MŰEGYETEM 1782

BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS
DEPARTMENT OF AUTOMATION AND APPLIED INFORMATICS

Attila Csaba Marosi

Önkéntes Számítási Rendszerek Kihívásai és Formális Aspektusai

---

Challenges and Formal Aspects of Volunteer Computing

Ph.D. Thesis Booklet

Thesis Advisors:

Péter Kacsuk
MTA SZTAKI, LPDS

Sándor Juhász
BME, AUT

Budapest, 2016.

MTA SZTAKI Hungarian Academy of Sciences
Institute for Computer Science and Control

Contents

1. Preliminaries and objectives

Compute capacity is either used or wasted. Unused capacity cannot be stored for later use. The term *cycle-scavenging* refers to (distributed) systems that utilize this idle computing cycles of connected computers.

Desktop Grids (DGs) offer no guarantees for service quality, they often utilize volatile, heterogeneous, non-dedicated resources for solving compute intensive tasks. DGs cycle-scavenge institutional desktop computers to solve compute intensive problems. On the other hand Volunteer Computing (VC) utilizes the idle resources (i.e., CPU cycles and storage) of private donated computers. These are typically home desktop computers behind firewalls and routers that can be considered even more heterogeneous and volatile. The inaccessibility and unavailability characteristics make running traditional HPC (High Performance Computing) workloads like tightly-coupled MPI (Message Passing Interface) and most HTC (High Throughput Computing) applications on volunteer computing unfeasible. Contrary to these MTC (Many Task Computing), especially *embarrassingly parallel* or parameter study [53, Section 1.4.4] problems where there is no dependency/ communication between jobs are well suited. In this case the same computation is performed using different input parameters that are usually achieved by partitioning a large parameter space.

Evidently VC resembles DGs whereas DGs are not fully equivalent to VC. However due to their similarities, they are often used as synonyms or in an interchanged meaning. I make an attempt in this thesis to formalize their characteristics and relationship. Formal modeling is applied that tries to grasp the semantic of their functionalities (opposed to comparisons based on properties or features). Further I use this modeling to formalize the Berkeley Open Infrastructure for Network Computing (BOINC) [50] VC system. The resulting models help categorizing existing DG/VC systems and aim to be a foundation for formalizing other middleware.

The second half of this thesis is dedicated to three – what I consider – major challenges and future directions of VC. First I propose a method that allows the collaboration (federation) of VC and DG projects.

Cloud Computing [48, 47] (CC) offers simple and cost effective outsourcing in dynamic service environments and allows the construction of service-based applications extensible with the latest achievements of diverse research areas, such as Grid Computing, Service-oriented computing, business processes and virtualization. CC is built using dedicated and reliable resources and provides uniform seemingly unlimited capacities. Volunteer Computing on the other hand uses volatile, heterogeneous and unreliable resources. As the second challenge (of VC in this thesis) I make an attempt starting from a definition for Cloud Computing to identify the required steps and formulate a definition for what can be considered as the next evolutionary stage for Volunteer Computing: Volunteer Cloud Computing (VCC). Among the many idiosyncrasies of VC to overcome (e.g., volatility, heterogeneity, reliability, responsiveness, scalability, etc.). Heterogeneity exists in VC at different levels. Resources can have a diverse set of hardware components (e.g., CPU type, features and speed; available memory, bandwidth and disk, etc.) and second they run a diverse set of software components (e.g., operating system, installed libraries, etc.). VC/DG middlewares can have their own set of requirements for applications in order to be compatible with them. Applications on their own have requirements (e.g., depend on specific software libraries and packages). The vision of Cloud Computing promises to provide a homogeneous environment. My goal is to identify methods and propose solutions that tackle the heterogeneities and thus, make a step towards Volunteer Clouds. (In this thesis the terms Volunteer Clouds and Volunteer Cloud Computing are used interchangeably.)

As mentioned previously VC is suited mainly for embarrassingly parallel or parameter study applications. These are usually run in form of batches where each batch represents an experiment. The total execution time of the batch from the submission until the finish of the last job is referred as *makespan*. The makespan of a batch is determined by the slowest (i.e., last finishing) job. The volatility and unreliability of the resources in VC can lead to prolonged makespan even if it is caused by a small fraction of jobs of a batch. This phenomena is referred as the "tail-effect" [45]. Tackling this is another step towards Volunteer Clouds. As the last challenge (of VC in this thesis) I am proposing algorithms that address prolonged makespan from different aspects derived from real world scenarios.

2. Methodological summary and results

The goal of my thesis is the investigation of DG and VC systems, focusing on formal model based comparison; federation and interoperability of different systems; tackling the heterogeneity of donated resources and systems; and reducing the so-called ”tail-effect” occurring during execution of batches of jobs on these systems. I present my results in form of formal models, definitions and algorithms. I elaborate the semantics of functionalities of DG/VC systems using the Abstract State Machines (ASM) method [51, 54]. ASM is a mathematically well founded framework for highlevel system design and analysis originally introduced as evolving algebras by Gurevich [54]. It allows hiding easily the non-important details at the high-level design phase by formulating the model on a conceptional level rather than based on implementation details and attributes. My models use an existing ASM model for grids [52, 49] as basis. I investigate the relations of the developed models and the base model using *input-output-* and *trace-equivalence*. I present my results for federation and volunteer clouds using models (however not ASM), also extending my previous ASM models where it is needed. I validated my models through the evaluation of their implementations. For these I used C/C++ programming language since all investigated systems support this to some extent (e.g., the BOINC API provides a client library). Properly written C/C++ code is cross-platform, it can be compiled for different operating systems. Due to a peculiarity of volunteer computing most of my implementations need to be able to run on different operating systems, without pre-installed libraries and execution environments (with the single exception of VirtualBox). I also present my algorithms for improving batch completion times using formal descriptions. I validated them with implementation and evaluation in real VC systems. For this I used the SQL and Python programming languages: due to a peculiarity of BOINC for the state-querying and assignments of jobs SQL was required, but I implemented the control logic using Python. I made measurements on EDGeS@home [42] and SZTAKI Desktop Grid [43] VC systems under real conditions. For the measurements I used my own logging routines. These were always instrumented versions of the original implementations. Since batch execution times are in the range of hours, and I assumed measurements are fine with seconds precision thus, I concluded the affect of instrumentation is negligible. However I confirmed this empirically. For the evaluation and visualisation of measurement results I used *The R Project for Statistical Computing* [44].

3. Novel scientific results

I summarize my research results in form of two thesis groups. In the first thesis group I classify and compare DG/VC systems using formal models that I've developed. The goal of my work presented here is to provide definitions and reference models for the two system archetypes (DG and VC). Using these it is possible to categorize existing systems and to develop formal models of existing systems for further usage. I start by using an informal aspect system to visualise the fundamental difference between DG and VC systems, and I also discuss existing systems based on these criteria. Next I use formal methods to represent DG/VC systems: contrary to informal methods, that compare usually based on a selected characteristic, with formal methods, I seize the semantics of the systems. To achieve this I start from an existing formal model for "traditional" grids (also known as service grids) [49, 52] which I reference as M_{GRID} in my dissertation. I evaluate desktop grids using a single formal model (see $M_{GROUND-DG}$ in Figure 3.1), and volunteer computing with two formal models (see $M_{VC-VOTE}$ and $M_{VC-SPOT}$ in Figure 3.1). I compare the models presenting their relation with each other and with the base model (M_{GRID}). My fourth model (see M_{BOINC} in Figure 3.1) represents a formal model of an existing system: BOINC. I start by formulating a criteria system with – what I consider – the most important attributes of BOINC. I show that the model correctly represents BOINC and is a correct refinement of the $M_{VC-VOTE}$ model. M_{BOINC} also aims to validate my previous models. Beyond this it serves as a template to formalize other existing DG/VC systems.

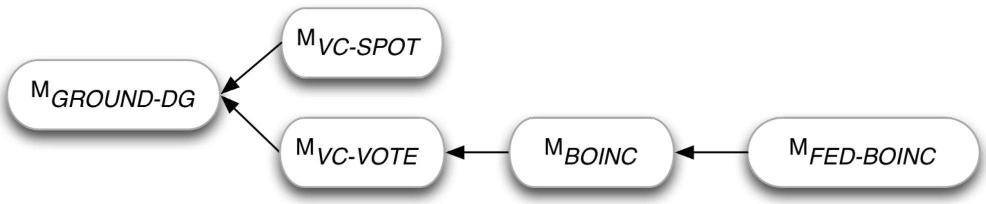

Figure 3.1: Proposed formal models and their relations.

In the second thesis group I investigate (*i*) the workload sharing between DG/VC systems; and part of the challenges of VCC I investigate (*ii*) the tackling of heterogeneity of resources and systems; and (*iii*) how to mitigate the tail effect caused by the volatility of volunteer resources. For workload sharing I use the M_{BOINC} model as this is the most detailed. The resulting model (see $M_{FED-BOINC}$ in Figure 3.1) contains a security mechanism, that extends the one of BOINC. I introduce a new application deployment mechanism that is built on this mechanism. This deployment mechanism is the foundation of workload sharing between systems. Additionally I identified three scenarios: (*i*) workload sharing between VC systems; (*ii*) hierarchical DG systems; and (*iii*) institutional resources supporting VC systems. From these I use the hierarchical DG systems scenario for presenting the process of the workload transfer.

I investigated the NIST definition for computing clouds [47]. Based on this I identify the challenges and limitations, that must be addressed for using volunteer resources to build CC services. In the remainder of my thesis I formulate a definition for VCC, and provide solutions for the identified challenges. The first identified challenge is the heterogeneity of resources and systems existing at different levels. I define different abstraction levels: (*i*) middleware abstraction satisfies dependencies between a given system and application; (*ii*) cross-middleware abstraction hides dependencies between systems; (*iii*) environment abstraction provides an uniform environment over the heterogeneous software environment of volunteer resources; and (*iv*) full abstraction provides an homogeneous environment over different operating systems and partially over heterogeneous hardware resources. I present the DC-API, GenWrapper and GBAC environments. DC-API hides the heterogeneity of DG/VC systems using the cross-middleware abstraction method. The GenWrapper environment I designed builds on DC-API (and thus, on cross-middleware abstraction), and middleware and environment abstraction, thus provides a middleware independent environment and successfully hides the dependencies between applications and systems. The GBAC environment I developed (relying on the defined cross-middleware, middleware, environment and full abstraction frameworks) provides an uniform software and hardware environment for executing jobs on volunteer resources. It also hides the differences between different VC/DG systems based on the abstraction levels, and satisfies the requirements between system and application. Based on the second indentified challenge, I present two environments and related algorithms to tackle the prolonged makespan caused by volatility of donated resources in VC systems. The first handles the VC system as a black-box (has no information about its internal workings). Extending the VC systems it monitors the state of the executed job batch, identifies ones that can cause delay, replicates them to ensure their successful execution. I demonstrated that using this method the makespan can be improved without serious extra-load for the VC system. The second environment and the corresponding algorithms extend the internal scheduler of BOINC by redirecting selected jobs from a batch to reliable resources. I demonstrated that selecting and redirecting specific jobs improves the makespan.

Next I formulate my theorems and list my corresponding publications.

Theorem Group I.

Comparision of and definitions for Desktop Grids and Volunteer Computing systems based on their formal models.

I developed an informal aspect system describing the main attributes of desktop grids and volunteer computing systems thus, it makes possible to classify existing systems. I evaluated several existing DG/VC systems based on this informal system and made attempt of categorizing them as DG or VC system. Next I used formal methods to represent DG/VC systems: contrary to informal methods, that compare usually based on a selected characteristic, with formal methods, I seize the semantics of the systems. To achieve this I start from an existing formal model for "traditional" grids (also known as service grids) [49, 52] which I reference as M_{GRID} .

Theorem I.1.: Formal model and definition for Desktop Grid
M_{GRID} model is an existing formal model for representing grid systems. Based on M_{GRID} I developed a formal model named $M_{GROUND-DG}$ for modelling desktop grids. I proved that systems represented by $M_{GROUND-DG}$ satisfy the requirements of M_{GRID}:

> **Theorem:** *$M_{GROUND-DG}$ is an extension of M_{GRID} and is a formal reference model for the architecture and application-management of desktop grids thus, enables the differentiation of desktop grids from the traditional grids represented by M_{GRID}.*

Based on this theorem, systems represented by model $M_{GROUND-DG}$ are also considered as grids based on M_{GRID}.

Related publications: [8], [22], [36] and [37].

Theorem I.2.: Formal model and definition for Volunteer Computing
Models $M_{VC-VOTE}$ and $M_{VC-SPOT}$ are formal models I developed for representing volunteer computing systems: $M_{VC-VOTE}$ represents systems with voting based result certification; and $M_{VC-SPOT}$ represents systems with spot-checking result certification. The models make it possible to evaluate existing systems whether they can be considered as desktop grids and/or volunteer computing systems:

> **Theorem:** *$M_{VC-VOTE}$ and $M_{VC-SPOT}$ model variants derived from $M_{GROUND-DG}$ represent formal models narrowed for representing volunteer computing systems that make possible the evaluation of computing infrastructures against volunteer computing systems.*

$M_{VC-VOTE}$ satisfies the requirements of model M_{GRID} since, it corresponds to the desktop grid definition by $M_{GROUND-DG}$. As a consequence systems modelled by $M_{VC-VOTE}$ are grids as per M_{GRID}, and desktop grids as per $M_{GROUND-DG}$, and correspond to the definition of desktop grids as per Theorem I.1. Similarly it can be shown for $M_{VC-SPOT}$ that it also satisfies the same requirements.

Related publications: [8], [22], [36] and [37].

Theorem I.3.: Formal model for BOINC, and its relation with model $M_{VC-VOTE}$
I created a criteria system that incorporates the most important aspects of BOINC. I demonstrated that the model M_{BOINC} I developed and is a formal model for BOINC

corresponds to these aspects. I demonstrated that M_{BOINC} is a correct refinement of $M_{VC-VOTE}$ thus, it represents a volunteer computing system, a desktop grid system and a grid system. I demonstrated that M_{BOINC} satisfies the requirements of Theorem I.2:

Theorem: *Model M_{BOINC} is a correct refinement of model $M_{VC-VOTE}$, further M_{BOINC} correctly represents the internal architecture and operation of BOINC thus, BOINC qualifies as a volunteer computing system, a desktop grid system and a grid system.*

Related publications: [2], [8], [22], [36] and [37].

Theorem Group II.

Workload sharing between Desktop Grids and Voluneer Computing Systems, and forming Volunteer Computing Clouds.

Desktop grids and volunteer computing systems operate isolated and independently from each other.

As an additional trait they provide jobs periodically thus, they are not able to utilize fully all their donated resources. These resources (if they are connected to multiple projects) will download tasks from others making the resource potentially unavailable for the project when it has work the next time. If these donated resources are not connected to other projects, then their compute capacity for the period is lost as compute capacity cannot be stored.

Volunteer computing systems collect heterogeneous resources with regard to capacity, configuration and volatility. Additionally each system has its own idiosyncrasies and requirements for operation and architecture of applications. Contrary to this cloud computing promises homogeneous resources and environments: homogeneous resources means processor architecture, available memory and disk. A Infrastructure as a Service (IaaS) cloud a client can request the required amount of hardware resources and the required number of (compute) instances. Homogeneous environment means the freely selectable operating systems, installed libraries and other components. Contrary to this VC systems collect donated resources with large variety of hardware and software architecture and configuration, and also the installed software libraries and components vary greatly.

Theorem II.1.: Workload sharing between Desktop Grids and Volunteer Computing Systems

The automatic application deployment method and security mechanism I developed as part of $M_{FED-BOINC}$ enables the transfer of tasks between projects of volunteer computing and desktop grid systems. I present the developed mechanisms via the BOINC volunteer computing system and one of the identified scenarios:

> **Theorem:** *The automatic application deployment method I developed makes the transfer of tasks between desktop grids and/or volunteer computing systems possible.*

Related publications: [4], [6], [9], [10], [11], [12], [14], [17], [33], [38], and [40].

Theorem II.2.: Volunteer Clouds
I defined different abstraction levels for volunteer computing systems: (*i*) middleware abstraction satisfies dependencies between a given system and application; (*ii*) cross-middleware abstraction hides dependencies between systems; (*iii*) environment abstraction provides an uniform environment over the heterogeneous software environment of volunteer resources; and (*iv*) full abstraction provides an homogeneous environment over different operating systems and partially over heterogeneous hardware resources. Based on these I developed the DC-API, GenWrapper and GBAC environments.

The GBAC environment I developed – that builds on the defined abstraction levels (see i-iv) – provides a homogeneous software and hardware environment for executing tasks on volunteer resources. It hides the requirement-differences of different desktop grid and volunteer computing systems; and satisfies the system-requirements that are present between applications and VC systems:

> **Theorem:** *The GBAC environment provides a homogeneous software and hardware environment for executing work units on resources of volunteer computing systems, and provides the fundamentals for the virtualized resource handling of a new computing infrastructure called Volunteer Computing Clouds.*

Related publications: [5], [7], [12], [14] [16], [18], [19], [20], [21], [23], [24], [25], [27], [28], [29], [30], [31], [32], [34], [35], [39] and [41].

Theorem II.3.: Improving makespan on Volunteer Computing Systems
I developed work unit handling algorithms that reduce the known "long-tail" effect for makespan in volunteer computing systems:

> **Theorem:** *Algorithms "Adapt redundancy", "Prioritize delayed jobs" and "Prioritize job sources" mitigate the negative footprint of the "tail-effect" for makespan on volunteer computing systems under the predefined conditions.*

I demonstrated with measurements that the algorithms successfully reduce makespan in real-world scenarios: deployed at SZTAKI Desktop Grid and EDGeS@home volunteer computing projects. The improvement and effectiveness of the algorithms depends on certain criteria that are detailed in my thesis.

Related publications: [1], [3], [13], [15] and [26].

4. Practical applicability of the scientific results

My formal models for modelling DG/VC systems allow to model – similarily to BOINC – other systems and evaluating and classifying existing ones.

The C/C++ implementation of the automatic application deployment and the introduced security mechanism is part of BOINC since 2008. The implementation that makes the interoperability of DG/VC systems possible was utilized by the HAGRID R&D project[1]: It connected the different DG/VC systems of the project consortium members accelerating the execution of a meterological application [36].

I defined – on basis of Thesis II.2 – the DC-API/Genwrapper and GBAC environments. The implementation of DC-API is part of BOINC extending the official BOINC API. The final merging is in progress. Implementations of GBAC and GenWrapper are available as separate open source projects[2, 3]. GBAC is going to be part of BOINC VBoxWrapper[4], its merging is in progress. Currently there are 58 scientific applications that are adapted for BOINC and XtremWeb using DC-API/ GenWrapper and GBAC. The list of applications and their short descriptions can be found in the Appendix of my thesis. I want to emphasize one application in particular from the end of the list (number 55): "*patterncount*". This application was adapted using GenWrapper, and is a fundamental part of a Ph.D. dissertation submitted and accepted at Brunel University [46].

The following Hungarian and European R&D projects utilized my results by providing scientific applications and supporting the further development of the DC-API/ GenWrapper and GBAC systems: EDGeS[5], EDGI[6], DEGISCO[7], CancerGrid[8], IDGF-SP[9], BioVeL[10], WEB2GRID[11] and AGRATéR[12].

The implementations of the algorithms defined in Thesis II.3 are running on different DG/VC systems operated by MTA SZTAKI: SZTAKI Desktop Grid [43] and EDGeS@home [42].

[1] `http://www.sztaki.hu/tudomany/projektek/projekt_informaciok/?uid=00188`, access date: 2013-01-01
[2] `http://genwrapper.sourceforge.net`, access date: 2014-05-26
[3] `http://gbac.sourceforge.net`, access date: 2014-05-26
[4] `http://boinc.berkeley.edu/trac/wiki/VboxApps`, access date: 2014-10-09
[5] `http://edges-grid.eu/`, access date: 2013-01-01
[6] `http://edgi-project.eu/`, access date: 2013-01-01
[7] `http://degisco.eu/`, access date: 2013-01-01
[8] `http://cancergrid.eu/`, access date: 2009-01-01
[9] `http://idgf-sp.eu/`, access date: 2014-05-26
[10] `http://www.biovel.eu/`, access date: 2016-02-07
[11] `http://www.egroup.hu/main/en/research/web2grid`, access date: 2014-05-26
[12] `http://agrater.hu/`, access date: 2016-02-07

Bibliography

Publications - Journal papers

[1] Jozsef Kovacs, Attila Csaba Marosi, Adam Visegradi, Zoltan Farkas, Peter Kacsuk, and Robert Lovas. "Boosting gLite with cloud augmented volunteer computing". In: *Futur. Gener. Comput. Syst.* 43-44 (2015), pp. 12–23. ISSN: 0167739X. DOI: 10.1016/j.future.2014.10.005. URL: http://www.sciencedirect.com/science/article/pii/S0167739X14001897. **(Impact Factor: 2.786)**

[2] Attila Csaba Marosi and Róbert Lovas. "Defining volunteer computing: a formal approach". In: *Comput. Res. Model.* 7.3 (2015), pp. 565–571. ISSN: 2076-7633. URL: http://eprints.sztaki.hu/8361/%20http://crm-en.ics.org.ru/journal/issue/167/.

[3] Máté Pataki and Attila Csaba Marosi. "Searching for Translated Plagiarism with the Help of Desktop Grids". In: *J. Grid Comput.* 11.1 (2013), pp. 149–166. ISSN: 1570-7873. DOI: 10.1007/s10723-012-9224-5. URL: http://dx.doi.org/10.1007/s10723-012-9224-5. **(Impact Factor: 1.667)**

[4] Miklós Kozlovszky, Krisztián Karoczkai, István Marton, Ákos Balasko, Attila Csaba Marosi, and Péter Kacsuk. "Enabling generic distributed computing infrastructure compatibility for workflow management systems". In: *Comput. Sci.* 13.3 (2012), pp. 61–78.

[5] Attila Marosi, József Kovács, and Peter Kacsuk. "Towards a volunteer cloud system". In: *Futur. Gener. Comput. Syst.* (2012), pages. ISSN: 0167-739X. DOI: 10.1016/j.future.2012.03.013. URL: http://www.sciencedirect.com/science/article/pii/S0167739X12000660. **(Impact Factor: 1.864)**

[6] P. Kacsuk, J. Kovacs, Z. Farkas, Attila Csaba Marosi, and Z. Balaton. "Towards a Powerful European DCI Based on Desktop Grids". In: *Journal of Grid Computing* 9 (2 2011). 10.1007/s10723-011-9186-z, pp. 219–239. ISSN: 1570-7873. URL: http://dx.doi.org/10.1007/s10723-011-9186-z. **(Impact Factor: 1.31)**

[7] Attila Csaba Marosi, Jozsef Kovacs, and Peter Kacsuk. "Utilizing the power of Desktop Grid systems by Web 2.0 communities". In: *Infocommunications Journal* 3.2 (2011), pp. 1–8. URL: http://eprints.sztaki.hu/6731/.

[8] 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". English. In: *Journal of Grid Computing* 7 (4 2009), pp. 439–461. ISSN: 1570-7873. DOI: 10.1007/s10723-009-9139-y. URL: http://dx.doi.org/10.1007/s10723-009-9139-y.

[9] Etienne Urbah, Peter Kacsuk, Zoltan Farkas, Gilles Fedak, Gabor Kecskemeti, Oleg Lodygensky, Attila Marosi, Zoltan Balaton, Gabriel Caillat, Gabor Gombas, et al. "Edges: Bridging EGEE to BOINC and XtremWeb". In: *J. Grid Comput.* 7.3 (2009), pp. 335–354.

[10] Zoltán Balaton, Zoltán Farkas, Gábor Gombás, Péter Kacsuk, Róbert Lovas, Attila Csaba Marosi, Ad Emmen, Gábor Terstyánszky, Tamás Kiss, Ian Kelley, Ian Taylor, Oleg Lodygensky, Miguel Cardenas-Montes, Gilles Fedak, and Filipe Araujo. "EDGeS: The Common Boundary Between Service and Desktop Grids". In: *Parallel Process. Lett.* 18.3 (Sept. 2008), pp. 433–445.

[11] Gábor Gombás, Attila Csaba Marosi, and Zoltán Balaton. "Grid Application Monitoring and Debugging Using the Mercury Monitoring System". In: Lecture Notes in Computer Science 3470 (2005). Ed. by Peter M.A. Sloot, Alfons G. Hoekstra, Thierry Priol, Alexander Reinefeld, and Marian Bubak, pp. 193–199. DOI: 10.1007/11508380_21. URL: http://dx.doi.org/10.1007/11508380%7B%5C_%7D21. **(Impact Factor: 0.302)**

Publications - Book chapters

[12] Attila Csaba Marosi and Peter Kacsuk. "Volunteer Clouds: From Volunteer Computing to interconnected infrastructures". In: *Dev. Interoper. Fed. Cloud Archit.* Ed. by Gabor Kecskemeti, Attila Kertesz, and Zsolt Nemeth. IGI Global, 2016, (ACCEPTED). ISBN: 9781522501534. DOI: 10.4018/978-1-5225-0153-4. URL: http://www.igi-global.com/book/developing-interoperable-federated-cloud-architecture/142192.

[13] Péter Kacsuk, Zoltán Farkas, József Kovács, Ádám Visegrádi, Attila Csaba Marosi, Róbert Lovas, Gabor Kecskemeti, Zsolt Nemeth, and Mark Gergely. "Desktop Grid in the Era of Cloud Computing". In: *Grid Comput. Tech. Futur. Prospect.* Ed. by Jorge G. Barbosa and Inês Dutra. Nova Science Publishers, 2015, pp. 187–206. ISBN: 978-1-63482-326-5.

[14] Peter Kacsuk, Attila Marosi, Lovas Robert, and Jozsef Kovacs. "Supporting Web 2.0 Communities by Volunteer Desktop Grids". In: *Deskt. Grid Comput.* Ed. by Gilles Fedak. Chapman and Hall/CRC, June 2012, pp. 287–307. ISBN: 978-1-4398-6214-8. DOI: 10.1201/b12206-16. URL: http://www.crcnetbase.com/doi/abs/10.1201/b12206-16.

[15] P. Kacsuk, Marosi Attila Csaba, M. Kozlovszky, S. Acs, and Z. Farkas. "Parameter Sweep Job Submission to Clouds". In: *Grids, Clouds and Virtualization.* Ed. by Massimo Cafaro and Giovanni Aloisio. Computer Communications and Networks. 10.1007/978-0-85729-049-6_6. Springer London, 2011, pp. 123–141. ISBN: 978-0-85729-049-6. URL: http://dx.doi.org/10.1007/978-0-85729-049-6_6.

[16] G. Kecskemeti, A. Kertesz, Attila Csaba Marosi, and P. Kacsuk. “Interoperable Resource Management for establishing Federated Clouds”. In: *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice*. Ed. by M. Villari, I. Brandic, and F. Tusa. IGI Global (USA), 2011, pp. 18–35. DOI: doi:10.4018/978-1-4666-1631-8.ch002.

[17] Zoltan Farkas, Attila Csaba Marosi, and Peter Kacsuk. “Job Scheduling in Hierarchical Desktop Grids”. English. In: *Remote Instrumentation and Virtual Laboratories*. Ed. by Franco Davoli, Norbert Meyer, Roberto Pugliese, and Sandro Zappatore. Springer US, 2010, pp. 79–97. ISBN: 978-1-4419-5595-1. DOI: 10.1007/978-1-4419-5597-5_8. URL: http://dx.doi.org/10.1007/978-1-4419-5597-5_8.

[18] Attila Csaba Marosi, Zoltan Balaton, Peter Kacsuk, and Daniel Drotos. “SZTAKI Desktop Grid: Adapting Clusters for Desktop Grids”. In: *Remote Instrumentation and Virtual Laboratories*. Ed. by Franco Davoli, Roberto Pugliese, Norbert Meyer, and Sandro Zappatore. 10.1007/978-1-4419-5597-5_12. Springer US, 2010, pp. 133–144. ISBN: 978-1-4419-5597-5. URL: http://dx.doi.org/10.1007/978-1-4419-5597-5_12.

[19] Attila Csaba Marosi, Gabor Gombas, Zoltan Balaton, and Peter Kacsuk. “Enabling Java applications for BOINC with DC-API”. In: *Distributed and Parallel Systems*. Ed. by Péter Kacsuk, Róbert Lovas, and Zsolt Németh. 10.1007/978-0-387-79448-8_1. Springer US, 2008, pp. 3–12. ISBN: 978-0-387-79448-8. URL: http://dx.doi.org/10.1007/978-0-387-79448-8_1.

Publications - Conference papers

[20] József Kovács, Géza Ódor, Attila Marosi, and Ádám Kornafeld. “Exploring University Classes in Nonequilibrium Systems on SZTAKI Desktop Grid”. In: *Third Almer. Deskt. Exp. Work.* Almere: AlmereGrid, pp. 13–16.

[21] Ferenc Horváth, Péter Ittzés, Dóra Ittzés, Zoltán Barcza, Laura Dobor, Dóra Hidy, and Attila Csaba Marosi. “Supporting environmental modelling with Taverna workflows, web services and desktop grid technology”. In: *Proc. 7th Int. Congr. Environ. Model. Softw.* Ed. by D.P. Ames, N.W.T. Quinn, and A.E. Rizzoli. San Diego, 2014, pp. 1–7. URL: http://www.iemss.org/sites/iemss2014/papers/iemss2014%7B%5C_%7Dsubmission%7B%5C_%7D220.pdf.

[22] Attila Csaba Marosi and Zsolt Nemeth. “Two Sides of a Coin: Formalizing Volunteer and Desktop Grid Computing”. In: *Proc. Cracow Grid Work. 2013*. Ed. by M. Bubak, M. Turała, and K. Wiatr. Kraków: ACK CYFRONET AGH, 2013, pp. 69–70. ISBN: 978-83-61433-08-8.

[23] A. Kertesz, G. Kecskemeti, M. Oriol, A. Marosi, X. Franch, and J. Marco. “Integrated Monitoring Approach for Seamless Service Provisioning in Federated Clouds”. In: *In proc. of the 20th Euromicro International Conference on Parallel, Distributed and Network-Based Computing (PDP’12)*. IEEE CS, 2012, pp. 567–574.

[24] Sandor Acs, Miklos Kozlovszky, Attila Csaba Marosi, and Zoltan Balaton. "Amazon EC2 infrastruktúrán futó dinamikusan skálázható PBS klaszter". In: *Networkshop 2011. Kaposvár, 2011.* Kaposvár: NIIF, Apr. 2011, pp. 1–11. URL: http://eprints.sztaki.hu/6629/.

[25] Attila Csaba Marosi. "Web2Grid: Desktop Grid a Web 2.0 szolgálatában". In: *Networkshop 2011. Kaposvár, 2011.* 2011, pp. 1–7. URL: http://eprints.sztaki.hu/6719/.

[26] Attila Csaba Marosi and P. Kacsuk. "Workers in the Clouds". In: *Parallel, Distributed and Network-Based Processing (PDP), 2011 19th Euromicro International Conference on.* Feb. 2011, pp. 519–526. DOI: 10.1109/PDP.2011.79.

[27] Attila Csaba Marosi, Gabor Kecskemeti, Attila Kertesz, and Peter Kacsuk. "FCM: an Architecture for Integrating IaaS Cloud Systems". In: *In proc. of the Second International Conference on Cloud Computing, GRIDs, and Virtualization (Cloud Computing 2011).* 2011, pages.

[28] Jozsef Kovacs, Geza Odor, Attila Csaba Marosi, and Adam Kornafeld. "Exploring University Classes in Nonequilibrium Systems on SZTAKI Desktop Grid". In: *Enter the Grid: Proceeding of the Third AlmereGrid Desktop Experience workshop.* AlmereGrid.nl, 2010, pp. 13–17.

[29] Attila Csaba Marosi, Akos Balasko, and Livia Kacsukne Bruckner. "Porting the SEE-GRID EMMIL application into EDGeS infrastructure". In: *Enter the Grid: Proceeding of the Third AlmereGrid Desktop Experience workshop.* AlmereGrid.nl, 2010, pp. 33–37.

[30] Attila Csaba Marosi, Peter Kacsuk, Gilles Fedak, and Oleg Lodygensky. "Sandboxing for Desktop Grids Using Virtualization". In: *Proceedings of the 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing.* PDP '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 559–566. ISBN: 978-0-7695-3939-3. DOI: http://dx.doi.org/10.1109/PDP.2010.90. URL: http://dx.doi.org/10.1109/PDP.2010.90.

[31] Attila Csaba Marosi, Attila Kovács, Péter Burcsi, and Ádám Kornafeld. "The BinSys application on the EDGeS infrastructure". In: *Enter the Grid: Proceeding of the Third AlmereGrid Desktop Experience workshop.* AlmereGrid.nl, 2010, pp. 38–43.

[32] Alejandro Rivero, Andres Bustos, Attila Csaba Marosi, Darrio Ferrer, and F. Serrano. "ISDEP, a fusion application deployed in the EDGeS project". In: *Enter the Grid: Proceeding of the Third AlmereGrid Desktop Experience workshop.* AlmereGrid.nl, 2010, pp. 18–22.

[33] Filipe Araujo, David Santiago, Diogo Ferreira, Jorge Farinha, Patricio Domingues, L Moura Silva, Etienne Urbah, Oleg Lodygensky, Haiwu He, Attila Csaba Marosi, et al. "Monitoring the EDGeS project infrastructure". In: *Parallel Distrib. Process. 2009. IPDPS 2009. IEEE Int. Symp.* IEEE. 2009, pp. 1–8.

[34] Attila Csaba Marosi, Zoltan Balaton, and Peter Kacsuk. "GenWrapper: A generic wrapper for running legacy applications on desktop grids". In: *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*. May 2009, pp. 1–6. DOI: 10.1109/IPDPS.2009.5161136.

[35] G Fedak, Haiwu He, O Lodygensky, Z Balaton, Z Farkas, G Gombas, P Kacsuk, R Lovas, A C Marosi, I Kelley, I Taylor, G Terstyanszky, T Kiss, M Cardenas-Montes, A Emmen, and F Araujo. "EDGeS: A Bridge between Desktop Grids and Service Grids". In: *ChinaGrid Annu. Conf. 2008. ChinaGrid '08. Third*. Aug. 2008, pp. 3–9. DOI: 10.1109/ChinaGrid.2008.44.

[36] Zoltan Balaton, Gabor Gombas, Peter Kacsuk, Adam Kornafeld, Attila Csaba Marosi, Gabor Vida, Norbert Podhorszki, and Tamas Kiss. "SZTAKI Desktop Grid: a Modular and Scalable Way of Building Large Computing Grids". In: *Workshop on Large-Scale and Volatile Desktop Grids, PCGrid 2007*. 2007, pp. 1–8.

[37] Csaba Attila Marosi, Gabor Gombas, Zoltan Balaton, Peter Kacsuk, and Tamas Kiss. "SZTAKI Desktop Grid: Building a Scalable, Secure Platform for Desktop Grid Computing". In: *CoreGRID Workshop - Making Grids Work*. Ed. by Marco Danelutto, Paraskevi Fragopoulou, and Vladimir Getov. Springer, 2007, pp. 365–376. ISBN: 978-0-387-78447-2.

[38] Peter Kacsuk, Attila Csaba Marosi, Jozsef Kovacs, Zoltan Balaton, Gabor Gombas, Gabor Vida, and Adam Kornafeld. "SZTAKI Desktop Grid: a Hierarchical Desktop Grid System". In: *Proceedings of the Cracow '06 Grid Workshop*. 2006.

[39] Adam Kornafeld, Kovacs Attila, Burcsi Peter, Norbert Podhorszki, Attila Csaba Marosi, Gabor Vida, and Gabor Gombas. "Szuperszámítógépes Teljesítmény Szuperszámítógép Nélkül - A Binsys Projekt". In: *Networkshop 2006*. Miskolc: NIIFI, 2006.

[40] A. Cs. Marosi, G. Gombas, and Z. Balaton. "Secure Application Deployment in the Hierarchical Local Desktop Grid". In: *Proc. of DAPSYS 2006 6th Austrian-Hungarian Workshop on Distributed and Parallel Systems*. Innsbruck, Austria, Sept. 2006, pp. 145–153.

Publications - Others

[41] Attila Csaba Marosi, Peter Kacsuk, Gilles Fedak, and Oleg Lodygensky. *Using Virtual Machines in Desktop Grid Clients for Application Sandboxing*. Tech. rep. TR-0140. Institute on Architectural Issues: Scalability, Dependability, Adaptability, CoreGRID - Network of Excellence, Aug. 2008. URL: http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0140.pdf.

Publications - Summary

	Number of publications	H-Index	Independent citations
MTMT[13]	39	6	101
Scopus[14]	25	7	148
Web of Science[15]	12	3	34
Google Scholar[16]	40	13	452 (Full)

References

[42] *EDGeS@home. http://home.edges-grid.eu/*. URL: http://home.edges-grid.eu/ (visited on 01/01/2013).

[43] *SZTAKI Desktop Grid. http://szdg.lpds.sztaki.hu/szdg*. URL: http://szdg.lpds.sztaki.hu/szdg (visited on 01/01/2014).

[44] *The R Project for Statistical Computing. http://www.r-project.org/*. URL: http://www.r-project.org/ (visited on 05/23/2013).

[45] Stephen C Winter, Christopher J Reynolds, Tamas Kiss, Gabor Z Terstyanszky, Pamela Greenwell, Sharron McEldowney, Sandor Acs, and Peter Kacsuk. "Buttressing volatile desktop grids with cloud resources within a reconfigurable environment service for workflow orchestration". en. In: *J. Cloud Comput. Adv. Syst. Appl.* 3.1 (Jan. 2014), p. 1. ISSN: 2192-113X. DOI: 10.1186/2192-113X-3-1. URL: http://www.journalofcloudcomputing.com/content/3/1/1.

[46] Mohammadmersad Ghorbani. "Computational analysis of CpG site DNA methylation". PhD thesis. Brunel University, 2013. URL: http://dspace.brunel.ac.uk/bitstream/2438/8217/1/FulltextThesis.pdf.

[47] Peter Mell and Timothy Grance. *The NIST Definition of Cloud Computing*. Tech. rep. 800-145. Gaithersburg, MD: National Institute of Standards and Technology (NIST), Sept. 2011. URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[48] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility". In: *Futur. Gener. Comput. Syst.* 25.6 (June 2009), pp. 599–616. ISSN: 0167739X. DOI: 10.1016/j.future.2008.12.001. URL: http://linkinghub.elsevier.com/retrieve/pii/S0167739X08001957.

[49] Attila Kertész and Zsolt Németh. "Formal Aspects of Grid Brokering". In: *Electron. Proc. Theor. Comput. Sci.* 14 (Dec. 2009), pp. 18–31. ISSN: 2075-2180. DOI: 10.4204/EPTCS.14.2. URL: http://arxiv.org/abs/0912.2549v1.

[13] https://vm.mtmt.hu//search/slist.php?lang=0&AuthorID=10017749

[14] http://www.scopus.com/authid/detail.url?authorId=55664653500

[15] https://apps.webofknowledge.com/CitationReport.do?product=UA&search_mode=CitationReport&SID=Z1BiRedaaVOwnrbbDda&page=1&cr_pqid=3&viewType=summary

[16] https://scholar.google.hu/citations?user=PG3OFrUAAAAJ&hl=hu

[50] D P Anderson, C Christensen, and B Allen. "Designing a Runtime System for Volunteer Computing". In: *SC 2006 Conf. Proc. ACM/IEEE*. Nov. 2006, p. 33. DOI: 10.1109/SC.2006.24.

[51] E Borger and Robert F Stark. *Abstract State Machines: A Method for High-Level System Design and Analysis*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003. ISBN: 3540007024.

[52] Z Németh and V Sunderam. "Characterizing grids: Attributes, definitions, and formalisms". In: *J. Grid Comput.* (2003), pp. 9–23. URL: http://link.springer.com/article/10.1023/A:1024011025052.

[53] Ian Foster. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Boston: Addison-Wesley Longman Publishing Co., Inc., 1995. ISBN: 0201575949. URL: http://dl.acm.org/citation.cfm?id=527029%20http://www.mcs.anl.gov/%7B~%7Ditf/dbpp/text/node10.html%7B%5C#%7DSECTION02244000000000000000.

[54] Yuri Gurevich. "Evolving algebras: An attempt to discover semantics". In: *Curr. Trends Theor. Comput. Sci.* (1993), pp. 1–27. URL: http://books.google.com/books?hl=en%7B%5C&%7Dlr=%7B%5C&%7Did=kVhZDTKYa8MC%7B%5C&%7Doi=fnd%7B%5C&%7Dpg=PA266%7B%5C&%7Ddq=EVOLVING+ALGEBRAS+:+AN+ATTEMPT+TO+DISCOVER+SEMANTICS%7B%5C&%7Dots=icN09xFApG%7B%5C&%7Dsig=GVMXAgYnyS6bvIFOweICHuXYAYM.

Nomenclature

API Application Programming Interface

CC Cloud Computing

DC-API Distributed Computing API

DG Desktop Grid

GBAC Generic BOINC Application Client

NIST National Institute of Standards and Technology

SG Service Grid

VC Volunteer Computing

VCC Volunteer Cloud Computing