Transactions of the SDPS:
Journal of Integrated Design and Process Science
18 (1), 2014, 1-4
DOI 10.3233/jid-2014-0005
http://www.sdpsnet.org

EXTENDED EDITORIAL

# Transdisciplinary Challenges of Scientific Cloud Computing

Daniel Versick[a]* and Peter Tröger[b]

[a] *University of Rostock, Germany*
[b] *Hasso Plattner Institute, University of Potsdam, Germany*

Cloud Computing implements the next generation distributed system architecture by offering on-demand compute and storage resources for arbitrary customers in a scalable and self-manageable fashion (Mell & Grance, 2011). Information Technology (IT) resources, platforms and services are made available at virtually unlimited scale for everybody, everywhere, and anytime. Cloud Computing therefore not only leads to an outsourcing of enterprise IT services, it furthermore fosters a breathing IT infrastructure which adapts itself to the current size and demand of any organization. Thus, it supports the trend of IT to higher automation, smaller costs, and increased service levels (Gibson & Kasravi, 2012).

The recent trend for Cloud Computing led to a variety of new commercial providers and to a wide industrial uptake of the according technology. During the last years, the market was subject to a tremendous growth of up to 36%, the size is estimated to be 19.5 billion USD in 2016 (Columbus, 2013). It is undebatable that, even in the light of unsolved data protection and trust issues, Cloud Computing made its way into the default tool chain of modern IT-based environments.

The NIST reference architecture specification (Liu et al., 2012) defines five major actors in Cloud Computing: *cloud consumer*, *cloud provider*, *cloud carrier*, *cloud auditor* and *cloud broker*. For most scenarios, it is sufficient to focus on the relationship between provider and consumer. Both have a particular balance in their shared control of the elastic cloud resources, expressed by three well-known service models *infrastructure-as-a-service (IaaS)*, *platform-as-a-service (PaaS)*, and *software-as-a-service (SaaS)*.

As orthogonal aspect to the service model, cloud infrastructures can follow the *public cloud* or *private cloud* deployment model. Hybrid setups may also exist, were private cloud installations perform dynamic offloading to public cloud offerings in case of scalability issues (Sotomayor, Montero, Llorente, & Foster, 2009).

On the lowest service level, the IaaS-based cloud service provisioning, the consumer gets maximum control of the cloud resources by the means of virtualization. Cloud systems in this category consist of virtualized hardware resources at the provider side, managed through web interfaces or remote APIs by the cloud consumer. The provisioning architecture has to be flexible and highly scalable, virtual resources are directly and dynamically instantiated, migrated, checkpointed, and destroyed by the consumers. The provider has the responsibility to manage the according hardware and low-level software scalability, but leaves the application and operating system responsibility completely to the user. Popular examples are the Amazon Elastic Compute Cloud (EC2) and the Microsoft Azure environment.

*Corresponding author. Email: daniel.versick@uni-rostock.de Tel: (+49)381-4987556.

Published by IOS Press

The next higher level of provider control is the PaaS model, were cloud providers allow the upload and execution of packaged consumer applications. Popular examples here are Heroku and Google AppEngine. PaaS offers a higher level of abstraction for the price of flexibility, since execution platform constraints must be considered by the consumer. On the other hand, significant software maintenance and execution environment efforts are handled by the cloud provider.

The highest level of provider control is in the SaaS delivery model, were a cloud provider offers tenant-enabled specialized software for remote usage. Here, the user has to live with the given amount of functionality, but gets a complete decoupling from any IT operation aspect in exchange. The most common example for SaaS is Salesforce, but companies such as SAP or Microsoft more and more transform their core software products into such cloud offerings.

The architecture and design of Cloud Computing systems is a cross-topic challenge that incorporates questions of computer architecture, operating systems design, human computer interaction as well as distributed computing. Since Cloud environments help to gain access to specialized software, dedicated computational resources, or new storage facilities, they are not only important for enterprises with varying resource demands. They become also increasingly interesting for scientific institutions and their scientific workloads, even if they never touched computation and storage outsourcing beforehand. The term *scientific cloud computing* therefore relates to a special class of cloud consumers. It is a steadily emerging field, due to the fact that resources at virtually unlimited scale, only restricted by costs and programmability aspects, are now available for arbitrary sciences. This aligns to another important trend, the increasing reliance on computational algorithms for solving challenging research problems in all sciences. Research fields like e.g. visualization of health care and medical data (Ng & Wei, 2011; Ochs, Geller, & Perl, 2011) or VLSI design rely on huge amounts of data or need tremendous computing power (Nikolaos G. Bourbakis, 2000).

Specific scientific fields already have a long-standing tradition in using remote computational capabilities. The most prominent example is the *high-performance computing (HPC)* community from natural sciences such as physics, biology, geography, medical or climate research (Vecchiola, Pandey, & Buyya, 2009). Traditional HPC can be interpreted as some kind of PaaS approach, were users develop specialized parallel computing applications and submit them for execution using provider-managed cluster frameworks. The *Grid Computing* trend extended this approach by supporting federated data centers, although for the users nothing dramatically changed in the usage pattern (Foster, Zhao, Raicu, & Lu, 2008).

Scientific Cloud Computing now opens new opportunities for any kind of research work that relies on computational or data analysis. One obvious way of implementing this is another application of the HPC paradigms on cloud resources (Marathe et al., 2013; Stein, 2010). In this scenario, cloud resources are, again, simply treated as execution resources for scientific parallel workload. The problem lies in the fact that this creates an increasingly critical ḿappingṕroblem, were scientific applications with their specialized needs meet IT infrastructures with their special requirements on applications. For this reason, the more interesting trend is the shift towards SaaS-based scientific cloud offerings, such as Wolfram Alpha. This approach decouples researchers from IT-specific issues and allows them to focus on their particular problem domain.

When a scientific cloud offering is implemented by providers, one crucial aspect becomes cost control and efficiency metrics, especially with the restricted budgets of their users. When scientific users want to rely on public cloud offerings, the question of trustworthiness and data protection (e.g. in medical research) becomes also very relevant. Scientific cloud users may even come up with completely new application scenarios, such as the utilization of clouds for novel teaching methods. For these scenarios, new resource brokerage and data handling approach are needed inside the cloud provisioning infrastructure.

The given set of challenges leads to the fact that scientific cloud infrastructures need to consider specific needs of their user communities, instead of forcing the scientific application to target the specifics of one Cloud runtime environment. This makes programming models (Lämmel, 2008), standardized APIs (Tröger & Merzky, 2013) and novel distributed infrastructures an interesting topic again. Another insight is that the mapping problem for scientific applications is inherently a transdisciplinary challenge. The cloud adoption schemes for procedures, numerical methods and best practices being valid in one scientific domain should be transferred to another one by minimized adaption effort. For this reason, there is a desperate need for a holistic view on the problem domain. Most scientists still have the burden to adopt the domain-specific problem into something that can benefit from the outsourcing capabilities of the cloud. This problem becomes even worse when the provider infrastructure is not prepared for the new kind of workload.

With the situation of combined challenges for scientific cloud providers and scientific cloud consumers, this special issue of the *Journal of Integrated Design and Process Science* focuses on some of the resulting transdisciplinary challenges. The authors in this issue present their ideas of what shall be done on the provider side to realize the best-possible service for scientific users. One example is the scheduling of virtual machines between different Cloud systems, which allows the implementation of crucial service level agreements. This is especially relevant for scientific workloads that are driven by resource constraints or time constraints, e.g. in weather prediction. Another article describes a completely novel architecture for deploying clouds on clouds, an approach that is especially relevant for cost optimization and cloud infrastructure research in itself. Context-based service access and virtualized teaching assignments are two novel ideas described in this issue, which both allow to map use cases from academia on cloud infrastructures.

We hope that this special issue gives you some insights into the problem domain of scientific cloud computing. The editors would like to thank all authors for their fascinating contributions and hard work in the editing process. A special thank goes to the editors-in-chief for their continuous support in the production of this special issue. Don't hesitate to contact the authors or editors for an exchange of thoughts and ideas about this emerging topic.

## References

Columbus, L. (2013). *Predicting enterprise cloud computing growth.* Retrieved from http://www.forbes.com/sites/louiscolumbus/2013/09/04/predicting-enterprise-cloud-computing-growth/

Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008, November). Cloud Computing and Grid Computing 360-Degree Compared. In *Grid Computing Environments Workshop* (p. 1-10). doi: 10.1109/GCE.2008.4738445

Gibson, J. D., & Kasravi, K. (2012). Predicting the future of it services with triz. *Journal of Integrated Design and Process Science*, *16*(2), 5-14. doi: 10.3233/jid-2012-0015

Lämmel, R. (2008). Google's mapreduce programming model - revisited. *Sci. Comput. Program.*, *70*, 1-30. doi: 10.1016/j.scico.2007.07.001

Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2012). *Nist cloud computing reference architecture: Recommendations of the national institute of standards and technology (special publication 500-292).* USA: CreateSpace Independent Publishing Platform.

Marathe, A., Harris, R., Lowenthal, D. K., de Supinski, B. R., Rountree, B., Schulz, M., & Yuan, X. (2013). A comparative study of high-performance computing on the cloud. In *Proceedings of the 22nd international symposium on high-performance parallel and distributed computing* (pp. 239–250). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2462902.2462919 doi: 10.1145/2462902.2462919

Mell, P., & Grance, T. (2011, September). *The nist definition of cloud computing* (Tech. Rep. No. 800-145). Gaithersburg, MD: National Institute of Standards and Technology (NIST). Retrieved from http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Ng, P. A., & Wei, C.-S. D. (2011). Healthcare information visualization and visual analytics. *Journal of Integrated Design and Process Science*, *15*(4), 1-1.

Nikolaos G. Bourbakis, M. M. (2000). A floorplanning-synthesis methodology for multiple chip module design. *Journal of Integrated Design and Process Science*, *4*(1), 67-81.

Ochs, C., Geller, J., & Perl, Y. (2011). A relationship-centric hybrid interface for browsing and auditing the umls. *Journal of Integrated Design and Process Science*, *15*(4), 3-25.

Sotomayor, B., Montero, R. S., Llorente, I. M., & Foster, I. (2009). Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing*, *13*(5), 14-22. doi: 10.1109/MIC.2009.119

Stein, L. (2010). The case for cloud computing in genome informatics. *Genome Biology*, *11*(5), 207. Retrieved from http://genomebiology.com/2010/11/5/207 doi: 10.1186/gb-2010-11-5-207

Tröger, P., & Merzky, A. (2013, September). Towards standardized job submission and control in infrastructure clouds. *Journal of Grid Computing*. Retrieved from http://link.springer.com/article/10.1007%2Fs10723-013-9275-2# doi: 10.1007/s10723-013-9275-2

Vecchiola, C., Pandey, S., & Buyya, R. (2009, Dec). High-performance cloud computing: A view of scientific applications. In *Pervasive systems, algorithms, and networks (ispan), 2009 10th international symposium on* (p. 4-16). doi: 10.1109/I-SPAN.2009.150

## Author Biographies

**Dr. Daniel Versick** is research assistant at the Computer Architecture Group of Rostock University, Germany. He studied Technical Computer Science and received his Ph.D. at the University of Rostock in 2010. His research interests include distributed computing, high-performance I/O, virtualization in data centers and embedded systems as well as power consumption measurement and optimization of IT systems. Dr. Versick is program committee member of several conferences i.a. with focus on distributed and green computing. He organized numerous scientific symposia and is author and co-author of various publications in his research fields.

**Dr. Peter Tröger** is a senior researcher at the Hasso Plattner Institute for Software Engineering, Germany. He received a doctoral degree from University of Potsdam in 2008 for his work about stateful service-oriented infrastructures. Peters current research interest is in dependability and programmability aspects of modern many-core environments. He is currently focusing on novel dependability modeling concepts and proactive failure prediction schemes, and has research collaborations with the SAP Innovation Center, Univa, FZ Jlich, Audi, and IBM Labs Bblingen. Peter contributed to more than 40 publications and co-authored 3 books in the area of dependable parallel and distributed systems.