen and carbon dioxide saturation, but also their cooperation with acid-base buffering properties of hemoglobin. All three of these connected phenomena fit to measured data in physiological ranges.

As any work, the presented model has its limitations. First of all, it does not include the effects of changes in Hb tetramer conformations. Also the model could be extended with dependences on electrolytes such as chloride, 2,3-bisphosphoglycerate or other organic phosphates and their binding reactions, as many research studies show these interactions [48–51]. The next extension of this model could be performed by the integration of an intracellular red cell environment to calculate with phosphate acid-base buffers, and finally the membrane changes with blood plasma, where the chloride shift reaches a Gibbs-Donnan equilibrium and establishes chloride, bicarbonate and hydrogen ion activity ratios. Having an integrated model of blood gases and acid-base is crucial if we want the precise computational algorithms of the current state of a patient. These calculations could be used, for example, inside the next generation of medical devices to estimate not only blood properties, but also the connected properties of circulation [37] or metabolic functions [39].

In this article, we present a hemoglobin model that integrates O_2, CO_2 and H^+ binding. The model is not just empirical, but is based on sound theoretical principles, such as the competitive binding of CO_2 and H^+ on the valine-1 NH_2 terminus, the Bohr and Haldane effect [9,52] and on the principle of detailed balance. The principle of detailed balance is used for the first time in the Adair type of model. The advantages of this approach include explicit formulation of mass and energy conservation principles: The model accumulates substances and heat inside hemoglobin forms, making it very useful for integration in higher-scale dynamic models.

Acknowledgements

In addition, the authors want to thank doc. MUDr. Jiří Kofránek, CSc, and Prof. MUDr. Emanuel Nečas, DrSc, for critical review and Austin Schaefer for English language editing.

This work was supported by grants from the Ministry of Industry and Trade of the Czech Republic: MPO TIP TI3/869 – Virtual patient for medical education; and by The Ministry of Education, Youth and Sports via the grant SVV-2014-260033.

Declaration of interest: The authors report no conflict of interest. The authors alone are responsible for the content and writing of the paper.

References

[1] Matthew JB, Morrow J, Wittebort RJ, Gurd F. Quantitative determination of carbamino adducts of alpha and beta chains in human adult hemoglobin in presence and absence of carbon monoxide and 2, 3-diphosphoglycerate. J Biol Chem 1977;252:2234–44.
[2] Perutz M. Species adaptation in a protein molecule. Mol Biol Evol 1983;1:1–28.
[3] Perutz M, Kilmartin J, Nishikura K, Fogg J, Butler P, Rollema H. Identification of residues contributing to the Bohr effect of human haemoglobin. J Mol Biol 1980;138: 649–68.
[4] Rossi-Bernardi L, Roughton F. The specific influence of carbon dioxide and carbamate compounds on the buffer power and Bohr effects in human haemoglobin solutions. J Physiol 1967;189:1–29.
[5] Forster R, Constantine H, Craw MR, Rotman H, Klocke R. Reaction of CO2 with human hemoglobin solution. J Biol Chem 1968;243:3317–26.
[6] Stadie WC, O'Brien H. The carbamate equilibrium I. The equilibrium of amino acids, carbon dioxide, and carbamates in aqueous solution; with a note on the Ferguson-Roughton carbamate method. J Biol Chem 1936;112:723–58.
[7] Morrow J, Matthew J, Wittebort R, Gurd F. Carbon 13 resonances of 13CO2 carbamino adducts of alpha and beta chains in human adult hemoglobin. J Biol Chem 1976;251:477–84.
[8] Bohr C, Hasselbalch K, Krogh A. Concerning a biologically important relationship–the influence of the carbon dioxide content of blood on its oxygen binding. Skand Arch Physiolog 1904;16:402.
[9] Siggaard-Andersen O, Garby L. The Bohr effect and the Haldane effect. Scand J Clin Lab Invest 1973;31:1–8.
[10] Tyuma I. The Bohr effect and the Haldane effect in human hemoglobin. Jpn J Physiol 1984;34:205.
[11] Matthew J, Hanania G, Gurd F. Electrostatic effects in hemoglobin: hydrogen ion equilibria in human deoxy-and oxyhemoglobin A. Biochemistry 1979;18:1919.
[12] Monod J, Wyman J, Changeux J-P. On the nature of allosteric transitions: a plausible model. J Mol Biol 1965;12:88–118.
[13] Eaton WA, Henry ER, Hofrichter J, Bettati S, Viappiani C, Mozzarelli A. Evolution of allosteric models for hemoglobin. IUBMB Life 2007;59:586–99.
[14] Hill AV. The combinations of haemoglobin with oxygen and with carbon monoxide. I. Biochem J 1913;7:471.
[15] Siggaard-Andersen O, Siggaard-Andersen M. The oxygen status algorithm: a computer program for calculating and displaying pH and blood gas data. Scand J Clin Lab Invest 1990;50:29–45.
[16] Adair GS. The hemoglobin system VI. The oxygen dissociation curve of hemoglobin. J Biol Chem 1925;63:529–45.
[17] Dash RK, Bassingthwaighte JB. Erratum to: Blood HbO_2 and $HbCO_2$ dissociation curves at varied O_2, CO_2, pH, 2, 3-DPG and temperature levels. Ann Biomed Eng 2010; 38:1683–701.
[18] Severinghaus JW. Simple, accurate equations for human blood O_2 dissociation computations. J Appl Physiol 1979; 46:599–602.
[19] Christiansen J, Douglas C, Haldane J. The absorption and dissociation of carbon dioxide by human blood. J Physiol 1914;48:244–71.
[20] Perrella M, Bresciani D, Rossi-Bernardi L. The binding of CO2 to human hemoglobin. J Biol Chem 1975;250: 5413–8.
[21] Mateják M, Kulhánek T, Šilar J, Privitzer P, Ježek F, Kofránek J. Physiolibrary-Modelica library for Physiology. 10th International Modelica Conference; 2014 March 12; Lund, Sweden: Linköping University Electronic Press; 2014. pp. 499–505.
[22] Mateják M. Physiology in Modelica. Mefanet J 2014; 2:10–4.
[23] Siggaard-Andersen O. Oxygen-linked hydrogen ion binding of human hemoglobin. effects of carbon dioxide and 2, 3-diphosphoglycerate I. Studies on erythrolysate. Scand J Clin Lab Invest 1971;27:351–60.
[24] Bauer C, Schröder E. Carbamino compounds of haemoglobin in human adult and foetal blood. J Physiol 1972;227: 457–71.
[25] Reeves RB. The effect of temperature on the oxygen equilibrium curve of human blood. Respir Physiol 1980; 42:317–28.
[26] Kulhanek T, Matejak M, Silar J, Kofranek J. Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud. Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on 1–4 June 2014; pp. 712–5.
[27] Sendroy J, Dillon RT, Van Slyke DD. Studies of gas and electrolyte equilibria in blood XIX. The solubility and physical state of uncombined oxygen in blood. J Biol Chem 1934;105:597–632.
[28] Maas A, Rispens P, Siggaard-Andersen O, Zijlstra W. On the reliability of the Henderson-Hasselbalch equation in routine clinical acid-base chemistry. Ann Clin Biochem 1984;21:26.
[29] Antonini E, Wyman J, Brunori M, Fronticelli C, Bucci E, Rossi-Fanelli A. Studies on the relations between molecular and functional properties of hemoglobin V. The influence of temperature on the Bohr effect in human and in horse hemoglobin. J Biol Chem 1965;240:1096–103.
[30] Sauro HM. Enzyme kinetics for systems biology. Future Skill Software (Ambrosius Publishing); 2011. pp. 20–1.
[31] Naeraa N, Petersen ES, Boye E. The influence of simultaneous, independent changes in pH and carbon dioxide tension on the in vitro oxygen tension-saturation relationship of human blood. Scand J Clin Lab Invest 1963;15:141–51.
[32] Siggaard-Andersen O, Salling N. Oxygen-linked hydrogen ion binding of human hemoglobin. Effects of carbon dioxide and 2, 3-diphosphoglycerate. II. Studies on whole blood. Scand J Clin Lab Invest 1971;27:361–6.
[33] Lide DR. CRC handbook of chemistry and physics. Boca Raton, FL: CRC Press; 2004.
[34] Atha DH, Ackers GK. Calorimetric determination of the heat of oxygenation of human hemoglobin as a function of pH and the extent of reaction. Biochemistry 1974;13:2376–2382.
[35] Hester RL, Brown AJ, Husband L, Iliescu R, Pruett D, Summers R, Coleman TG. HumMod: a modeling environment for the simulation of integrative human physiology. Front Physiol 2011;2.
[36] Kofranek J, Matousek S, Rusz J, Stodulka P, Privitzer P, Matejak M, Tribula M. The Atlas of Physiology and Pathophysiology: Web-based multimedia enabled interactive simulations. Comput Methods Programs Biomed 2011;104: 143–53.
[37] Jarvis SS, Levine BD, Prisk GK, Shykoff BE, Elliott AR, Rosow E, Blomqvist CG, Pawelczyk JA. Simultaneous determination of the accuracy and precision of closed-circuit cardiac output rebreathing techniques. J Appl Physiol 2007; 103:867–74.

[38] Sun X-G, Hansen JE, Ting H, Chuang M-L, Stringer WW, Adame D, Wasserman K. Comparison of exercise cardiac output by the Fick Principle using oxygen and carbon dioxide. CHEST J 2000;118:631–40.
[39] Rees SE, Andreassen S. Mathematical models of oxygen and carbon dioxide storage and transport: the acid-base chemistry of blood. Crit Rev Biomed Eng 2005;33.
[40] Imai K, Yonetani T. PH dependence of the Adair constants of human hemoglobin. Nonuniform contribution of successive oxygen bindings to the alkaline Bohr effect. J Biol Chem 1975;250:2227–31.
[41] Adair G, Cordero N, Shen T. The effect of temperature on the equilibrium of carbon dioxide and blood and the heat of ionization of haemoglobin. J Physiol 1929;67:288–98.
[42] Chipperfield J, Rossi-Bernardi L, Roughton F. Direct calorimetric studies on the heats of ionization of oxygenated and deoxygenated hemoglobin. J Biol Chem 1967; 242:777–83.
[43] Weber RE, Campbell KL. Temperature dependence of haemoglobin-oxygen affinity in heterothermic vertebrates: mechanisms and biological significance. Acta Physiolog 2011; 202:549–62.
[44] Weber RE, Fago A, Campbell KL. Enthalpic partitioning of the reduced temperature sensitivity of O2 binding in bovine hemoglobin. Comparat Biochem Physiol A: Molec Integrat Physiol 2014;176:20–5.
[45] Szabo A, Karplus M. A mathematical model for structure-function relations in hemoglobin. J Mol Biol 1972; 72:163–97.
[46] Brunori M, Coletta M, Di Cera E. A cooperative model for ligand binding to biological macromolecules as applied to oxygen carriers. Biophys Chem 1986;23:215–22.
[47] Henry ER, Bettati S, Hofrichter J, Eaton WA. A tertiary two-state allosteric model for hemoglobin. Biophys Chem 2002;98:149–64.
[48] de Bruin SH, Rollema HS, Janssen LH, van Os GA. The interaction of chloride ions with human hemoglobin. Biochem Biophys Res Commun 1974;58:210–5.
[49] Rollema HS, de Bruin SH, Janssen L, Van Os G. The effect of potassium chloride on the Bohr effect of human hemoglobin. J Biol Chem 1975;250:1333–9.
[50] Benesch R, Benesch RE, Yu CI. Reciprocal binding of oxygen and diphosphoglycerate by human hemoglobin. Proc Natl Acad Sci USA 1968;59:526.
[51] Bunn HF, Briehl RW. The interaction of 2, 3-diphosphoglycerate with various human hemoglobins. J Clin Invest 1970; 49:1088.
[52] Siggaard-Andersen O, Salling N, Nörgaard-Pedersen B, Rørth M. Oxygen-linked hydrogen ion binding of human hemoglobin. effects of carbon dioxide and 2, 3-diphosphoglycerate: III. Comparison of the Bohr effect and the Haldane effect. Scand J Clin Lab Invest 1972;29:185–93.

Bibliography

[1] Tomáš Kulhánek and Milan Šárek. Processing of medical images in virtual distributed environment. In *Proceedings of the 2009 Euro American Conference on Telematics and Information Systems: New Opportunities to increase Digital Citizenship*, page 10. ACM, 2009. doi:10.1145/1551722.1551732. (cited on p. 1, 2, 4, 37, 47)

[2] Tomáš Kulhánek, Marek Frič, and Milan Šárek. Remote Analysis of Human Voice–Lossless Sound Recording Redirection. *Analysis of Biomedical Signals and Images. Proceedings of 20th International EURASIP Conference (BIOSIGNAL)*, pages 394–397, 2010. URL: http://bs2010.biosignal.cz/papers/1092.pdf. (cited on p. 1, 2, 4, 37, 51)

[3] Tomáš Kulhánek. Infrastructure for data storage and computation in biomedical research. *European Journal for Biomedical Informatics (EJBI)*, 6(1):55–58, 2010. URL: http://www.ejbi.org/img/ejbi/ejbi2010-1.pdf. (cited on p. 1, 2, 4, 37, 57)

[4] Tomáš Kulhánek, Marek Mateják, Jan Šilar, and Jiří Kofránek. Parameter estimation of complex mathematical models of human physiology using remote simulation distributed in scientific cloud. In *Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on*, pages 712–715, 2014. doi:10.1109/BHI.2014.6864463. (cited on p. 1, 3, 4, 38, 63)

[5] Tomáš Kulhánek, Jiří Kofránek, and Marek Mateják. Modeling of short-term mechanism of arterial pressure control in the cardiovascular system: Object oriented and acausal approach. *Computers in Biology and Medicine*, 54(1):137–144, 2014. doi:10.1016/j.compbiomed.2014.08.025. (cited on p. 1, 2, 4, 30, 69)

[6] Tomáš Kulhánek, Martin Tribula, Jiří Kofránek, and Marek Mateják. Simple models of the cardiovascular system for educational and research purposes. *MEFANET Journal*, 2(2):56–63, 2014. (cited on p. 1, 2, 4, 30, 79)

[7] Marek Mateják, Tomáš Kulhánek, and Stanislav Matoušek. Adair-Based Hemoglobin Equilibrium with Oxygen, Carbon Dioxide and Hydrogen Ion Activity. *Scandinavian journal of clinical and laboratory investigation*, 2014.

doi:10.3109/00365513.2014.984320. (cited on p. 1, 3, 4, 38, 39, 40, 42, 89)

[8] Tomáš Kulhánek, J Šilar, Marek Mateják, Pavol Privitzer, Jiří Kofranek, and Martin Tribula. Distributed computation and parameter estimation in identification of physiological systems. *VPH conference*, 2010. (cited on p. 1, 3)

[9] Tomáš Kulhánek, Filip Ježek, Marek Mateják, Jan Šilar, Pavol Privitzer, Martin Tribula, and Jiří Kofránek. RESTful web service to build loosely coupled web based simulation of human physiology. *Transactions of Japanese Society for Medical and Biological Engineering*, 51(Supplement):R–32, 2013. doi:10.11239/jsmbe.51.R-32. (cited on p. 1, 3)

[10] J Kofránek, M Mateják, P Privitzer, M Tribula, T Kulhánek, J Šilar, and R Pecinovský. HumMod-Golem Edition: large scale model of integrative physiology for virtual patient simulators. In *Proceedings of World Congress in Computer Science 2013 (WORLDCOMP'13), International Conference on Modeling, Simulation and Visualisation Methods (MSV'13)*, pages 182–188, 2013. (cited on p. 1, 2, 30)

[11] Marek Mateják, Tomáš Kulhánek, Jan Šilar, Pavol Privitzer, Filip Ježek, and Jiří Kofránek. Physiolibrary - Modelica library for Physiology. In *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*, pages 499–505. Linköping University Electronic Press, 2014. doi:10.3384/ecp14096499. (cited on p. 1, 2)

[12] Tomáš Kulhánek and Milan Šárek. Virtualizace a integrace v gridovém pacs systému (cze) virtualization and integration in grid pacs system (eng). In *Mefanet 2008*. MSD Brno, 2008. URL: http://www.mefanet.cz/res/file/mefanet2008/prispevky/25_kulhanek.pdf. (cited on p. 1, 2)

[13] Milan Šárek and Tomáš Kulhánek. Nové směry medicínských aplikací sdružení CESNET. In *sborník příspěvků MEDSOFT*, pages 145–148, 2009. URL: http://www.creativeconnections.cz/medsoft/2009/Medsoft_2009_%C5%A0%C3%A1rek_Milan.pdf. (cited on p. 1, 2)

[14] Tomáš Kulhánek. Virtual Distributed Environment for Exchange of Medical Images. In D Kuželová, editor, *Proceeding of PhD. Conference 2009 was published as: Doktorandské dny '09*, pages 62–64. Institute of Computer Science/MatfyzPress, 2009. URL: http://www.cs.cas.cz/hakl/doktorandsky-den/files/2009/sbornik-dd-2009.pdf. (cited on p. 1, 2)

[15] Tomáš Kulhánek, Marek Frič, and Milan Šárek. Vzdálený přístup k virtuálním výukovým a výzkumným aplikacím - podpora foniatrick-

ých vyšetření. In Daniel Schwarz, Jaroslav Majerník, Ladislav Dušek, Stanislav Štípek, and Vladimír Mihál, editors, *Mefanet 2009*. MSD Brno, 2009. URL: http://www.mefanet.cz/res/file/mefanet2009/prispevky/kulhanek_full.pdf. (cited on p. 1)

[16] Tomáš Kulhánek, Marek Frič, and Milan Šárek. Vzdálená analýza lidského hlasu - bezeztrátové nahrávání zvuku přes ip sítě. In *sborník příspěvků MEDSOFT*, pages 96–101, 2010. (cited on p. 1, 2)

[17] Tomáš Kulhánek, Jan Šilar, Jiří Kofránek, Marek Mateják, Pavol Privitzer, and Martin Tribula. Od výukového modelu k identifikaci fyziologického systému. In Daniel Schwarz, Martin Komenda, Jaroslav Majerník, Stanislav Štípek, Vladimír Mihál, and Ladislav Dušek, editors, *Mefanet 2010*. MSD Brno, 2010. URL: http://www.mefanet.cz/res/file/mefanet2010/prispevky/kulhanek-full.pdf. (cited on p. 1)

[18] Tomáš Kulhánek. From Educational Models Towards Identification of Physiological Systems. *Mefanet Report*, 04:69–72, 2011. URL: http://www.mefanet.cz/res/file/reporty/mefanet-report-2011.pdf. (cited on p. 1, 3)

[19] Tomáš Kulhánek. Infrastructure for Data Storage and Computation in Biomedical Research. In D Kuželová, editor, *Proceeding of PhD. Conference 2011 was published as: Doktorandské dny '11*, pages 90–94. Institute of Computer Science/MatfyzPress, 2011. URL: http://www.cs.cas.cz/hakl/doktorandsky-den/files/2011/sbornik-dd-2011.pdf. (cited on p. 1, 2)

[20] Tomáš Kulhánek, Marek Frič, and Jaroslav Hrb. Vzdálená analýza lidského hlasu v reálném čase. In *sborník příspěvků MEDSOFT*, pages 180–184, 2012. URL: http://www.creativeconnections.cz/medsoft/2012/Medsoft_Kulhánek_Tomáš.pdf. (cited on p. 1, 2)

[21] Tomáš Kulhánek, Marek Mateják, Jan Šilar, Pavol Privitzer, Martin Tribula, Filip Ježek, and Jiří Kofránek. HYBRIDNÍ ARCHITEKTURA PRO WEBOVÉ SIMULÁTORY. In *sborník příspěvků MEDSOFT*, pages 115–121, 2013. URL: http://www.creativeconnections.cz/medsoft/2013/Medsoft_2013_Kulhanek.pdf. (cited on p. 1, 3)

[22] Tomáš Kulhánek, Marek Mateják, Jan Šilar, and Jiří Kofránek. IDENTIFIKACE FYZIOLOGICKÝCH SYSTÉMŮ. In *sborník příspěvků MEDSOFT*, pages 148–153, 2014. URL: http://www.creativeconnections.cz/medsoft/2014/Medsoft_2014_Kulhanek.pdf. (cited on p. 1, 3)

[23] Tomáš Kulhánek and Marek Frič. Remote analysis of human voice - environment for voice training and ORL medicine. In *EGI Technical Forum, Prague*, Prague, 2012. (cited on p. 1, 38)

[24] Jiří Kofránek, Pavol Privitzer, Martin Tribula, Jan Šilar, Tomáš Kulhánek, Marek Mateják, and Filip Ježek. Virtual patient simulator, CZ Utility model. Application number: 2014-30329. Issue number: 27613, 2014. URL: http://isdv.upv.cz/portal/pls/portal/portlets.pts.det?xprim=10082946. (cited on p. 1, 3)

[25] Victor Maojo and Casimir A. Kulikowski. Bioinformatics and Medical Informatics: Collaborations on the Road to Genomic Medicine?, 2003. doi:10.1197/jamia.M1305. (cited on p. 5)

[26] F. Martin-Sanchez, I. Iakovidis, S. Nø rager, V. Maojo, P. De Groen, et al. Synergy between medical informatics and bioinformatics: Facilitating genomic medicine for future health care. *Journal of Biomedical Informatics*, 37:30–42, 2004. doi:10.1016/j.jbi.2003.09.003. (cited on p. 5)

[27] Jiří Kofránek, Rudolf Pecinovský, and Petr Novák. Kopenograms – Graphical Language for Structured Algorithms. In *International Conference Foundations of Computer Science*, pages 90–96, 2012. (cited on p. 6)

[28] Rudolf Bayer and E. M. McCreight. Organization and maintenance of large ordered indexes. *Acta Informatica*, 1(3):173–189, 1972. doi:10.1007/BF00288683. (cited on p. 6)

[29] Rudolf Bayer. Symmetric binary B-Trees: Data structure and maintenance algorithms. *Acta Informatica*, 1(4):290–306, 1972. doi:10.1007/BF00289509. (cited on p. 6)

[30] Alan Cobham. The intrinsic computational difficulty of functions. In *Proc. 1964 Congress for Logic,Methodology and Philosophy of Science.*, pages 24–30, 1964. (cited on p. 6)

[31] Stephen A. Cook. An overview of computational complexity. *Communications of the ACM*, 26(6):400–408, June 1983. doi:10.1145/358141.358144. (cited on p. 6)

[32] M R Garey and D S Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences). *Computers and Intractability*, page 340, 1979. (cited on p. 6, 8)

[33] Richard E Korf. Depth-First Iterative-Deepening: An Optimal Admissible Tree Search. *Artificial Intelligence*, 27:97–109, 1985. (cited on p. 6)

[34] J Pearl and R E Korf. Search Techniques. *Annual Review of Computer Science*, 2:451–467, 1987. doi:10.1146/annurev.cs.02.060187.002315. (cited on p. 6)

[35] Stephen A. Cook. The Complexity of Theorem-Proving Procedures. In *STOC '71 Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971. doi:10.1145/800157.805047. (cited on p. 7)

[36] Richard M. Karp. *Reducibility among combinatorial problems*. Springer US, 1972. doi:10.1007/978-1-4684-2001-2_9. (cited on p. 7)

[37] Planck Collaboration, P. a. R. Ade, N. Aghanim, M. I. R. Alves, C. Armitage-Caplan, et al. Planck 2013 results. I. Overview of products and scientific results. March 2013. doi:10.1051/0004-6361/201321529. (cited on p. 7)

[38] CH Papadimitriou. *Computational complexity*. Addison-WesleyPublishing Company, 1995. (cited on p. 8)

[39] Michael Sipser. *Introduction to the Theory of Computation*. 2012. (cited on p. 8)

[40] Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of AFIPS Conference*, pages 483–485, 1967. (cited on p. 10, 40)

[41] John L. Gustafson. Reevaluating Amdahl's law. *Communications of the ACM*, 31(5):532–533, May 1988. doi:10.1145/42411.42415. (cited on p. 11)

[42] David R. Butenhof. *Programming with POSIX Threads*. 1997. (cited on p. 12)

[43] B Chapman, G Jost, and Ruud Van Der Pas. *Using OpenMP*, volume 10. MIT press, 2008. (cited on p. 13)

[44] Peter S. Pacheco. *Parallel Programming with MPI*. Morgan Kaufmann, 1997. (cited on p. 13)

[45] Javier Diaz, Camelia Muñoz Caro, and Alfonso Niño. A survey of parallel programming models and tools in the multi and many-core era. *IEEE Transactions on Parallel and Distributed Systems*, 23(8):1369–1386, 2012. doi:10.1109/TPDS.2011.308. (cited on p. 13)

[46] Cleve Moler. Matrix computation on distributed memory multiprocessors. *Hypercube multiprocessors*, pages 181–195, 1986. (cited on p. 13)

[47] Ian Foster. *Designing and Building Parallel Programs*. Addison-Wesley Publishing Company, 1995. (cited on p. 13)

[48] Erick Cantu-Paz. *Designing efficient and accurate parallel genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 1999. (cited on p. 13)

[49] Raymond Greenlaw, H James Hoover, and Walter L Ruzzo. *Limits to Parallel Computation*. Oxford university press, Oxford, 1995. (cited on p. 13)

[50] PH Madden. Dispelling the myths of parallel computing. *IEEE Design & Test of Computers*, (April):58–64, 2012. doi:10.1109/MDT.2012.2230391. (cited on p. 13)

[51] David Culer, Jaswinder Pal Singh, and Anoop Gupta. *Parallel Computer Architecture*. Morgan Kaufmann, 1998. (cited on p. 13)

[52] Thomas Rauber and Gudula Rünger. *Parallel programming: For multicore and cluster systems*. Springer, 2013. (cited on p. 13)

[53] Andrew S Tanenbaum and Maarten Van Steen. *Distributed Systems: Principles and Paradigms, 2/E*. 2007. (cited on p. 14)

[54] Thomas Erl. *SOA Principles of Service Design*. Prentice Hall, 2008. (cited on p. 16)

[55] Roy Thomas Fielding. Chapter 5: Representational State Transfer (REST). *Architectural Styles and the Design of Network-based Software Architectures, Dissertation*, 2000. (cited on p. 16)

[56] Steve Vinoski. REST eye for the SOA guy. *IEEE Internet Computing*, 11:82–84, 2007. doi:10.1109/MIC.2007.22. (cited on p. 16)

[57] Martin Fowler. *Patterns of Enterprise Application Architecture*. Pearson Education, Inc., 2003. (cited on p. 16)

[58] Gregor Hohpe and Bobby Woolf. *Enterprise Integration Patterns*. Pearson Education, Inc., 2002. (cited on p. 16)

[59] I Foster and C Kesselman. *The grid 2: blueprint for a new computing infrastructure*. Morgan Kaufmann, 2004. (cited on p. 16)

[60] Ian Foster, Carl Kesselman, and Steven Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International journal of high performance computing applications*, 15(3):200–222, 2001. (cited on p. 16)

[61] A Chervenak, I Foster, C Kesselman, C Salisbury, and S Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of network and computer applications*, 23(3):187–200, 2000. (cited on p. 17)

[62] Ian Foster, Carl Kesselman, Jeffrey M Nick, and Steven Tuecke. The Physiology of the Grid: An Open Grid Service Architecture for Distributed Systems Integration. *Grid computing: making the global infrastructure a reality*, pages 217–249, 2003. (cited on p. 17)

[63] Ian Foster, Carl Kesselman, Gene Tsudik, and Steven Tuecke. A security architecture for computational grids. In *Proceedings of the 5th ACM conference on Computer and communications security - CCS '98*, pages 83–92, New York, New York, USA, November 1998. ACM Press. doi:10.1145/288090.288111. (cited on p. 17)

[64] Václav Novák and Josef Vojtěch. Deployment of a DWDM System with CLA Optical Amplifiers in the CESNET2 Network. *Networking Studies*, page 27, 2007. (cited on p. 17)

[65] I. Bird, B. Jones, and K.F. Kee. The Organization and Management of Grid Infrastructures. *Computer*, 42(1):36–46, January 2009. doi:10.1109/MC.2009.28. (cited on p. 18)

[66] Dagmar Adamova. Computing framework for the LHC: current status and challenges of the High Luminosity LHC future. In *52 International Winter Meeting on Nuclear Physics, Bormio,Italy*, number January, 2014. URL: http://pos.sissa.it/archive/conferences/212/021/Bormio2014_021.pdf. (cited on p. 18)

[67] M. Litzkow and M. Livny. Experience with the Condor distributed batch system. *IEEE Workshop on Experimental Distributed Systems*, 1990. doi:10.1109/EDS.1990.138057. (cited on p. 18)

[68] E Laure, S Fisher, A Frohner, C Grandi, P Kunszt, et al. Programming the Grid with gLite. *Computational Methods in Science and Technology*, 12:33–45, 2006. (cited on p. 18)

[69] I. Foster and C. Kesselman. Globus: a Metacomputing Infrastructure Toolkit, 1997. doi:10.1177/109434209701100205. (cited on p. 18)

[70] M Riedel, E Laure, L Field, J P Navarro, J Casey, et al. Interoperation of world-wide production e-Science infrastructures. In *Concurrency and Computation: Practice and Experience 21.8*, number March, pages 961–990. 2009. doi:10.1002/cpe. (cited on p. 18)

[71] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. SETI@home: an experiment in public-resource computing. *Communications of the ACM*, 45(11):56–61, November 2002. doi:10.1145/581571.581573. (cited on p. 18)

[72] DP Anderson. Boinc: A system for public-resource computing and storage. In *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*. IEEE, 2004. doi:10.1109/GRID.2004.14. (cited on p. 19, 35)

[73] Zoltan Balaton, Gabor Gombas, Peter Kacsuk, Adam Kornafeld, Jozsef Kovacs, et al. SZTAKI desktop grid: a modular and scalable way of building large computing grids. In *Parallel and Distributed Processing Symposium*. IEEE, 2007. doi:10.1109/IPDPS.2007.370665. (cited on p. 19)

[74] Peter Kacsuk, Jozsef Kovacs, Zoltan Farkas, Attila Csaba Marosi, Gabor Gombas, and Zoltan Balaton. SZTAKI Desktop Grid (SZDG): A Flexible and Scalable Desktop Grid System. *Journal of Grid Computing*, 7(4):439–461, September 2009. doi:10.1007/s10723-009-9139-y. (cited on p. 19)

[75] Gilles Fedak, Franck Cappello, and Universitc Paris Sud. XtremWeb : A Generic Global Computing System. In *Proceedings of First IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 582–587, 2001. doi:10.1109/CCGRID.2001.923246. (cited on p. 19)

[76] Werner Herr, D I Kaltchev, F Schmidt, and E Mcintosh. Large Scale Beam-beam Simulations for the CERN LHC using Distributed Computing. Technical Report June 2006, 2006. URL: http://cds.cern.ch/record/972345. (cited on p. 19)

[77] N Hø imyr, J Blomer, P Buncic, M Giovannozzi, A Gonzalez, et al. BOINC service for volunteer cloud computing. *Journal of Physics: Conference Series*, 396(3):032057, December 2012. doi:10.1088/1742-6596/396/3/032057. (cited on p. 19)

[78] P. Kacsuk, Z. Farkas, and G. Fedak. Towards Making BOINC and EGEE Interoperable. *2008 IEEE Fourth International Conference on eScience*, pages 478–484, December 2008. doi:10.1109/eScience.2008.111. (cited on p. 19)

[79] Etienne Urbah, Peter Kacsuk, Zoltan Farkas, Gilles Fedak, Gabor Kecskemeti, et al. EDGeS: Bridging EGEE to BOINC and XtremWeb. *Journal of Grid Computing*, 7(3):335–354, September 2009. doi:10.1007/s10723-009-9137-0. (cited on p. 19)

[80] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, et al. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, 37:164, 2003. doi:10.1145/1165389.945462. (cited on p. 19)

[81] Lamia Youseff, Rich Wolski, Brent Gorda, and Chandra Krintz. Paravirtualization for HPC Systems. *Frontiers of High Performance Computing and*

Networking â€“ ISPA 2006 Workshops, pages 474–486, 2006. doi:10.1007/11942634_49. (cited on p. 19)

[82] Miroslav Ruda, Zdeněk Šustr, Ji\vrí Sitera, David Antoš, Lukáš Hejtmánek, Petr Holub, and Miloš Mulač. Virtual Clusters as a New Service of MetaCentrum, the Czech NGI. In *Cracow Grid Workshop'09*, pages 66–71, 2009. (cited on p. 19)

[83] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. Cloud computing and grid computing 360-degree compared. *Grid Computing Environments ...*, 2008. doi:10.1109/GCE.2008.4738445. (cited on p. 19, 21)

[84] Peter Mell and Timothy Grance. The NIST definition of cloud computing. 2011. URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. (cited on p. 20)

[85] Ashiq Anjum, Richard Hill, Richard McClatchey, Nik Bessis, and Andrew Branson. Glueing grids and clouds together: a service-oriented approach. *International Journal of Web and Grid Services*, 8(3):248–265, 2012. doi:10.1504/IJWGS.2012.049169. (cited on p. 20)

[86] Dejan Milojicic, Ignacio M Llorente, and Ruben S Montero. OpenNebula: A Cloud Management Tool. *IEEE Internet Computing*, 15(2):11–14, 2011. doi:10.1109/MIC.2011.44. (cited on p. 20)

[87] R Kumar, N Gupta, S Charu, K Jain, and SK Jangir. Open Source Solution for Cloud Computing Platform Using OpenStack. 3(5):89–98, 2014. URL: http://www.ijcsmc.com/docs/papers/May2014/V3I5201427.pdf. (cited on p. 20)

[88] Sixto Ortiz. The problem with cloud-computing standardization. *Computer*, 44(July):13–16, 2011. doi:10.1109/MC.2011.220. (cited on p. 20)

[89] Georg Hager and Gerhard Wellein. *Introduction to High Performance Computing for Scientists and Engineers*. 2010. doi:10.1201/EBK1439811924. (cited on p. 20)

[90] John Levesque and Gene Wagenbreth. *High Performance Computing: Programming and Applications*. CRC Press, 2010. (cited on p. 20)

[91] Ioan Raicu, Ian T. Foster, and Yong Zhaoyo. Many-task computing for grids and supercomputers. In *2008 Workshop on Many-Task Computing on Grids and Supercomputers, MTAGS 2008*, 2008. doi:10.1109/MTAGS.2008.4777912. (cited on p. 21)

[92] Ioan Raicu. *Many-task computing: bridging the gap between high-throughput computing and high-performance computing.* PhD thesis, 2009. (cited on p. 21)

[93] Ioan Raicu, Ian Foster, Mike Wilde, Zhao Zhang, Kamil Iskra, et al. Middleware support for many-task computing. *Cluster Computing*, 13:291–314, 2010. doi:10.1007/s10586-010-0132-9. (cited on p. 21)

[94] Alexandru Iosup, Simon Ostermann, Nezih Yigitbasi, Radu Prodan, Thomas Fahringer, and D Epema. Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. *IEEE Transactions on Parallel and Distributed Systems*, 22(6):931–945, 2010. (cited on p. 21)

[95] James Pasley. How BPEL and SOA are changing web services development. *IEEE Internet Computing*, 9(June):60–67, 2005. doi:10.1109/MIC.2005.56. (cited on p. 21)

[96] Jia Yu and Rajkumar Buyya. A taxonomy of scientific workflow systems for grid computing. *ACM SIGMOD Record*, 34(3):44, 2005. doi:10.1145/1084805.1084814. (cited on p. 21)

[97] Yong Zhao, Ioan Raicu, and Ian Foster. Scientific Workflow Systems for 21st Century e-Science, New Bottle or New Wine? *IEEE Congress on Services - Part I, 2008.*, pages 467 – 471, 2008. arXiv:0808.3545. (cited on p. 21)

[98] V. Curcin and M. Ghanem. Scientific workflow systems - can one size fit all? *2008 Cairo International Biomedical Engineering Conference*, 2008. doi:10.1109/CIBEC.2008.4786077. (cited on p. 21)

[99] Peter Kacsuk, Tamas Kiss, and Gergely Sipos. Solving the grid interoperability problem by P-GRADE portal at workflow level. *Future Generation Computer Systems*, 24:744–751, 2008. doi:10.1016/j.future.2008.02.008. (cited on p. 21)

[100] Peter Kacsuk. P-GRADE portal family for grid infrastructures. *Concurrency Computation Practice and Experience*, 23:235–245, 2011. doi:10.1002/cpe.1654. (cited on p. 21, 35)

[101] Kassian Plankensteiner, Radu Prodan, Matthias Janetschek, Thomas Fahringer, Johan Montagnat, et al. Fine-Grain Interoperability of Scientific Workflows in Distributed Computing Infrastructures. *Journal of Grid Computing*, 11:429–455, 2013. doi:10.1007/s10723-013-9261-8. (cited on p. 21)

[102] Nancy Wilkins-Diehr. Special Issue: Science Gateways—Common Community Interfaces to Grid Resources. *Concurrency and Computation: Practice*

and Experience, 19(6):743–749, April 2007. URL: http://doi.wiley.com/10.1002/cpe.1098, doi:10.1002/cpe.1098. (cited on p. 21)

[103] Marlon Pierce, Suresh Marru, Borries Demeler, Raminderjeet Singh, and Gary Gorbet. The Apache Airavata Application Programming Interface: Overview and Evaluation with the UltraScan Science Gateway. In *2014 9th Gateway Computing Environments Workshop*, pages 25–29. IEEE, November 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7021845, doi:10.1109/GCE.2014.15. (cited on p. 21)

[104] Shahbaz Memon, Morris Riedel, Florian Janetzko, Borries Demeler, Gary Gorbet, et al. Advancements of the UltraScan scientific gateway for open standards-based cyberinfrastructures. *Concurrency and Computation: Practice and Experience*, 26(13):2280–2291, September 2014. doi:10.1002/cpe.3251. (cited on p. 21)

[105] Peter Kacsuk, Zoltan Farkas, Miklos Kozlovszky, Gabor Hermann, Akos Balasko, Krisztian Karoczkai, and Istvan Marton. WS-PGRADE/gUSE Generic DCI Gateway Framework for a Large Variety of User Communities. *Journal of Grid Computing*, 10:601–630, 2012. doi:10.1007/s10723-012-9240-5. (cited on p. 21)

[106] Issac N. Bankman, editor. *Handbook of Medical Imaging; Processing and Analysis*. 2000. (cited on p. 24)

[107] K. Slavicek, O. Dostal, M. Javornik, and M. Drdla. MEDIMED - Regional Centre for Medicine Image Data Processing. *2010 Third International Conference on Knowledge Discovery and Data Mining*, pages 310–313, January 2010. doi:10.1109/WKDD.2010.133. (cited on p. 25, 27)

[108] Peeter Ross, Ruth Sepper, and Hanna Pohjonen. Cross-border teleradiology-experience from two international teleradiology projects. *European journal of radiology*, 73(1):20–5, January 2010. doi:10.1016/j.ejrad.2009.10.016. (cited on p. 25)

[109] Vanessa Saliba, Helena Legido-Quigley, Riina Hallik, Ain Aaviksoo, Josip Car, and Martin McKee. Telemedicine across borders: a systematic review of factors that hinder or support implementation. *International journal of medical informatics*, 81(12):793–809, December 2012. doi:10.1016/j.ijmedinf.2012.08.003. (cited on p. 25)

[110] S. G. Erberich, M. Bhandekar, M. D. Nelson, a. Chervenak, and C. Kesselman. DICOM grid interface service for clinical and research PACS: A globus

toolkit web service for medical data grids. *International Journal of Computer Assisted Radiology and Surgery*, 1(ii):100–102, 2006. doi:10.1007/s11548-006-0013-0. (cited on p. 25, 26)

[111] Stephan G Erberich, Jonathan C Silverstein, Ann Chervenak, Robert Schuler, Marvin D Nelson, and Carl Kesselman. Globus MEDICUS - federation of DICOM medical imaging devices into healthcare Grids. *Studies in health technology and informatics*, 126:269–278, 2007. (cited on p. 25, 26)

[112] H Duque, J Montagnat, J Pierson, L Brunie, and I E Magnin. DM 2 : A Distributed Medical Data Manager for Grids. In *Biogrid'03 workshop*, pages 138–147, Tokyo, Japan. (cited on p. 25)

[113] Johan Montagnat, Ákos Frohner, Daniel Jouvenot, Christophe Pera, Peter Kunszt, et al. A Secure Grid Medical Data Manager Interfaced to the gLite Middleware. *Journal of Grid Computing*, 6(1):45–59, July 2007. doi:10.1007/s10723-007-9088-2. (cited on p. 25)

[114] Dagmar Krefting, Julian Bart, Kamen Beronov, Olga Dzhimova, Jürgen Falkner, et al. MediGRID: Towards a user friendly secured grid infrastructure. *Future Generation Computer Systems*, 25(3):326–336, 2009. doi:10.1016/j.future.2008.05.005. (cited on p. 25)

[115] Dagmar Krefting, Michal Vossberg, Andreas Hoheisel, and Thomas Tolxdorff. Simplified implementation of medical image processing algorithms into a grid using a workflow management system. *Future Generation Computer Systems*, 26(4):681–684, 2010. doi:10.1016/j.future.2009.07.004. (cited on p. 25)

[116] K Skaburskas, F Estrella, J Shade, D Manset, J Revillard, et al. Health-e-Child: a grid platform for european paediatrics. *Journal of Physics: Conference Series*, 119:082011, 2008. doi:10.1088/1742-6596/119/8/082011. (cited on p. 25)

[117] Siegfried Benkner, Antonio Arbona, Guntram Berti, Alessandro Chiarini, Robert Dunlop, et al. @neurIST: Infrastructure for advanced disease management through integration of heterogeneous data, computing, and complex processing services. *IEEE Transactions on Information Technology in Biomedicine*, 14(6):1365–1377, 2010. doi:10.1109/TITB.2010.2049268. (cited on p. 25)

[118] Martin Kuba, Ondrej Krajícek, Petr Lesný, Jan Vejvalka, and Tomás Holecek. Grid empowered sharing of medical expertise. *Studies in health technology and informatics*, 120:273–282, 2006. (cited on p. 25)

[119] Martin Kuba, Daniel Kouřil, and Michal Procházka. Sealed grid with downloadable services. In Thierry Priol and Marco Vanneschi, editors, *Towards Next Generation Grids*, pages 83–91. Springer US, 2007. doi:10.1007/978-0-387-72498-0_8. (cited on p. 26)

[120] I Foster. Globus toolkit version 4:Software for service-oriented systems. *Journal of Computer Science and Technology*, 21(4):513–520, 2006. (cited on p. 26)

[121] William Allcock, John Bresnahan, Rajkumar Kettimuthu, Michael Link, Catalin Dumitrescu, Ioan Raicu, and Ian Foster. The Globus striped GridFTP framework and server. *Proceedings of the ACM/IEEE 2005 Supercomputing Conference, SC'05*, 2005(c):1–11, 2005. doi:10.1109/SC.2005.72. (cited on p. 26)

[122] JW Cooley and JW Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, pages 297–301, 1965. URL: http://www.jstor.org/stable/2003354. (cited on p. 27)

[123] M. Frigo and S.G. Johnson. The Design and Implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1386650, doi:10.1109/JPROC.2004.840301. (cited on p. 27)

[124] Wendy DeLeo LeBorgne and Barbara D. Weinrich. Phonetogram changes for trained singers over a nine-month period of vocal training. *Journal of Voice*, 16(1):37–43, 2002. doi:10.1016/S0892-1997(02)00070-X. (cited on p. 27)

[125] FLORIS L Wuyts, LOUIS Heylen, FONS Mertens, M D Caju, RAOUL Rooman, P H de Heyning, and Marc De Bodt. Effects of age, sex, and disorder on voice range profile characteristics of 230 children. *ANNALS OF OTOLOGY RHINOLOGY AND LARYNGOLOGY*, 112(6):540–548, 2003. (cited on p. 27)

[126] Jan G. Švec and Harm K. Schutte. Videokymography: High-speed line scanning of vocal fold vibration. *Journal of Voice*, 10(2):201–205, 1996. doi:10.1016/S0892-1997(96)80047-6. (cited on p. 27)

[127] Jan G Svec, Frantisek Sram, and Harm K Schutte. Videokymography in voice disorders: what to look for? *The Annals of otology, rhinology, and laryngology*, 116(3):172–80, March 2007. URL: http://europepmc.org/abstract/MED/17419520. (cited on p. 27)

[128] Marek Frič. Parametrizovaný fonetogram obecných řečových a hlasových projevů - ParVRP, 2007. URL: http://zvuk.hamu.cz/vyzkum/dokumenty/TL12x.pdf. (cited on p. 27, 28)

[129] Marek Frič, Tomáš Kulhánek, and Jaroslav Hrb. Systém pro vzdálené přístupnou analýzu hlasu RealVoiceLab, 2012. URL: http://zvuk.hamu.cz/vyzkum/dokumenty/TL46x.pdf. (cited on p. 27, 28)

[130] P Kohl, E J Crampin, T A Quinn, and D Noble. Systems biology: an approach. *Clinical pharmacology and therapeutics*, 88:25–33, 2010. doi:10.1038/clpt.2010.92. (cited on p. 29)

[131] P. J. Hunter and M. Viceconti. The VPH-Physiome Project: Standards and Tools for Multiscale Modeling in Clinical Applications. *IEEE Reviews in Biomedical Engineering*, 2:40–53, 2009. doi:10.1109/RBME.2009.2036204. (cited on p. 29)

[132] A C Guyton, T G Coleman, and H J Granger. Circulation: overall regulation. *Annual review of physiology*, 34:13–46, 1972. doi:10.1146/annurev.ph.34.030172.000305. (cited on p. 29)

[133] Robert L Hester, Radu Iliescu, Richard Summers, and Thomas G Coleman. Systems biology and integrative physiological modelling. *The Journal of physiology*, 589(Pt 5):1053–60, March 2011. doi:10.1113/jphysiol.2010.201558. (cited on p. 29)

[134] Robert L Hester, Alison J Brown, Leland Husband, Radu Iliescu, Drew Pruett, Richard Summers, and Thomas G Coleman. HumMod: a modeling environment for the simulation of integrative human physiology. *Frontiers in physiology*, 2(April):12, January 2011. doi:10.3389/fphys.2011.00012. (cited on p. 29)

[135] Erik Butterworth, Bartholomew E. Jardine, Gary M. Raymond, Maxwell L. Neal, and James B. Bassingthwaighte. JSim, an open-source modeling system for data analysis. *F1000Research*, 2, May 2014. doi:10.12688/f1000research.2-288.v2. (cited on p. 30)

[136] P J Hunter. The IUPS Physiome Project: a framework for computational physiology. *Progress in biophysics and molecular biology*, 85(2-3):551–69, January 2004. doi:10.1016/j.pbiomolbio.2004.02.006. (cited on p. 30)

[137] Tommy Yu, Catherine M Lloyd, David P Nickerson, Michael T Cooling, Andrew K Miller, et al. The Physiome Model Repository 2. *Bioinformatics (Oxford, England)*, 27(5):743–744, 2011. doi:10.1093/bioinformatics/btq723. (cited on p. 30)

[138] Michael Hucka, ABBJ Finney, Benjamin J Bornstein, Sarah M Keating, Bruce E Shapiro, et al. Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup Language (SBML) project. *Systems biology*, 1(1):41–53, 2004. (cited on p. 30)

[139] Nicolas Le Novère, Benjamin Bornstein, Alexander Broicher, Mélanie Courtot, Marco Donizelli, et al. BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic acids research*, 34:D689–D691, 2006. doi:10.1093/nar/gkj092. (cited on p. 30)

[140] J Kofránek and J Rusz. Restoration of Guyton´s diagram for regulation of the circulation as a basis for quantitative physiological model development. *Physiological research / Academia Scientiarum Bohemoslovaca*, 59(6):897–908, January 2010. (cited on p. 30)

[141] J Kofranek, M Matejak, and P Privitzer. Hummod-large scale physiological models in modelica. In *Proceedings 8th Modelica Conference, Dresden, Germany*, pages 713–724, 2011. (cited on p. 30, 38, 39, 40)

[142] Javier Fernandez de Canete, P del Saz-Orozco, D Moreno-Boza, and E Duran-Venegas. Object-oriented modeling and simulation of the closed loop cardiovascular system by using SIMSCAPE. *Computers in biology and medicine*, 43(4):323–33, May 2013. doi:10.1016/j.compbiomed.2013.01.007. (cited on p. 30)

[143] J Fernandez de Canete, J Luque, J Barbancho, and V Munoz. Modelling of long-term and short-term mechanisms of arterial pressure control in the cardiovascular system: An object-oriented approach. *Computers in biology and medicine*, 47:104–112, April 2014. doi:10.1016/j.compbiomed.2014.01.006. (cited on p. 30)

[144] Peter Fritzson and Peter Bunus. Modelica-a general object-oriented language for continuous and discrete-event system modeling and simulation. In *Simulation Symposium, 2002. Proceedings. 35th Annual*, pages 365–380. IEEE, 2002. (cited on p. 31)

[145] Michael Tiller. Modelica by Example, 2014. URL: http://book.xogeny.com. (cited on p. 31)

[146] Michael C K Khoo. *Physiological control systems*. IEEE press, 2000. (cited on p. 31)

[147] M. Hofmann. On the Complexity of Parameter Calibration in Simulation Models. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 2(4):217–226, October 2005. doi:10.1177/154851290500200405. (cited on p. 31)

[148] Carmen G Moles, Pedro Mendes, and Julio R Banga. Parameter Estimation in Biochemical Pathways: A Comparison of Global Optimization Methods.

Genome Research, pages 2467–2474, 2003. doi:10.1101/gr.1262503. (cited on p. 31)

[149] William A Pruett, Leland D Husband, Graham Husband, Muhammad Dakhlalla, Kyle Bellamy, Thomas G Coleman, and Robert L Hester. A Population Model of Integrative Cardiovascular Physiology. *PloS one*, 8(9):e74329, 2013. (cited on p. 31)

[150] Shuji Takahashi, Kenta Sakawa, Yoichi Shiraishi, Hiroshi Miyashita, and Science Technology. Modeling, simulation and parameter estimation of the cardiovascular system by using Model Based approach. pages 493–500, 2013. (cited on p. 31)

[151] Mohamed a. Abbass, Emad ElSamahy, and Ahmed Genedy. An optimized cardiovascular model: A pattern search approach. *Proceedings - 2012 International Conference on Biomedical Engineering and Biotechnology, iCBEB 2012*, 1:45–48, 2012. doi:10.1109/iCBEB.2012.62. (cited on p. 31)

[152] Sergio Maffioletti, Riccardo Murri, B Jonen, and S Scheuring. Computational workflows with GC3Pie. In *Proceedings of the EGI Community Forum*, 2012. (cited on p. 31)

[153] Marty Humphrey, Norm Beekwilder, Jonathan L. Goodall, and Mehmet B. Ercan. Calibration of watershed models using cloud computing. *2012 IEEE 8th International Conference on E-Science*, (1048222):1–8, October 2012. doi:10.1109/eScience.2012.6404420. (cited on p. 31)

[154] Bryan a. Tolson and Christine a. Shoemaker. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources Research*, 43(August 2006):1–16, 2007. doi:10.1029/2005WR004723. (cited on p. 31)

[155] John H Holland. *Adaptation in Natural and Artificial Systems*, volume Ann Arbor. 1975. URL: http://www.citeulike.org/group/664/article/400721. (cited on p. 32)

[156] T Blochwitz, M Otter, M Arnold, and C Bausch. The functional mockup interface for tool independent exchange of simulation models. In *Proceeding of 8th Modelica Conference, Dresden*, 2011. (cited on p. 33)

[157] J. C. Helton, J. D. Johnson, C. J. Sallaberry, and C. B. Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering and System Safety*, 91:1175–1209, 2006. doi:10.1016/j.ress.2005.11.017. (cited on p. 34)

[158] Andrea Saltelli, Stefano Tarantola, Francesca Campolongo, and Marco Ratto. *Sensitivity Analysis in Practice*. 2004. doi:10.1002/0470870958. (cited on p. 34)

[159] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global Sensitivity Analysis. The Primer*. 2008. URL: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470059974.html, doi:10.1002/9780470725184. (cited on p. 34)

[160] W van Meurs. *Modeling and Simulation in Biomedical Engineering: Applications in Cardiorespiratory Physiology*. McGraw-Hill Professional, 2011. (cited on p. 38, 39, 40)

[161] Ann L. Chervenak, Theo G M Van Erp, Carl Kesselman, Mike D'Arcy, Janet Sobell, et al. A system architecture for sharing de-identified, research-ready brain scans and health information across clinical imaging centers. *Studies in Health Technology and Informatics*, 175:19–28, 2012. doi:10.3233/978-1-61499-054-3-19. (cited on p. 43)

[162] Chris Anderson. The Long Tail: How Endless Choice Is Creating Unlimited Demand. *Word Journal Of The International Linguistic Association*, page 256, 2006. URL: http://www.amazon.co.uk/dp/184413850X, doi:30. (cited on p. 45)

[163] Timothy H. Vines, Arianne Y K Albert, Rose L. Andrew, Florence Débarre, Dan G. Bock, et al. The availability of research data declines rapidly with article age. *Current Biology*, 24:94–97, 2014. arXiv:1312.5670, doi:10.1016/j.cub.2013.11.014. (cited on p. 45)

[164] P. Bryan Heidorn. Shedding Light on the Dark Data in the Long Tail of Science. *Library Trends*, 57(2):280–299, 2008. doi:10.1353/lib.0.0036. (cited on p. 45)

[165] Christof Weinhardt, Arun Anandasivam, Benjamin Blau, Nikolay Borissov, Thomas Meinl, Wibke Michalk, and Jochen Stöß er. Cloud Computing – A Classification, Business Models, and Research Directions. *Business & Information Systems Engineering*, 1:391–399, 2009. doi:10.1007/s12599-009-0071-2. (cited on p. 45)

[166] Kristóf Ralovich, Lucian Itu, Viorel Mihalef, Puneet Sharma, Razvan Ionasec, et al. Hemodynamic assessment of pre- and post-operative aortic coarctation from MRI. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-*

Assisted Intervention, 15:486–93, 2012. URL: http://www.ncbi.nlm.nih.gov/pubmed/23286084. (cited on p. 45)

[167] Bernhard Gleich and Jürgen Weizenecker. Tomographic imaging using the nonlinear response of magnetic particles. *Nature*, 435(7046):1214–7, June 2005. URL: http://www.nature.com.ezproxy.techlib.cz/nature/journal/v435/n7046/full/nature03808.html, doi:10.1038/nature03808. (cited on p. 45)

[168] Emine U. Saritas, Patrick W. Goodwill, Laura R. Croft, Justin J. Konkle, Kuan Lu, Bo Zheng, and Steven M. Conolly. Magnetic particle imaging (MPI) for NMR and MRI researchers. *Journal of Magnetic Resonance*, 229:116–126, 2013. URL: http://dx.doi.org/10.1016/j.jmr.2012.11.029, doi:10.1016/j.jmr.2012.11.029. (cited on p. 45)

[169] Peter Hunter, Tara Chapman, Peter V Coveney, Bernard de Bono, Vanessa Diaz, et al. A vision and strategy for the virtual physiological human: 2012 update. *Interface Focus*, 3(2), February 2013. URL: http://rsfs.royalsocietypublishing.org/content/3/2/20130004.abstract. (cited on p. 46)