

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 particular goals of the thesis were:

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

CHAPTER 1. INTRODUCTION

- Identify use cases in other fields of biomedicine which are suitable to utilize the power of grid-computing and cloud-computing infrastructure.
- Develop and test prototype application utilizing grid or cloud technologies.

The thesis tries to discuss the hypothesis in different areas of biomedical research and its application and tries to find answers to the following additional questions:

- *Is it beneficial of utilizing grid-computing and cloud-computing technology for processing medical information and how?* In the time of starting the work on this thesis, it was believed that grid-computing may be an answer to scalability issues e.g. for exchanging large amount of data or doing demanding long-term computation.
- *What are the limitations of processing medical information in grid or cloud?*
- *How can the grid-computing and cloud-computing influence the direction of biomedical research?* There was an idea that grid-computing technology may inspire current architecture of distributed system for e.g. exchanging medical images (explained in section 3.1) and influence the direction of the information systems in hospitals.

Answers to these questions based on the following chapters are summarized in section 5.1.

1.2 Thesis Contribution

The author claims that the following contribution was made to the state of the art of biomedical informatics and computational biology.

- 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 project Globus MEDICUS and deployed it in the servers networked via academic network CESNET and integrated with existing regional PACS system. Other co-author coordinated the work with operators of regional PACS system and selected hospitals.
- 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 consolidate and share the physical resources to provide virtual environment for specific needs of particular use-cases. 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] and popularized as [16, 20]. The author of this thesis designed and customized the existing

network protocol to transfer voice signal losslessly and deployed application on remote virtual server. Other co-authors implemented the algorithms and application to analyze 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. Additionally, author advised and implemented several basic blocks and models of pulsatile 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 published as [4] and gradual development of related technologies were published and popularized as [8, 9, 18, 22]. Author of this thesis designed the architecture for distributed parameter estimation algorithm, integrated models and implemented pilot deployment utilizing scientific cloud-computing infrastructure. Other co-authors implemented complex models of human physiology in Modelica language and tested several algorithms for parameter estimation.
- Improved mathematical model of oxygen, carbon dioxide and hydrogen ion binding to Hemoglobin [7]. Author of this thesis implemented this model in Modelica and identified the parameters of the model. Other co-authors analyzed and proposed the new mathematical model based on basic physical and chemical laws and relation published in literature.
- 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 implemented simulation scenarios for educational purposes.
- Virtual patient simulator prototype registered as utility model by the Industrial Property Office in the Czech Republic [24]. Author designed and developed specific module to control multiple instances of virtual simulator within virtual classroom via a web server application. Other co-authors designed and implemented models of human physiology, clinically relevant educational scenarios and implemented 3D visualization of selected scenarios using game engine Unity 3D¹.

¹<http://unity3d.com/> accessed March 2015

1.3 Thesis Structure

This thesis is interdisciplinary, 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. The chapter 2 provides an overview of the state of the art in the theory of computation, parallel computation, distributed computing especially grid-computing and cloud-computing.

Introduction to selected areas of biomedical research domains and related particular methods are introduced in chapter 3.1 for sharing medical images, in chapter 3.2 for voice science and chapter 3.3 for 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 6 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 6 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 6 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 6 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 6 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 6 is the paper [6] *Simple models of the cardiovascular system for educational and research purposes* published in Mefanet Journal 2014.

Appendix 6 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.

From computer science (informatics) 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, next sections introduces briefly theoretical and practical aspects and consequences of theory of computation, parallelism, distributed computing. Section 2.1 introduces some important problem classes from the view of computational complexity theory.

Parallel computation introduce in some condition speedup needed for computing concurrently, 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 accomplish the task and solve the problem. There are several ways how to express algorithm, e.g. in text in programming language or pseudo-code or flowcharts are used. In further text the kopenograms will be used as graphical language for structured algorithms to supplement Unified Modeling Language (UML) diagrams proposed by Kofranek et al.[25]¹.

The computational complexity theory 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 and size of input problem n meaning that a time complexity denoted by $O(g(n))$ is not growing faster than the function g . Formally $f(n) = O(g(n))$ if and only if there exists constant c and positive integer n_0 that for each

¹<http://www.kopenogram.org> accessed March 2015

CHAPTER 2. STATE OF THE ART

$n \geq n_0: f(n) \leq c \times g(n)$.

$O(1)$ denotes algorithms that takes constant time regardless of size of the input. $O(n)$ denotes linear time algorithms. E.g. sequential search algorithm showed in pseudo-code and in kopenogram 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 $f(n) = 0.03 \cdot n = O(n)$.²

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

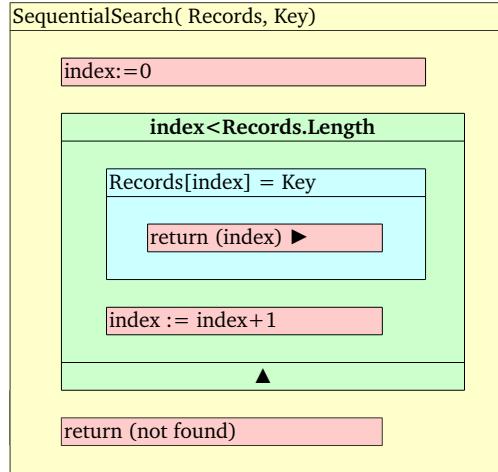


Figure 2.1: Pseudo-code (left) and kopenogram (right) of sequential search algorithm with complexity $O(n)$. The red blocks are commands, i.e. setting the index or returning results. The green block represents the loop with entry condition. The index is incremented and in programming languages this can 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.

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 recognized as tractable first noted by Cobham or Cook [28, 29]. Other algorithms whose time cannot be bounded by any polynomial function are called exponential and are recognized as intractable as published by Garey and Cook [30, 29]. Because 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.

There were identified a class of problem where it is not known whether a polynomial time algorithm exists, however, if such solution exists, it can be verified in polynomial time – class *NP-complete*. If some polynomial algorithm will be found in future solving some NP-complete problem, then a derived algorithm will solve other problems within the class

²In searching problem the sequential search algorithm is a brute force approach trying all values and better approach is e.g. binary search algorithm on sorted list taking logarithmic time complexity $O(\log(n))$ which outperforms the sequential search. B-trees are most used structure for holding the sorted list of elements in production application or databases[26, 27].

of *NP-complete* problems in polynomial time too as denoted by Cook and Karp[31, 32]. Current best known class of algorithm to solve such problems are based on brute-force search of all possible values and it is open question whether better exists.

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 $O(k^n)$ ³.

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 millisecond. 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 [35].

If we presume technological update and the computation speed will increase, 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 with only exponential algorithm are denoted as intractable.

	present computer	10 times faster	100 times faster	1000 times faster
n	3600000 1×	36000000 10×	360000000 100×	36000000000 1000×
n^2	1897 1×	6000 3.16×	18973 10×	60000 31.6×
n^5	20 1×	32 1.59×	51 2.51×	81 3.98×
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 [30]. The whole complexity theory is covered e.g. in book by Ch.Papadimitriou [36] or in a book by M.Sipser [37].

³e.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].

As that technological speedup will impact mainly the class of problems which are solvable by polynomial algorithm. Other non-exact methods are used to find at least some solution for the problems solvable only by exponential algorithms. These are e.g.:

- 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.

A speedup of a computation on P processors can be defined as:

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

In fig. 2.2 is example of serial and parallel execution computation of the same algorithm.

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:

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

On unlimited number of processors it can be formulated theoretical upper bound of speedup which depends on α only, denoted as Amdahl's law [38].:

$$S = \lim_{P \rightarrow \infty} \frac{1}{\alpha + \frac{1 - \alpha}{P}} = \frac{1}{\alpha} \quad (2.3)$$

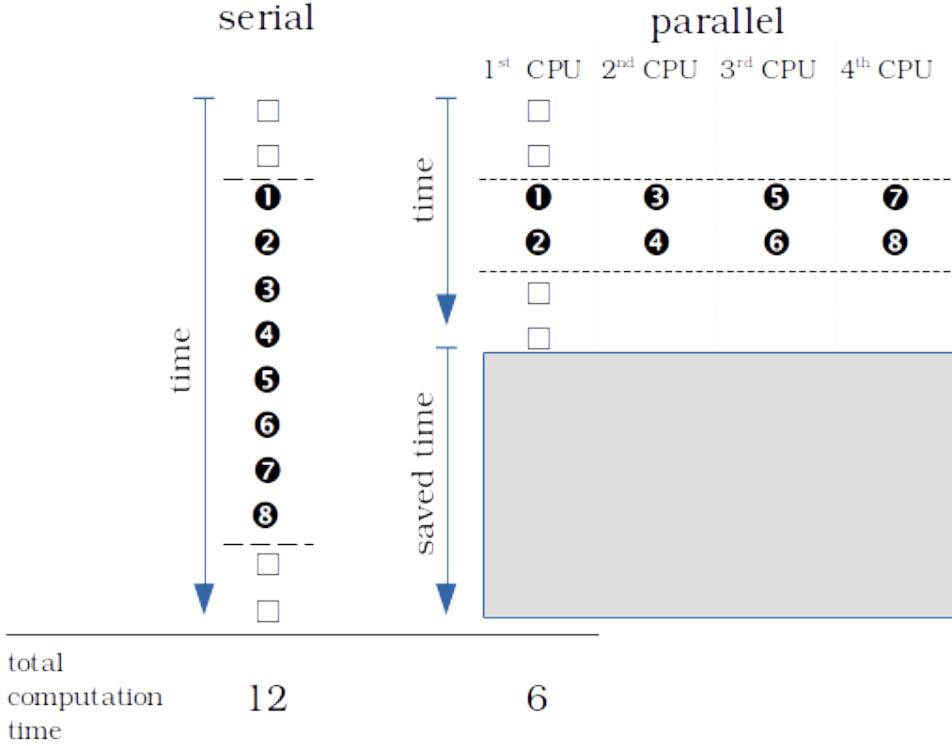


Figure 2.2: Comparison of serial and parallel execution of instructions. The instructions with numbers can be executed in parallel. In this case the serial computation takes 12 cycles, parallel computation on 4 CPU takes 6 cycles and the speedup is 2. If we have 8 CPUs, then a computation will be finished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

E.g. when a 33% of computation cannot be parallelized($\alpha = 0.33$) then the speedup on 8 processors can be theoretically $S(8) = \frac{1}{0.33+0.7/8} = 2.4$ and theoretical speedup on unlimited number of processors is $S = 1/0.33 = 3$. See more at fig 2.3.

However, α could be sometimes hard to estimate. Additionally, the computing of the fixed size problem on high number of processors can misrepresent the speedup expectation. Therefore, Gustafson reformulated the law and described another approach to 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 [39]:

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

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

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

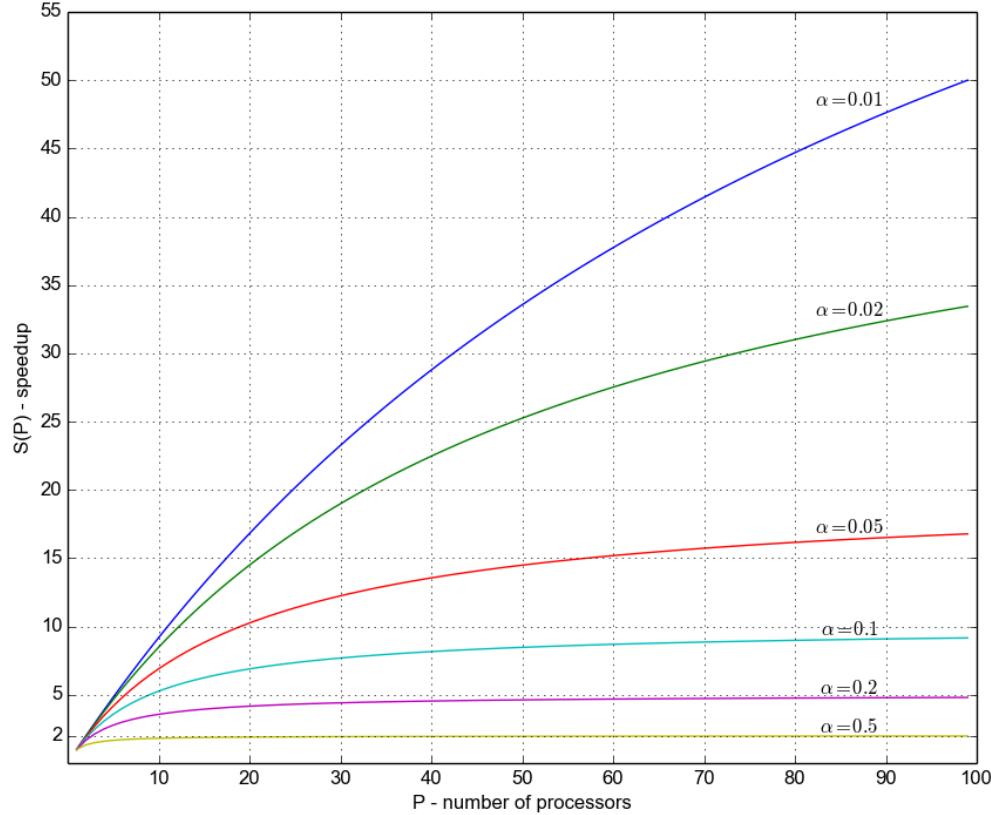


Figure 2.3: Speedup gained on 1 to 100 processors per Amdahl's law for different α values.

can be addressed with higher number of processors and this should be considered rather with regard to Gustafson's law (2.4).

2.2.1 Programming model

There are several levels how the parallelism is realized:

- *Instruction level parallelism* - if the instructions are independent, then they can be executed at the same time by multiple central processing unit(CPU), e.g. by several cores of the multi-core processor. Programs are usually written as a sequence of instruction and instruction parallelism depends mainly on compiler capabilities to recognize or reorder the instruction to execute the instruction in parallel. Instruction parallelism started to be systematically utilized in multi-core processor era.
- *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 programmable

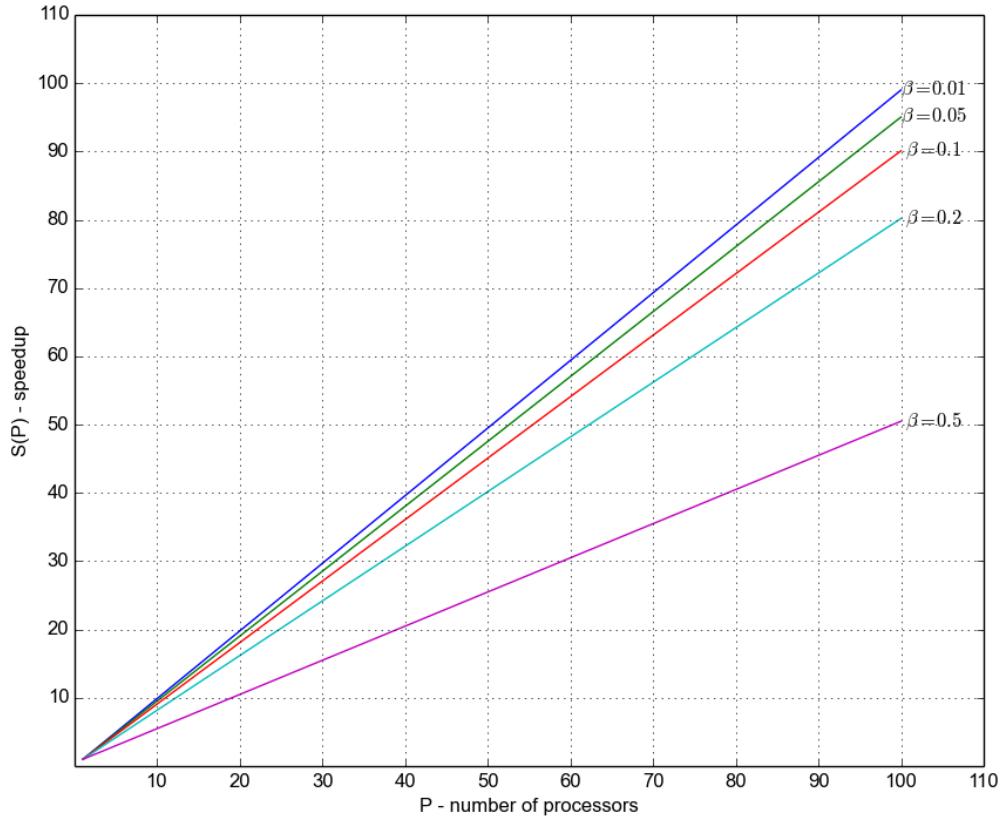


Figure 2.4: Speedup gained on 1 to 100 processors per Gustafson's law for different β values.

API such as CUDA⁴ and OpenCL⁵

- *Loop parallelism* - the computation may contain an iteration on some large data structure. Usually such iterative processing is programmed as a loop and if i^{th} iteration is independent on $(i-1)^{\text{th}}$ 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. E.g. Master/Worker pattern is realized(master sets up a pool of worker processes and a set of tasks which is distributed to them). Fork/Join pattern - main process forks into several threads executing concurrently 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 interact, these are the most common forms:

⁴<https://developer.nvidia.com/cuda-zone> accessed February 2015

⁵<https://www.khronos.org/opencl/> accessed February 2015

- The *threads* are several concurrent execution paths which are independent, but in general share the same memory. It was standardized e.g. as a POSIX threads (Pthreads) and implemented in many platforms. Further reading about Pthreads can be found in a book by Butenhof [40]. There are other implementation going beyond the POSIX standard introduced in other languages and programming environment.
- *shared memory*. The *OpenMP*⁶ is a shared memory application interface standardized and implemented in by several compilers for C, C++ and Fortran. It uses also multithreaded model, however, programming is task oriented and more abstract than using threads as described by Chapman et al. [41].
- *message passing*. The *Message Passing Interface*(MPI) is specification for performing task communication by passing messages between tasks. Further reading about MPI can be found in a book of Pacheco [42].

More about parallel programming models can be found in survey by Diaz et al. [43].

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 [44] are currently used to render 2D and 3D graphics.
- Parameter study, where the same computation is performed using different set of input parameters [45].
- Brute-force search algorithm, where subset of possible candidates for solution are generated and checked in parallel.
- Genetic algorithm and other evolutionary algorithms [46].

In contrast to embarrassingly parallel problems, there exists inherently sequential algorithms that cannot be significantly speeded up by parallel computing.

Both aspects scalability(speed up gained by parallel computing) and effectivity (time demand based on the size of input data = time complexity) should be considered as highly scalable algorithms can be outperformed by a sequential algorithm solving same problem with better time complexity as noted e.g. by Madden [47].

Further aspects of parallel computing can be found in the earlier book about design and build parallel programs of I.Foster [45], book of D.Culler et al [48] or book of T.Rauber and G.Räuder [49].

2.2.2 Summary

To summarize this section; parallel computing can introduce speedup on current computational technology and some computation problems may become feasible. Additionally,

⁶<http://openmp.org/> accessed February 2015

overhead caused by parallelization and fraction of non-parallelizable 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 computer network.

Main motivation of using distributed computing technologies are (1) sharing, storing and exchanging resources (2) provide and consume computational services (3) access the much higher capacity of storage and computation than available locally (4) connecting the users, developers, people.

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 resources - e.g. processor time - is not scheduled for particular task for some reason and task cannot finish 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 [50].

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 fulfill 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

The parallel programming model (section 2.2.1) is used to realize the distributed computing in local computer or server 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 usually incorporates one or several new layers, some of them are called middleware delivering services or APIs for application and hides specific implementation across heterogeneous computing platforms(operating system and hardware).

System Architecture

As the algorithms and programs needed to solve increasing number of problems and changing requirements, a view on software architecture is needed to construct and order several programs and algorithms into wider robust system aiming to solve broader set of 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.

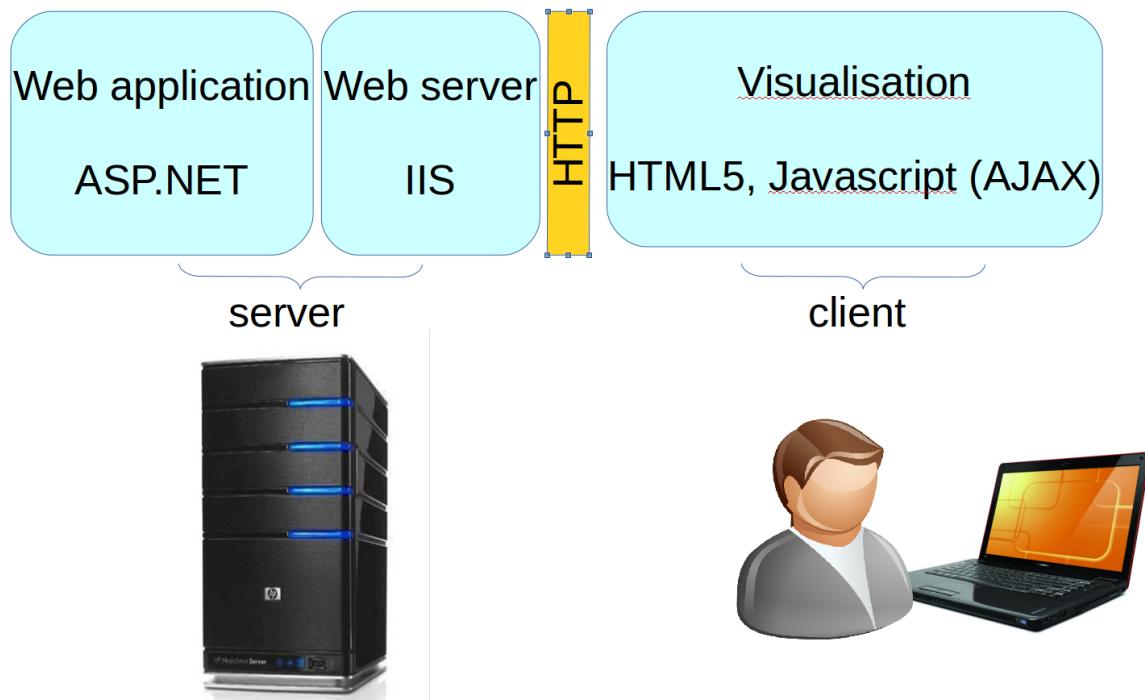


Figure 2.5: Example of client-server architecture involving web server which is middleware between web application and server platform. Client visualization in HTML5 web page communicates with server via HTTP protocol.

Service oriented architecture (SOA) is high level programming model based on self contained units of functionality –service – wrapped with documented interfaces. SOA

introduces a new service layer in the client-server architecture which separate service interface from its implementation. SOA principles and paradigms are described e.g. in book of T. Erl. [51].

Another approach represents objects and data of the system as resources with standard set of operations, create, read, update, delete (CRUD). Representation State Transfer (REST) specifies several architectural constraints that help scalability, performance and presents functionality via fixed number of operation and uniform resource location proposed by Fielding [52]. 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 current 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 [53].

Software architecture of the enterprise application, distributed systems and some repeating patterns are cataloged e.g. in a book by Fowler et. al[54] or Nilsson [55]. Integration patterns are discussed with focus on the ways of connecting heterogeneous parts of the system by Hohpe et al.[56].

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 and are relevant to the rest of this work:

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 maintained 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 idle 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 access to them is provided as a service with an impression of sole use it by user or process.

2.3.2 Research and education network

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

CHAPTER 2. STATE OF THE ART

The national grid initiative in Czech Republic -METACENTRUM⁷ is interconnected via high-speed network CESNET 2 utilizing the technology of transferring data over optical cables using Dense wavelength division multiplexing (DWDM) [57] ad seen in fig. 2.6.

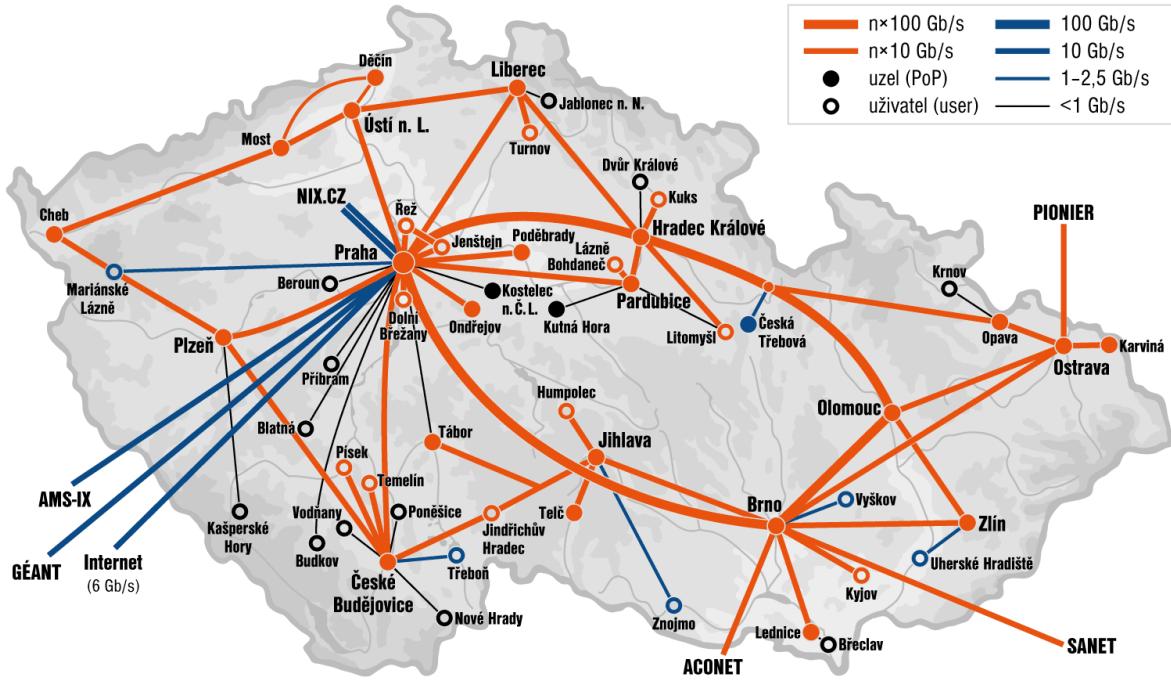


Figure 2.6: CESNET2 network topology as from december 2014 is maintained by association of universities and academy of science CESNET. Interconnects mainly departments of universities and academy of sciences via rented physical network. It provides connection to general Internet via the czech NIX.cz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

2.3.3 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 [58]. Foster et al.[59, 58] and Chervenak et al. [60] 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 storage, 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) [61]. 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

⁷<http://www.metacentrum.cz> department of CESNET as national grid infrastructure

X.509 certificate [62]. The administration and maintenance of such networked infrastructures are not 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)⁸ – was established during 2010 and supports integration and coordination activities of national grid initiatives (NGI) across national boundaries with respect to the scientific collaboration which goes also across national boundaries.

The growth 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 [63]. 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 [64] and is one of the largest grid deployed in the grid infrastructure. As the hardware infrastructure is built with a philosophy of federated access to resources owning by research institution, universities etc, other scientists from different scientific disciplines can become users of this powerful infrastructure too. The development of virtualization, 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 underutilized computers while preserving rights of the owners [65]. glite [66]⁹, ARC¹⁰ and Globus [67]¹¹ are major grid middleware operational in EGI.

Currently there is an effort to maintain interoperability to connect different application with different resources from the different grid infrastructures. Riedel et al. reports on such effort to involve technology providers as well as deployment teams to participate on open standards for security, data management etc. [68].

2.3.4 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 extraterrestrial intelligence (SETI@Home)¹². 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 amount 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 [69]. After the success of the SETI@Home

⁸<http://www.egi.eu>

⁹<http://glite.cern.ch> accessed February 2015

¹⁰<http://www.nordugrid.org/arc/> accessed February 2015

¹¹<http://toolkit.globus.org/toolkit/> accessed February 2015

¹²<http://setiathome.ssl.berkeley.edu/>

there were built general-purpose frameworks to facilitate development of projects that will use similar philosophy of computing on desktop computers connected via Internet such as BOINC [70], SZTAKI extension to BOINC [71, 72], XtremWeb [73] 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 Large Hadron Collider (LHC) project on the desktop grid infrastructure [74, 75].

The average performance of BOINC projects is 8.073 PetaFLOPS with active 294 764 volunteers computing on 502 238 computers (March 18th 2015). E.g. SETI@Home 24 hour performance is 1.95 PetaFLOPS. Although desktop grids and service grids are two different approaches to gather computing power from large number of computing resources, there is an effort to interoperate and e.g. share capacity for a project, e.g. EdgE project published by Kacsuk et al.[76] and Urbah et al. [77].

2.3.5 Virtualization

Virtualization technology separates the physical hardware layer from the software environment emulating a new virtual hardware layer. Hypervisor or virtual machine manager manages guest virtual machines, translates I/O operations between virtual device and physical device, translates instructions from virtual CPU to physical CPU. This introduces some overhead and performance degradation of virtual system compared to physical. However, recent virtualization technology introduced several techniques which reduces overhead and eliminate specific hardware features and instructions which are hard to virtualize as reported by Barham et al. and Youseff et al. [78, 79]. 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 [80]. Currently there exists several commercial, free or even opensource virtualization implementation which are provided by different vendors and it's hypervisor - VMWare¹³, XEN¹⁴, KVM¹⁵, VirtualBox¹⁶ etc.

2.3.6 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 [81].

¹³<http://www.vmware.com/> accessed March 2015

¹⁴<http://www.xenproject.org/> accessed March 2015

¹⁵<http://www.linux-kvm.org/> accessed March 2015

¹⁶<https://www.virtualbox.org/> accesse March 2015

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 [82].

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 EC2¹⁷, Microsoft Windows Azure ¹⁸, Google cloud¹⁹ 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 proposed by Anjum et al. [83]. Access is provided to the same users of grid-infrastructure and currently most used platforms are OpenNebula [84] and OpenStack [85].

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[86].

2.3.7 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. 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 [87, 88]. 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 heterogeneous 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

¹⁷<http://aws.amazon.com/ec2/> accessed February 2015

¹⁸<http://azure.microsoft.com/> accessed February 2015

¹⁹<https://cloud.google.com/> accessed February 2015

MTC was proposed and implemented by Raicu et al.[89, 90, 91]. 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 [92].

2.3.8 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 [93]. 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. [94, 95, 96]. Workflows in cloud computing are covered by web technologies programming languages (Javascript) [81].

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-infrastructure a mutual workflow management system is used to integrate them as proposed by Kacsuk et al. [97, 98] or an interoperability is solved by separating abstract workflow representation and concrete implementation showed on selected existing workflow systems introduced by Planensteiner et al. [99].

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[100]. For building the scientific gateways several frameworks were developed e.g. Apache Airavata [101, 102] or WS-PGRADE/gUSE[103]. And the concrete instances are available for broader area of scientific community.

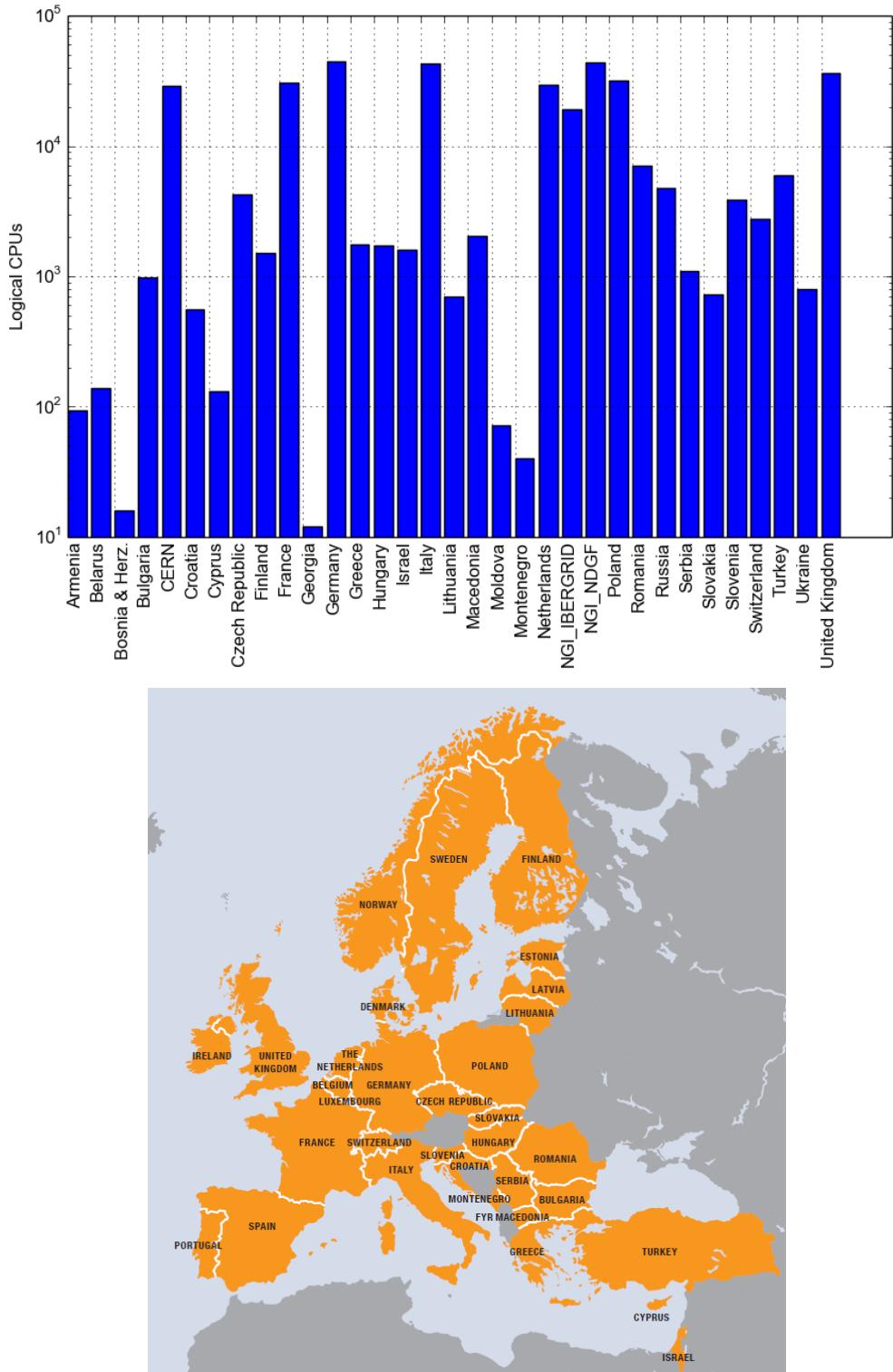


Figure 2.7: Countries involved in EGI and number of CPUs within EGI infrastructure in 2012, the total logical CPU capacity at the end of 2012 was 349,720 cores(in graph per countries). CPU capacity was 433,957 cores in March 2014. Sources: EGI Compendium 2013, EGI statistics at <http://www.egi.eu>.

CHAPTER 2. STATE OF THE ART

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 provider¹ or application database are available in broader environment e.g. in EGI.eu application database². Additionally, workflow systems and scientific gateways mentioned in section 2.3.8 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 user interface, integrate with existing applications, utilize protocol compatibility etc. An effort to obtain first results is high, however if the prepared templates, script or application is reused for further computational request an 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 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

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

²<https://appdb.evi.eu/> accessed February 2015

CHAPTER 3. METHODS

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[104].

Acquisition of the medical image is covered with different modalities (different types of equipment and sensors) by radiologists or other specialists. DICOM³ format and protocol becomes actually 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.

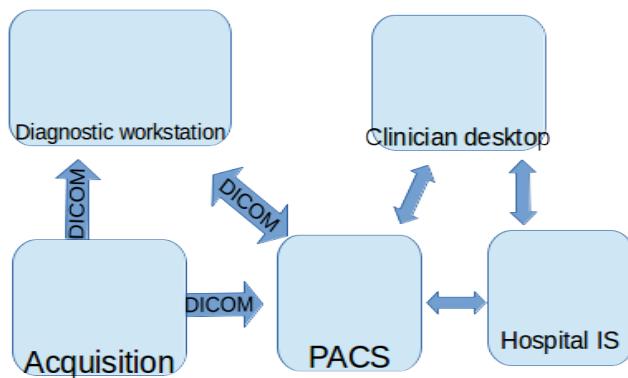


Figure 3.1: Typical workflow of medical image in hospital. Data acquisition is made by modalities (magnetic resonance, ultrasonography, X-ray radiography, etc.) and using DICOM format and protocol it can be directly transferred and visualized by diagnostic workstation. With metadata filled by an expert 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 telecommunication and network technologies enabled telemedicine – providing healthcare over remote distance. It requires to share and exchange sensitive data of real patient among different healthcare providers and such data may be very valuable for further research. Security and encryption should be addressed and DICOM standard itself doesn't solve security issues appropriately, thus encryption 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 Virtual Private

³DICOM: <http://dicom.nema.org/> accessed January 2015

Network VPN channel to the central node and exchange of medical images are realized by routing the data flow from one VPN channel to the other⁴. Another approach is used in the project MEDIMED held by Masaryk University in Brno. Instead of dedicated VPN channel, they use Secure Sockets Layer (SSL) encryption over standard TCP/IP communication and regional hospitals and healthcare providers are interconnected via the MEDIMED servers as presented by Slavicek et al. [105]. In other countries, there were tested cross-border teleradiology in projects Baltic e-health, R-Bay and others published by Ross et al. [106] and Saliba et al. [107]. 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 birth 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.[108, 109] 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 published⁵. Similar effort was done with a project Medical Data Manager which uses gLite grid middleware published by Duque, Montagnat et al.[110, 111]⁶ or MediGRID project published by Krefting et al.[112, 113]. Additionally, processing of images within selected use-cases supported by the grid-computing infrastructure is introduced[113]. Health-e-child project aimed to interconnect research institution and hospitals in United Kingdom, France and Italy for the purpose of grid-based healthcare platform for pediatric health-care [114]. 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.[115]. 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.[116].

Storing the sensitive medical information even de-identified or anonymised is 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. [117].

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 in-

⁴ePACS:<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

CHAPTER 3. METHODS

frastructures. 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 2.3.3). 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 transferred via different channels, *partial file transfer* some application may not need to access the whole file, but a smaller portion of it etc. as described by Foster et al. and Allcock et al. [118, 119]. 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 [118].

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 [108, 109] contains a DICOM Grid Interface Service (DGIS) and integrates the open source PixelMed™ Java DICOM Toolkit⁷ 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 anywhere 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[105]) 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.

⁷<http://www.pixelmed.com/> accessed February 2015

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 (FFT) with computational complexity $O(n \log(n))$ [120, 121]. 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 [122, 123].

Other methods to analyze 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 visualizing and analyzes of 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 [124, 125].

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 [126, 127]. 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 primarily in Microsoft Windows platform, however, today clients and servers exists for several other platforms. Next to remote command-line, remote execution it allows accessing 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.[126, 127]. The implementation is done in MATLAB environment utilizing Signal Pro-

CHAPTER 3. METHODS

cessing Toolbox⁸ 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 6.

The computation of frequencies and amplitude from the recorded samples utilizes effective Fast Fourier Transformation which has time complexity $O(n \log(n))$. 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.

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 its 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.

Application of the mathematical modeling techniques towards the biomedical research is sometimes called as systems biology approach combining the reductionism and integration as denoted by Kohl et al.[128]. Application towards the clinical practice 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.[129].

One of the earliest complex and integrative modeling effort was a model of circulation and its regulation published by Guyton et al. in 1972 [130] which via derivative and

⁸<http://www.mathworks.com/products/signal/> accessed February 2015

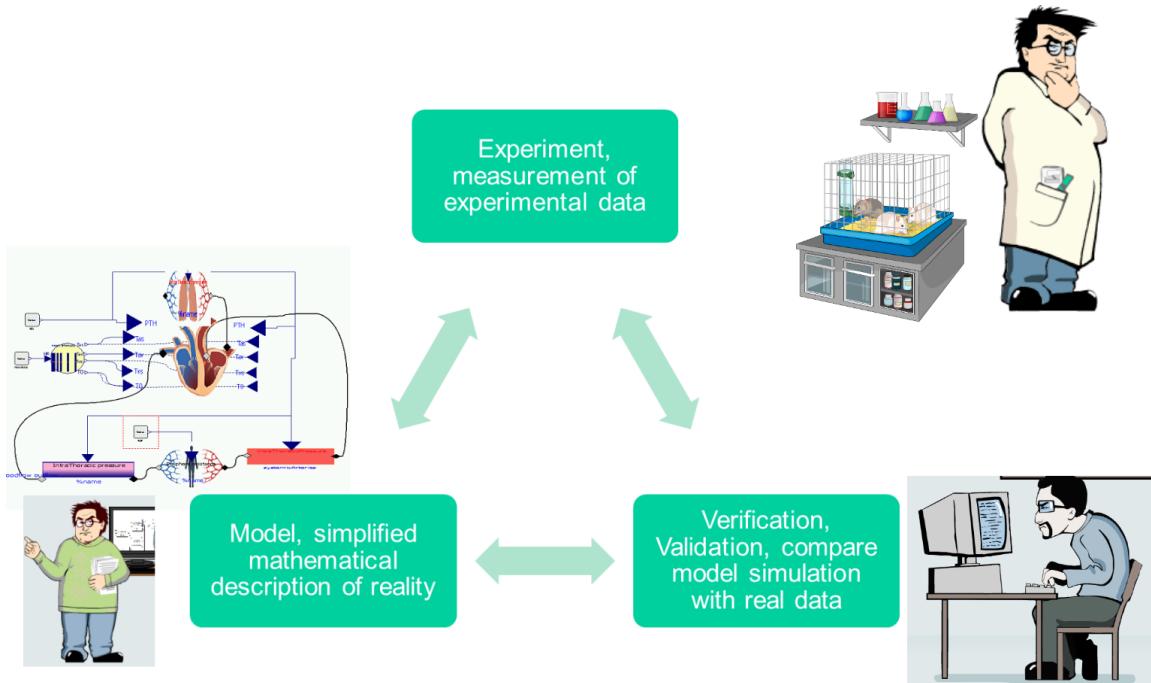


Figure 3.2: Schematic illustration of scientific process. The experiments produce data which are interpreted and a hypothesis is formalized as a model. Validation compares the model simulation with experiment, if model satisfies the criteria - is in agreement with real experiments, then the validated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

technological upgrade continues as "Human Model" or "HumMod" introduced by Hester et al. [131, 132] with a focus on integration effort. Different approach of modeling human physiology is a database of smaller models focusing on some particular physiological phenomenon. E.g. the NSR Physiome project introduces JSIM⁹ Java based simulation system to support modeling in physiology. Repository of several hundred of models were published using this system [133]. The similar effort is done by IUPS Physiome project and repository of models are based on XML standard languages CellML and FieldML [134, 135]. 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 [136, 137].

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 association¹⁰. 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 modeling

⁹JSIM: <http://www.physiome.org/jsim/> accessed January 2015

¹⁰<http://www.modelica.org> accessed February 2015

CHAPTER 3. METHODS

and computing. E.g. Kofranek et al. describes Guyton's 1972 model in MATLAB® Simulink [138] and the derivative HumMod in acausal object-oriented Modelica language [139, 10]. Fernandez et al. describes models of cardiovascular pulsatile system using MATLAB Simscape [140] and recently in Modelica [141].

Thus, there is an open debate whether in-house domain specific language and tools like JSIM, CellML and FieldML, SBML or HumMod reached its 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 covering complexity of models like HumMod.

The paper [5] *Modeling of Short-term Mechanism of arterial pressure in the cardiovascular system: Object-oriented and acausal approach* in Appendix 6 published disputation about causal and acausal approach in using Modelica for modeling pulsatile cardiovascular system (CVS) and possible enhancement for more complex models.

The paper [6] *Simple Models of the Cardiovascular System for Educational and Research Purposes* in Appendix 6 published detailed methodology of modeling lumped parameter pulsatile CVS in Modelica.

Common guide to the Modelica language and it's capabilities are in the book of Fritzson [142] or in the on-line book by M.Tiller [143].

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 sometimes called system identification and the objective of parameter estimation is usually to minimize the following function (to find least squares of the differences between predicted and measured values):

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

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 . In general mathematical models of biological systems are in most cases non-linear and some of them are non-differentiable, therefore global optimization methods must be used. Algorithmically, the problem of parameter estimation was shown to belonging to the *NP-complete* problems [144] which implicates that current exact algorithm is brute force search – trying all possible values of parameters and simulate the model with them and find minimum of the objective function 3.1. Further reading about parameter estimation and system identification can be found in book edited by Eykhoff [145] or book of Khoo [146],

p. 159].

The heuristic methods (evolution strategies), randomization methods (Monte Carlo) and others can be used to find at least some solution in a reasonable time. The evolution strategies were identified as robust with potential to utilize parallel computing as shown by Moles et al.[147]

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 of the software Wolfram *Mathematica* were used [148]. The iterative improvement method in the software MATLAB Simulink® was used in estimating 2 parameters of simple cardiovascular model by Takahashi et al. [149]. Several methods were compared in estimating multiple parameters of cardiovascular system in MATLAB Simulink® by Abbass et al. [150].

Maffioletti et al. published GC3Pie framework utilizing evolutionary algorithms and introduced workflow to identify parameters of models for economical predictions using grid computing [151]. Humphrey et al. calibrated hydrology models utilizing commercial Windows Azure cloud computing infrastructure with a significant speedup on the modified dynamically dimensioned search algorithm[152, 153].

3.3.3 Methods for Parameter Estimation



Figure 3.3: Kopenogram of genetic algorithm.

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[154] and the algorithm steps are schematically presented in figure 3.3.

The iteration within the loop " $\nabla \dots$ while not satisfied" depends on previous iteration,

CHAPTER 3. METHODS

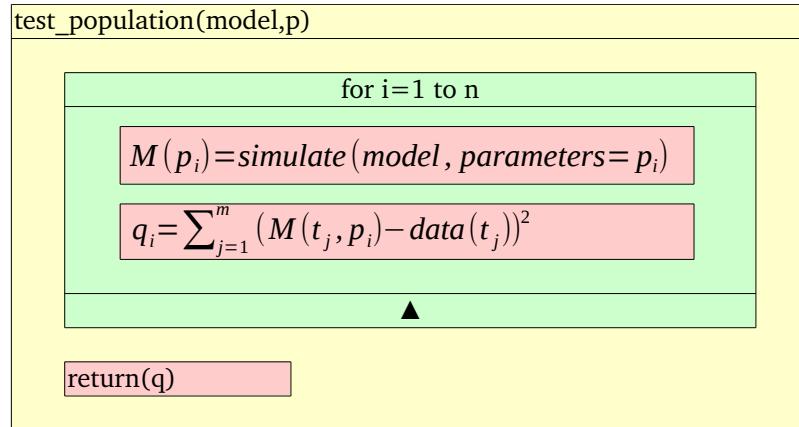


Figure 3.4: Kopenogram of genetic algorithm and specific test of the population for quality in case of parameter estimation. Model is simulated according to individual i with parameters p_i and the quality q_i is counted per the objective function 3.1.

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 therefore loop parallelism (section 2.2.1) can be utilized and implemented here.

Architecture of system for parameter estimation

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) which is standardized XML metadata packaged together with binary library .DLL (or .SO) following standardized API [155]¹¹. In the time of writing the thesis the most stable Modelica tool export was Dymola¹² 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 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 framework¹³ 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

¹¹<https://www.fmi-standard.org/> accessed February 2015

¹²<http://www.dynasim.se> - Dymola tool, accessed March 2015

¹³<https://servicestack.net/> accessed February 2015

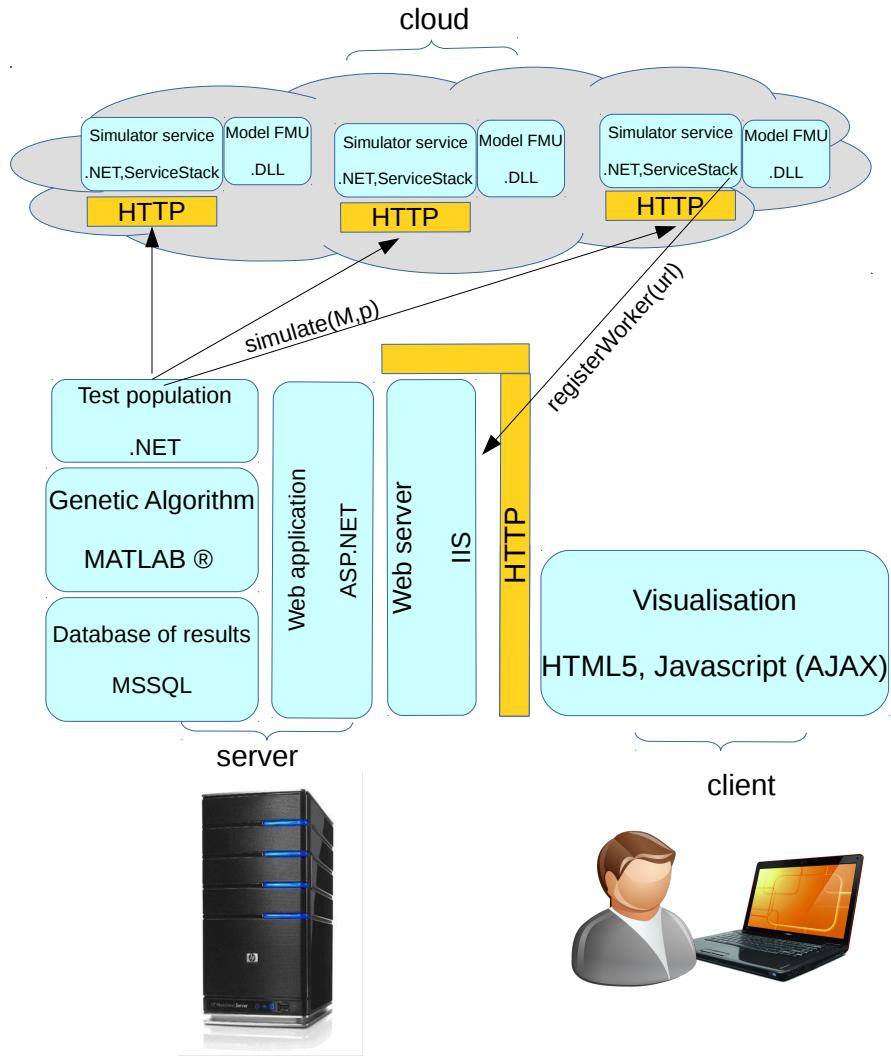


Figure 3.5: Architecture of the system employing genetic algorithm and distributing the task *simulate* into cloud computing environment.

described in section 4.3.

3.3.4 Parameter Sweep

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

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. [156] or a books by Saltelli et al.[157, 158].

CHAPTER 3. METHODS

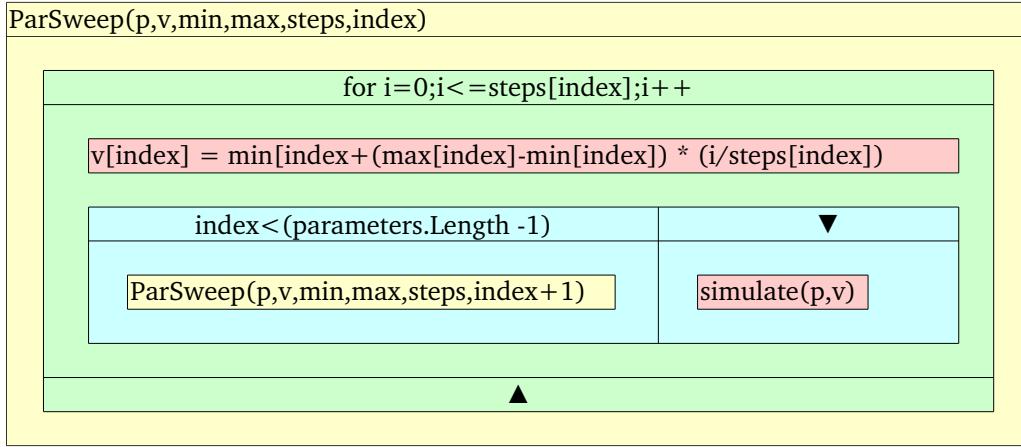


Figure 3.6: Kopenogram of recursive parameter sweep algorithm. $p, v, \text{min}, \text{max}$ and steps are arrays with the same dimension holding parameter name, value, starting and stopping value and number of steps which needs to be performed between starting and stopping value per each index.

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: $O(1000^n)$. 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.[98].

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[70]¹⁴ is customized following the task parallelism and master/worker programming model (mentioned 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.

The results are described in section 4.3.

¹⁴<http://boinc.berkeley.edu/> accessed February 2015

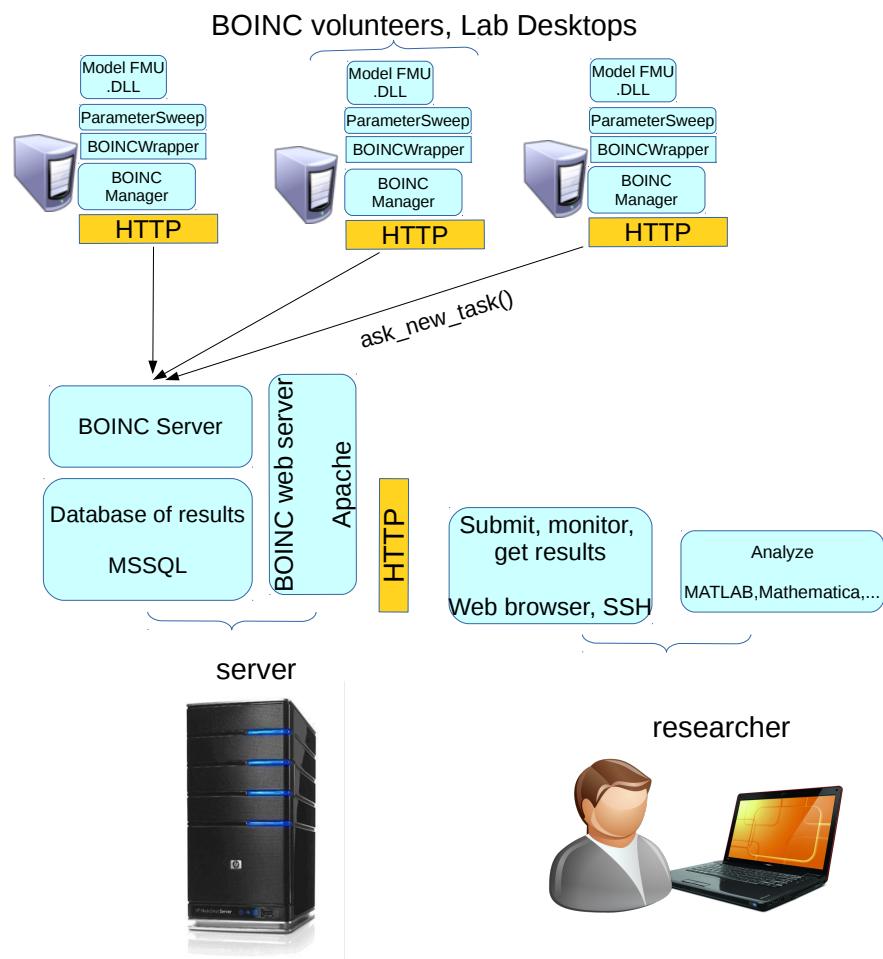


Figure 3.7: Architecture of parameter sweep application. The whole parameter space is divided into smaller spaces which are resolved by the BOINC workers

CHAPTER 3. METHODS

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 6 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 6 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 6 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

CHAPTER 4. RESULTS

Globus MEDICUS. The virtual machine template was also deployed to different cloud computing infrastructures. One to the Amazon EC2¹ and second to the pilot scientific cloud launched in the begining of 2012 –MetaCloud². 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 6 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 6 where a mathematical model of hemoglobin integrating O₂, CO₂ and H⁺ binding based on theoretical principles, which were verified on the parameter estimation algorithm system[4] together with methods available in Wolfram MATHEMATICA 9.0³.

Thus overall performance and speedup estimation were tested against the Modelica implementation of complex physiological model HumMod [139], the Modelica implementation of model of haemodynamics of cardivaoscular system presented by Meurs [159], 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 · sin(b · (x - c)) + x · d and named as "SinusCurve".

complexity	name	T1 _[s]	T2 _[s]	T3 _[s]	T4 _[s]	α	S
high	HumMod [139]	4639	4639	4618	4616	8.858×10^{-5}	11 290
medium	Meurs2011[159]	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).

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

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 [139]	6463	6461	6451	6458	0.000 352 8	2835
medium	Meurs2011[159]	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.

4 groups depending on its 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 METACENTRUM⁴. 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.

model name	distributed in cloud				distributed in local cluster				
	overhead		est. serial		overhead		est. serial		
	T2-T3 _[s]	fraction[%]	T _{es} [s]	S _{es}		T2-T3 _[s]	fraction[%]	T _{es} [s]	S _{es}
HumMod [139]	20.98	0.4523	4619	1.005	9.858	0.1525	6453	1.002	
Meurs2011 [159]	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_{es} – estimated time of serial version of computation. S_{es} – estimated speedup of serial version of computation against the parallel on 1 processor.

The speedup was measured on 10 - 60 CPUs and compared to predicted speedup as seen in figure 4.1). Different measurement was done using 80 and 160 CPUs with as seen in table 4.4.

complexity	name	S(80) _[s]	S(160)
high	HumMod [139]	85.20	161.9
medium	Meurs2011[159]	72.73	68.94
low	Matejak2014[7]	15.70	15.37

Table 4.4: Scalability on 80 CPUs and 160 CPUs of genetic algorithm with population 640 individuals for 20 generations on different cluster, performed about 10 times more simulation. Speedup estimated based on the T1 column of table 4.1.

To summarize the results, the simple models scale up to the 20 processors with speedup of 15. Medium scales up to 80 processors with speedup about 72 and complex models scales up to 160 processors with speedup near 160. Practically there were obtained good approximation after 20 generation and good estimation after another 180 generations, which

⁴<http://www.metacentrum.cz> accessed March 2015

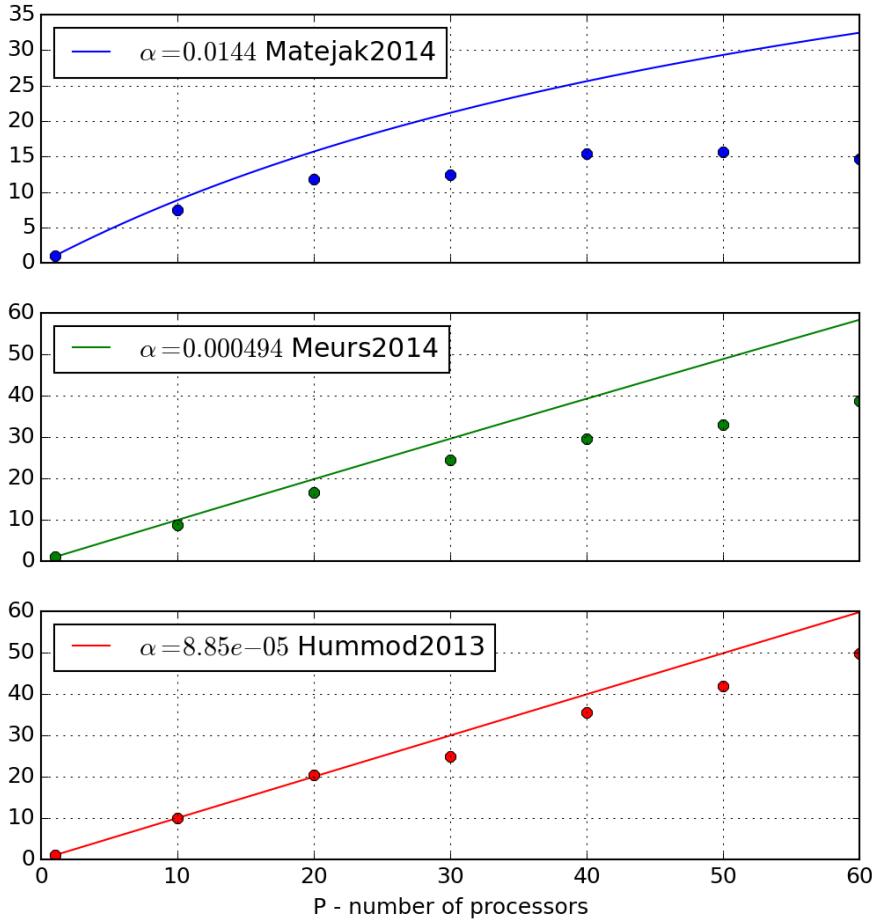


Figure 4.1: Estimated speedup (lines) per Amdahl's law (eq.2.3 [38]) for different α of several Modelica models and real measured speedup (points) on cloud deployed on 1-6 virtual machines on physical hardware (2x 6-core Intel E5-2620 2GHz, 1Gbit/s Ethernet).

implies that the computation time could be reduced from 5 days to 48 minutes in the case of HumMod and from 18 hours to 15 minutes in case of medium complex model.

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 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 distribut-

ed computation e.g. in cloud-computing environment is worth to be used. This is case of medium and high complex model simulation of "Meurs2011"[159] and "HumMod2013"[139].

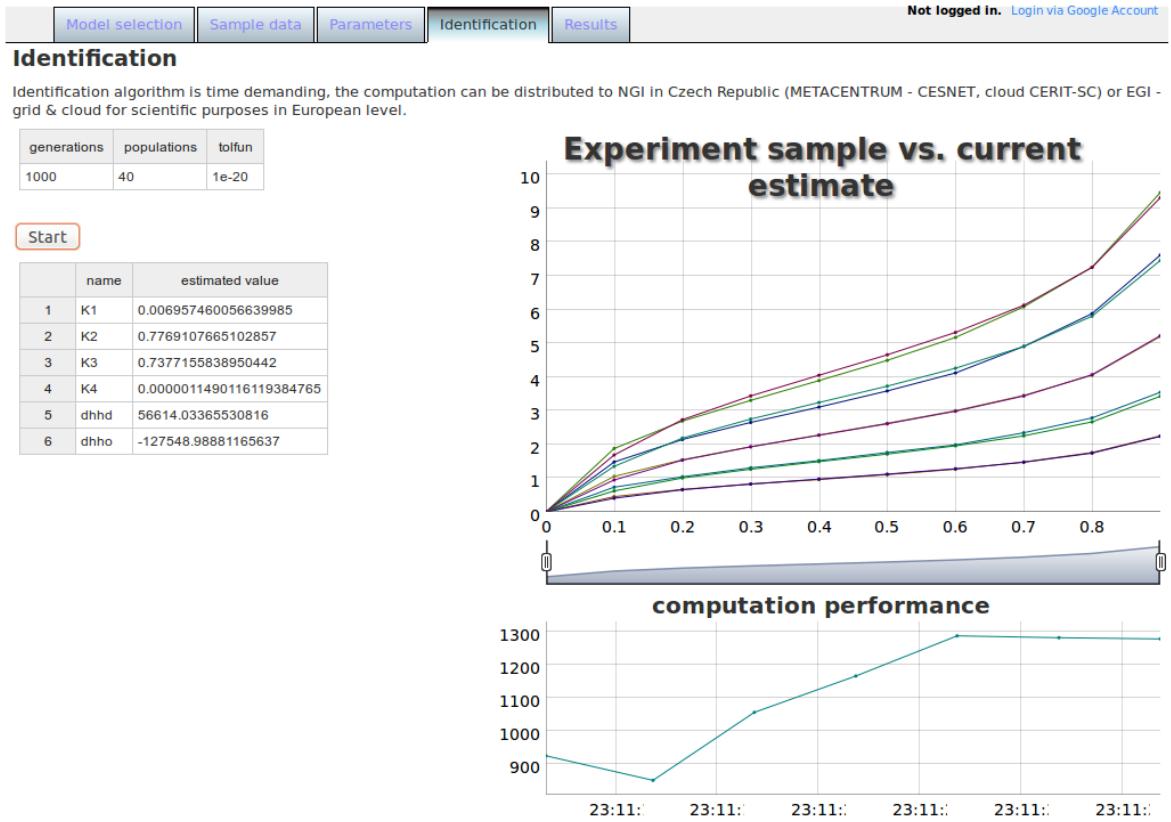


Figure 4.2: User interface of web application for parameter estimation. In this case for the model Matejak2014[7]. The top left table list the parameters for the genetic algorithm(1000 generations with population size of 40 and cumulative change in generation limit which ends the algorithm earlier). The middle left table shows model parameters and current best values which fits the sample data. Chart shows how the sample data fits with the model simulation. The right bottom chart shows the performance of computation in number of model simulations per second.

4.3.1 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.

4.3.2 Remote Simulation and Local Visualization

As an extra result of an architecture for parameter estimation is a hybrid web simulator system, where single instance of a worker node is utilized as a backend for simulation

CHAPTER 4. RESULTS

engine. Frontend presented as a web application implemented using HTML and Javascript language visualizes the simulation values obtained from the worker node performing the simulation as seen in figure 4.3. The results were published in [9] and popularized in [21].

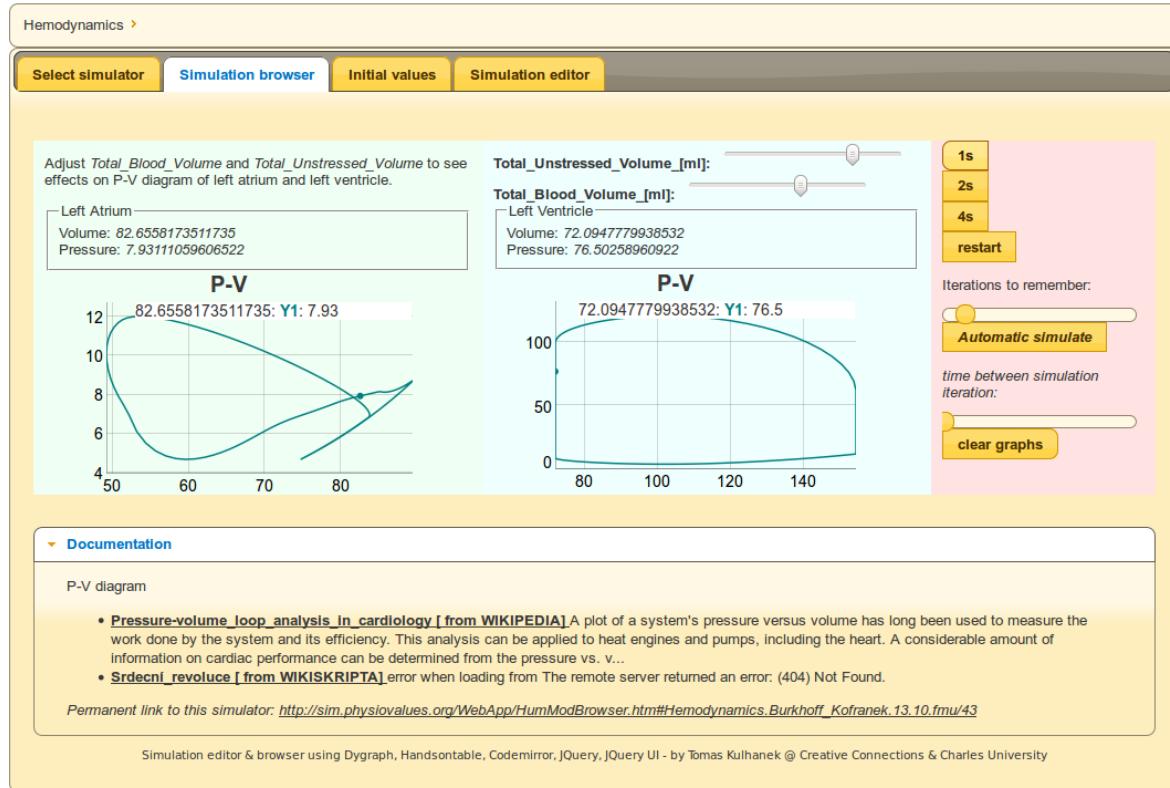


Figure 4.3: Web application to visualize simulation. In this case pressure volume diagrams of left atrium and left ventricle of the model of hemodynamics of cardiovascular system.

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 [105]) and a grid based solution (Globus MEDICUS[109]). Remote Desktop Protocol (RDP) is also the key point in integrating the application of voice analysis[126] into remote environment accessible via Internet presented in section 4.2. Key point in case of parameter estimation is a standard Functional Mockup Interface (FMI)[155] which allows controlling and simulating a physiological model in customized tool not related to modeling technology as presented in section 4.3.

The selection of a joint element also increase chances of reusability of such system in future development when usually requirements change and reconstruction of a system or architecture is needed. E.g. the presented solution based on the Globus MEDICUS is in general a data warehouse storing one or more copies of DICOM images, however, federated files and metadata stored within home institutions which shares only network infrastructure to interchange the DICOM studies seems to be preferred and more acceptable solution by hospitals thus the authors of Globus MEDICUS in their further development followed a way of federation of medical images stored within home institutions as published by Chervenak et al. [160]. The grid-computing infrastructure seems to be suitable for research and educational purposes rather than for clinical use.

The remote access to application via network protocol keeps the majority of user experience in the case of remote voice analysis presented in section 4.2. Such kind of service can be deployed on any web server and a occasional need to educate or perform higher number of analysis concurrently can be satisfied with cloud-computing deployment. The application process sound signal which is currently analysed by Fast Fourier Transformation algorithm quite effectively. Another challenge becomes to analyze the sound signal connected with high speed video or videokymography methods which needs to transfer, process and store large ammount of data. Such future application might utilize the results

CHAPTER 5. CONCLUSION

of grid-based systems for sharing medical images.

In the case of application for parameter estimation presented in section 4.3 the computation is sensitive on communication overhead. For simple models local high performance computing (HPC) resources are most beneficial, for medium and highly complex models the deployment of worker nodes into cloud-computing environment is worth to consider. The application for parameter sweep is embarrassingly parallel and suitable for high throughput computing (HTC) which is the focus of current grid-computing infrastructures.

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. This determined the platform of the worker node and 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 compatible system. To utilize service-grid infrastructure an export of the model into FMU library and implementation of the wrapper service must be done in the grid-computing platform which is usually Linux based system. Another option could be to use WINE¹ – compatibility layer capable of running Windows applications on several POSIX-compliant operating systems, such as Linux, Mac OSX, BSD.

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 case of integrating with service grid-middleware or with desktop-grid framework, expert knowledge is needed to configure and customize the system. This is the case of sharing medical images presented in section 4.1) and parameter estimation and parameter sweep which was tried with the desktop-grid approach - BOINC framework^{4.3.1}. Virtualization facilitates integrating effort as presented in case of remote analysis of human voice (section 4.2) and as presented in the case of deployment of worker nodes in cloud computing environment for parameter estimation (section 4.3).

Based on the previous results and ideas it can be formulated the answer to the questions from the section 1.1:

- *Is it beneficial of utilizing grid-computing and cloud-computing technology for processing medical information and how?*

Grid-computing and cloud-computing may significantly speedup parameter study of medium and complex models in computational physiology. Such speedup might influence applicability in clinical use. For the case of sharing and processing medical images or analysis of voice signals, grid-computing or cloud-computing introduce technology to facilitate cooperation among community of users from different geographically dispersed areas and facilitate sharing large data sets.

¹<https://www.winehq.org/> WINE. Accessed March 2015

- *What are the limitations of processing medical information in grid or cloud?*

Limitation are given by the effort needed to integrate or port an application to do computation or share data. Cost of porting application to cloud-computing is reduced by virtualization technology than to grid-computing environment, which needs additional work to adapt the application for grid-computing platform and API.

From programming model point of view, limitation are given by the theoretical features of algorithms and problems to solve. Grid-computing and cloud-computing is not general solution for hard problems (NP-complete problems) as discussed in section 2.1. However connected with non-exact methods a concurrent processing of many tasks may bring acceptable non-exact solution.

- *How can the grid-computing and cloud-computing influence the direction of biomedical research?*

There are research infrastructures which were established to coordinate the research in biology and medicine, e.g. Integrated Structural Biology Infrastructure for Europe (INSTRUCT)², European Life Science Infrastructure for Biological Information (ELIXIR)³, European Biomedical Imaging Infrastructure (Euro-BioImaging)⁴ and others which technologically rely on grid-computing and cloud-computing infrastructures for science. The purpose of these initiatives is to understand high-level phenotypes from genomic, metabolomic, proteomic, imaging and other types of data and they require also multiscale mathematical models and simulation as noted e.g. by Hunter et al. [161] in his strategy for Virtual Physiological Human (VPH)⁵.

The integration with multidimensional models of geometrical, mechanical properties and the time-dependence of the compartments data taken from the medical and biological repositories is a challenge for current "physiome" projects and might be very beneficial with current most complex models of human physiology which are based mainly on lumped-parameter approach. The area of parameter study presented in section 4.3 may deliver more exact lumped-parameter models and practical application for clinical and further research towards patient specific health care, in silico trials and drug discovery.

Based on the previous answers, it can be formulated another question for further research in the technology domain:

How can biomedical research influence the direction of grid-computing and cloud-computing development?

One area of discussion about this theme is how to preserve the scientific data to prevent loss of them [162, 163] and how to facilitate access to computational resources for

²<https://www.structuralbiology.eu/> accessed March 2015

³<http://www.elixir-europe.org/> accessed March 2015

⁴<http://www.eurobioimaging.eu/> accessed March 2015

⁵<http://www.vph-institute.org/> accessed March 2015

CHAPTER 5. CONCLUSION

large amount of small scientific groups which have limited resources to port, integrate or customize their current tools and processes – to support "long-tail" of science. The "long-tail" movement was noted and described first by Anderson [164] in business domain. The long-tail term comes from a feature of statistical distribution e.g. pareto distribution where only few (e.g. 20% – head) elements has high probability of some event (e.g. product being sold), while the rest (e.g. 80% – tail) has small probability. Thus, most business focus on hits (20% of products, the 80-20 rule). The expansion of Internet and it's related technologies caused reduced sales, marketing and delivery costs for the products from the niche (80% of products) – long-tail. Strategy focusing on this kind of products became profitable and successful e.g. for companies such as Amazon or Apple.[164]. 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]. Platform as a service may be a type of access to cloud-computing infrastructure to bridge the gap between programming application utilizing the powerful scientific infrastructure and keep focus on domain specific research as it is achieved by research portals and workflows for grid-computing infrastructures presented in section ??.

5.2 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. 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. Methods for parameter study was shown and parameter study of complex models gain substantial speedup by utilizing cloud-computing deployment, which makes such kind of complex study applicable in physiological and biological research.

Chapter 6

Appendices A-G

included separately

CHAPTER 6. APPENDICES A-G

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](https://doi.org/10.1145/1551722.1551732). (cited on p. 1, 2, 4, 37)
- [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)
- [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)
- [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](https://doi.org/10.1109/BHI.2014.6864463). (cited on p. 1, 3, 4, 38)
- [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.combiomed.2014.08.025](https://doi.org/10.1016/j.combiomed.2014.08.025). (cited on p. 1, 3, 4, 30)
- [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, 3, 4, 30)
- [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](https://doi.org/10.3109/00365513.2014.984320). (cited on p. 1, 3, 4, 38, 39, 40, 41)

BIBLIOGRAPHY

- [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, 42)
- [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 (WORLD-COMP'13), International Conference on Modeling, Simulation and Visualisation Methods (MSV'13)*, pages 182–188, 2013. (cited on p. 1, 3, 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, 3)
- [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 foniatrických 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_Kulhanek_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, 42)
- [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)

BIBLIOGRAPHY

- [25] 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. 5)
- [26] Rudolf Bayer and E. M. McCreight. Organization and maintenance of large ordered indexes. *Acta Informatica*, 1(3):173–189, 1972. [doi:10.1007/BF00288683](https://doi.org/10.1007/BF00288683). (cited on p. 6)
- [27] Rudolf Bayer. Symmetric binary B-Trees: Data structure and maintenance algorithms. *Acta Informatica*, 1(4):290–306, 1972. [doi:10.1007/BF00289509](https://doi.org/10.1007/BF00289509). (cited on p. 6)
- [28] 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)
- [29] Stephen A. Cook. An overview of computational complexity. *Communications of the ACM*, 26(6):400–408, June 1983. [doi:10.1145/358141.358144](https://doi.org/10.1145/358141.358144). (cited on p. 6)
- [30] 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, 7)
- [31] 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](https://doi.org/10.1145/800157.805047). (cited on p. 7)
- [32] Richard M. Karp. *Reducibility among combinatorial problems*. Springer US, 1972. [doi:10.1007/978-1-4684-2001-2__9](https://doi.org/10.1007/978-1-4684-2001-2__9). (cited on p. 7)
- [33] Richard E Korf. Depth-First Iterative-Deepening: An Optimal Admissible Tree Search. *Artificial Intelligence*, 27:97–109, 1985. (cited on p. 7)
- [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](https://doi.org/10.1146/annurev.cs.02.060187.002315). (cited on p. 7)
- [35] 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](https://doi.org/10.1051/0004-6361/201321529). (cited on p. 7)
- [36] CH Papadimitriou. *Computational complexity*. Addison-Wesley Publishing Company, 1995. (cited on p. 7)
- [37] Michael Sipser. *Introduction to the Theory of Computation*. 2012. (cited on p. 7)
- [38] 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. 8, 40)

BIBLIOGRAPHY

- [39] John L. Gustafson. Reevaluating Amdahl's law. *Communications of the ACM*, 31(5):532–533, May 1988. [doi:10.1145/42411.42415](https://doi.org/10.1145/42411.42415). (cited on p. 9)
- [40] David R. Butenhof. *Programming with POSIX Threads*. 1997. (cited on p. 12)
- [41] B Chapman, G Jost, and Ruud Van Der Pas. *Using OpenMP*, volume 10. MIT press, 2008. (cited on p. 12)
- [42] Peter S. Pacheco. *Parallel Programming with MPI*. Morgan Kaufmann, 1997. (cited on p. 12)
- [43] 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](https://doi.org/10.1109/TPDS.2011.308). (cited on p. 12)
- [44] Cleve Moler. Matrix computation on distributed memory multiprocessors. *Hypercube multiprocessors*, pages 181–195, 1986. (cited on p. 12)
- [45] Ian Foster. *Designing and Building Parallel Programs*. Addison-Wesley Publishing Company, 1995. (cited on p. 12)
- [46] Erick Cantu-Paz. *Designing efficient and accurate parallel genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 1999. (cited on p. 12)
- [47] PH Madden. Dispelling the myths of parallel computing. *IEEE Design & Test of Computers*, (April):58–64, 2012. [doi:10.1109/MDT.2012.2230391](https://doi.org/10.1109/MDT.2012.2230391). (cited on p. 12)
- [48] David Culer, Jaswinder Pal Singh, and Anoop Gupta. *Parallel Computer Architecture*. Morgan Kaufmann, 1998. (cited on p. 12)
- [49] Thomas Rauber and Gudula Rünger. *Parallel programming: For multicore and cluster systems*. Springer, 2013. (cited on p. 12)
- [50] Andrew S Tanenbaum and Maarten Van Steen. *Distributed Systems: Principles and Paradigms*, 2/E. 2007. (cited on p. 13)
- [51] Thomas Erl. *SOA Principles of Service Design*. Prentice Hall, 2008. (cited on p. 15)
- [52] Roy Thomas Fielding. Chapter 5: Representational State Transfer (REST). *Architectural Styles and the Design of Network-based Software Architectures, Dissertation*, 2000. (cited on p. 15)
- [53] Steve Vinoski. REST eye for the SOA guy. *IEEE Internet Computing*, 11:82–84, 2007. [doi:10.1109/MIC.2007.22](https://doi.org/10.1109/MIC.2007.22). (cited on p. 15)
- [54] Martin Fowler. *Patterns of Enterprise Application Architecture*. Pearson Education, Inc., 2003. (cited on p. 15)

BIBLIOGRAPHY

- [55] Jimmy Nilsson. *Applying Domain-Driven Design and Patterns*. Pearson Education, Inc., 2006. (cited on p. 15)
- [56] Gregor Hohpe and Bobby Woolf. *Enterprise Integration Patterns*. Pearson Education, Inc., 2002. (cited on p. 15)
- [57] 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. 16)
- [58] I Foster and C Kesselman. *The grid 2: blueprint for a new computing infrastructure*. Morgan Kaufmann, 2004. (cited on p. 16)
- [59] 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)
- [60] 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. 16)
- [61] 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. 16)
- [62] 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](https://doi.org/10.1145/288090.288111). (cited on p. 17)
- [63] 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](https://doi.org/10.1109/MC.2009.28). (cited on p. 17)
- [64] 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. 17)
- [65] 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](https://doi.org/10.1109/EDS.1990.138057). (cited on p. 17)
- [66] 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. 17)

BIBLIOGRAPHY

- [67] I. Foster and C. Kesselman. Globus: a Metacomputing Infrastructure Toolkit, 1997. [doi:10.1177/109434209701100205](https://doi.org/10.1177/109434209701100205). (cited on p. 17)
- [68] 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](https://doi.org/10.1002/cpe). (cited on p. 17)
- [69] 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](https://doi.org/10.1145/581571.581573). (cited on p. 17)
- [70] 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](https://doi.org/10.1109/GRID.2004.14). (cited on p. 18, 34)
- [71] 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](https://doi.org/10.1109/IPDPS.2007.370665). (cited on p. 18)
- [72] 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](https://doi.org/10.1007/s10723-009-9139-y). (cited on p. 18)
- [73] Gilles Fedak, Franck Cappello, and Universit 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](https://doi.org/10.1109/CCGRID.2001.923246). (cited on p. 18)
- [74] 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. 18)
- [75] N Høimyr, 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](https://doi.org/10.1088/1742-6596/396/3/032057). (cited on p. 18)
- [76] 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](https://doi.org/10.1109/eScience.2008.111). (cited on p. 18)

BIBLIOGRAPHY

- [77] 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](https://doi.org/10.1007/s10723-009-9137-0). (cited on p. 18)
- [78] 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](https://doi.org/10.1145/1165389.945462). (cited on p. 18)
- [79] Lamia Youseff, Rich Wolski, Brent Gorda, and Chandra Krintz. Paravirtualization for HPC Systems. *Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops*, pages 474–486, 2006. [doi:10.1007/11942634_49](https://doi.org/10.1007/11942634_49). (cited on p. 18)
- [80] Miroslav Ruda, Zdeněk Šustr, Jiří 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. 18)
- [81] 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](https://doi.org/10.1109/GCE.2008.4738445). (cited on p. 18, 20)
- [82] 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. 19)
- [83] Ashiq Anjum, Richard Hill, Richard McClatchey, Nik Besis, 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](https://doi.org/10.1504/IJWGS.2012.049169). (cited on p. 19)
- [84] 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](https://doi.org/10.1109/MIC.2011.44). (cited on p. 19)
- [85] 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. 19)
- [86] Sixto Ortiz. The problem with cloud-computing standardization. *Computer*, 44(July):13–16, 2011. [doi:10.1109/MC.2011.220](https://doi.org/10.1109/MC.2011.220). (cited on p. 19)
- [87] Georg Hager and Gerhard Wellein. *Introduction to High Performance Computing for Scientists and Engineers*. 2010. [doi:10.1201/EBK1439811924](https://doi.org/10.1201/EBK1439811924). (cited on p. 19)
- [88] John Levesque and Gene Wagenbreth. *High Performance Computing: Programming and Applications*. CRC Press, 2010. (cited on p. 19)

BIBLIOGRAPHY

- [89] 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](https://doi.org/10.1109/MTAGS.2008.4777912). (cited on p. 20)
- [90] Ioan Raicu. *Many-task computing: bridging the gap between high-throughput computing and high-performance computing*. PhD thesis, 2009. (cited on p. 20)
- [91] 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](https://doi.org/10.1007/s10586-010-0132-9). (cited on p. 20)
- [92] 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. 20)
- [93] 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](https://doi.org/10.1109/MIC.2005.56). (cited on p. 20)
- [94] 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](https://doi.org/10.1145/1084805.1084814). (cited on p. 20)
- [95] 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](https://arxiv.org/abs/0808.3545). (cited on p. 20)
- [96] 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](https://doi.org/10.1109/CIBEC.2008.4786077). (cited on p. 20)
- [97] 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](https://doi.org/10.1016/j.future.2008.02.008). (cited on p. 20)
- [98] Peter Kacsuk. P-GRADE portal family for grid infrastructures. *Concurrency Computation Practice and Experience*, 23:235–245, 2011. [doi:10.1002/cpe.1654](https://doi.org/10.1002/cpe.1654). (cited on p. 20, 34)
- [99] 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](https://doi.org/10.1007/s10723-013-9261-8). (cited on p. 20)

BIBLIOGRAPHY

- [100] 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. 20)
- [101] 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. 20)
- [102] 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. 20)
- [103] 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. 20)
- [104] Issac N. Bankman, editor. *Handbook of Medical Imaging; Processing and Analysis*. 2000. (cited on p. 24)
- [105] 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, 26, 43)
- [106] 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)
- [107] 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)
- [108] 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)

BIBLIOGRAPHY

- [109] 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, 43)
- [110] 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)
- [111] 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](https://doi.org/10.1007/s10723-007-9088-2). (cited on p. 25)
- [112] 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](https://doi.org/10.1016/j.future.2008.05.005). (cited on p. 25)
- [113] 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](https://doi.org/10.1016/j.future.2009.07.004). (cited on p. 25)
- [114] 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](https://doi.org/10.1088/1742-6596/119/8/082011). (cited on p. 25)
- [115] 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](https://doi.org/10.1109/TITB.2010.2049268). (cited on p. 25)
- [116] 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)
- [117] 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](https://doi.org/10.1007/978-0-387-72498-0_8). (cited on p. 25)
- [118] 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)

BIBLIOGRAPHY

- [119] 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](https://doi.org/10.1109/SC.2005.72). (cited on p. 26)
- [120] 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)
- [121] 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](https://doi.org/10.1109/JPROC.2004.840301). (cited on p. 27)
- [122] 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](https://doi.org/10.1016/S0892-1997(02)00070-X). (cited on p. 27)
- [123] 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)
- [124] 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](https://doi.org/10.1016/S0892-1997(96)80047-6). (cited on p. 27)
- [125] 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)
- [126] Marek Frič. Parametrický fonogram obecných řečových a hlasových projevů - ParVRP, 2007. URL: <http://zvuk.hamu.cz/vyzkum/dokumenty/TL12x.pdf>. (cited on p. 27, 43)
- [127] Marek Frič, Tomáš Kulhánek, and Jaroslav Hrb. Systém pro vzdáleně přístupnou analýzu hlasu RealVoiceLab, 2012. URL: <http://zvuk.hamu.cz/vyzkum/dokumenty/TL46x.pdf>. (cited on p. 27)
- [128] 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](https://doi.org/10.1038/clpt.2010.92). (cited on p. 28)

BIBLIOGRAPHY

- [129] 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](https://doi.org/10.1109/RBME.2009.2036204). (cited on p. 28)
- [130] 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](https://doi.org/10.1146/annurev.ph.34.030172.000305). (cited on p. 28)
- [131] 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](https://doi.org/10.1113/jphysiol.2010.201558). (cited on p. 29)
- [132] 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](https://doi.org/10.3389/fphys.2011.00012). (cited on p. 29)
- [133] Erik Butterworth, Bartholomew E. Jardine, Gary M. Raymond, Maxwell L. Neal, and James B. Bassingthwaigte. JSim, an open-source modeling system for data analysis. *F1000Research*, 2, May 2014. [doi:10.12688/f1000research.2-288.v2](https://doi.org/10.12688/f1000research.2-288.v2). (cited on p. 29)
- [134] 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](https://doi.org/10.1016/j.pbiomolbio.2004.02.006). (cited on p. 29)
- [135] 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](https://doi.org/10.1093/bioinformatics/btq723). (cited on p. 29)
- [136] 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. 29)
- [137] 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](https://doi.org/10.1093/nar/gkj092). (cited on p. 29)
- [138] 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)

BIBLIOGRAPHY

- [139] 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, 41)
- [140] 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.combiomed.2013.01.007](https://doi.org/10.1016/j.combiomed.2013.01.007). (cited on p. 30)
- [141] 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.combiomed.2014.01.006](https://doi.org/10.1016/j.combiomed.2014.01.006). (cited on p. 30)
- [142] 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. 30)
- [143] Michael Tiller. Modelica by Example, 2014. URL: <http://book.xogeny.com>. (cited on p. 30)
- [144] 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](https://doi.org/10.1177/154851290500200405). (cited on p. 30)
- [145] P Eykhoff. *Trends and progress in system identification*. IFAC series for graduates, research workers and practising engineers;vol. 1. Pergamon, Oxford, 1981. (cited on p. 30)
- [146] Michael C K Khoo. *Physiological control systems*. IEEE press, 2000. (cited on p. 31, 33)
- [147] 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](https://doi.org/10.1101/gr.1262503). (cited on p. 31)
- [148] 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)
- [149] 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)
- [150] 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](https://doi.org/10.1109/iCBEB.2012.62). (cited on p. 31)

BIBLIOGRAPHY

- [151] 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)
- [152] 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](https://doi.org/10.1109/eScience.2012.6404420). (cited on p. 31)
- [153] 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](https://doi.org/10.1029/2005WR004723). (cited on p. 31)
- [154] 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. 31)
- [155] 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. 32, 43)
- [156] 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](https://doi.org/10.1016/j.ress.2005.11.017). (cited on p. 33)
- [157] Andrea Saltelli, Stefano Tarantola, Francesca Campolongo, and Marco Ratto. *Sensitivity Analysis in Practice*. 2004. [doi:10.1002/0470870958](https://doi.org/10.1002/0470870958). (cited on p. 33)
- [158] 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](https://doi.org/10.1002/9780470725184). (cited on p. 33)
- [159] W van Meurs. *Modeling and Simulation in Biomedical Engineering: Applications in Cardiorespiratory Physiology*. McGraw-Hill Professional, 2011. (cited on p. 38, 39, 41)
- [160] 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](https://doi.org/10.3233/978-1-61499-054-3-19). (cited on p. 43)
- [161] 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. 45)

BIBLIOGRAPHY

- [162] 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](https://arxiv.org/abs/1312.5670), doi:10.1016/j.cub.2013.11.014. (cited on p. 45)
- [163] 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)
- [164] 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. 46)
- [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. 46)