Open Access

Open Journal of Cloud Computing (OJCC)
Volume 2, Issue 1, 2015
http://www.ronpub.com/ojcc
ISSN 2199-1987

Scalable Distributed Computing Hierarchy: Cloud, Fog and Dew Computing*

Karolj Skala[A], Davor Davidović[A], Enis Afgan[A,B], Ivan Sović[A], Zorislav Šojat[A]

[A] Centre for Informatics and Computing, Rudjer Bošković Institute, Bijenčka cesta 54,
10000 Zagreb, Croatia, {skala, ddavid, isovic, eafgan, sojat}@irb.hr

[B] Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218, USA, enis.afgan@jhu.com

ABSTRACT

The paper considers the conceptual approach for organization of the vertical hierarchical links between the scalable distributed computing paradigms: Cloud Computing, Fog Computing and Dew Computing. In this paper, the Dew Computing is described and recognized as a new structural layer in the existing distributed computing hierarchy. In the existing computing hierarchy, the Dew computing is positioned as the ground level for the Cloud and Fog computing paradigms. Vertical, complementary, hierarchical division from Cloud to Dew Computing satisfies the needs of high- and low-end computing demands in everyday life and work. These new computing paradigms lower the cost and improve the performance, particularly for concepts and applications such as the Internet of Things (IoT) and the Internet of Everything (IoE). In addition, the Dew computing paradigm will require new programming models that will efficiently reduce the complexity and improve the productivity and usability of scalable distributed computing, following the principles of High-Productivity computing.

TYPE OF PAPER AND KEYWORDS

Visionary paper: *Distributed computing, grid computing, cloud computing, fog computing, dew computing, high-productivity computing*

1 INTRODUCTION

The history of distributed computing dates back to 1960s when the first distributed system, IBM System/360 [17, 12], was introduced. Since then, the distributed paradigm emerged as an alternative to expensive supercomputers, powerful but large and inflexible machines that were hard to modify and update. This alternative was required in order to handle new and increasing users needs and application demands. Opposed to supercomputers, distributed computing systems are networks of large number of attached nodes or entities (e.g. computational nodes) formed from computers (machines) connected through a fast local network [2][3].

This architectural design brings several new features, such as high computational capabilities and resource sharing. High computational capabilities are achieved by joining together a large number of compute units via a fast network, while resource sharing allows different distributed entities to be shared among different users based on the resource availability and user's requirements. Moreover, adding, removing and accessing the resource is easy and can be done in a uniform way, allowing multiple devices to communicate and share data between themselves.

The biggest boost in the development of distributed computing occurred around year 2000 when the proces-

---

[*] This is a OpenAir FP7 Pilot Granted Paper (www.openaire.eu).

sor clock rate, network bandwidth and RAM (Random-Access Memory) capacity reached the Giga range. This occurrence has been denoted as the G-phenomena [21] and started the fifth phase of development of computing systems (see Figure 1). Specifically, 1 GHz processors were released by Intel and AMD, gigabit Ethernet was in use and the first computers with 1 GB of RAM became available. This alignment made a virtual integration of spatially distributed computers plausible, further enabling rapid development of distributed systems, thus creating conditions for space-independent concept of distributed systems.

Started by the Grid and Cloud computing paradigms, the G-phenomena was one of the two main driving forces (namely, hardware and software) that have led to the current development of Information and Communications Technologies (ICT). The primary predisposition for that development was achieving substantial speed improvements of processors and their interconnections, and the ability to process more data in memory. High-performance distributed computing systems were founded on Grid computing paradigm while scalable distributed computing systems evolved through Cloud computing paradigm and the Internet of Things.

Today, distributed systems play an important role in almost every aspect of our everyday living. One of the most popular and widely used distributed systems in the world is the Internet, without which the contemporary everyday life would be hard to imagine. The other examples, more hidden from the public view, are large distributed computational and storage infrastructures called Grids and Clouds, mostly used to analyze tremendous amounts of data coming from numerous business, research and development activities such as DNA sequencing, climate changes, medicine, banking, and telecommunications. One of the first large-scale distributed computing systems developed specifically for many-task applications was the Grid [14][5].

The first grid infrastructures were started in late 1990s through several Grid-oriented projects in the United States, while the two major European projects, started in early 2000s, the UK e-Science program and European Union Data Grid project [10]. These gave shape to what is known today as the European Grid Initiative [18] - a pan-European distributed research infrastructure. The availability of the first public Grid infrastructures stimulated the expansion of scientific computing research, and progressively, Grid computing became one of the major paradigms for scientific applications. However, high utilization of public Grid infrastructures by many different groups with different applications and their technical and bureaucratic issues limited their widespread usage.

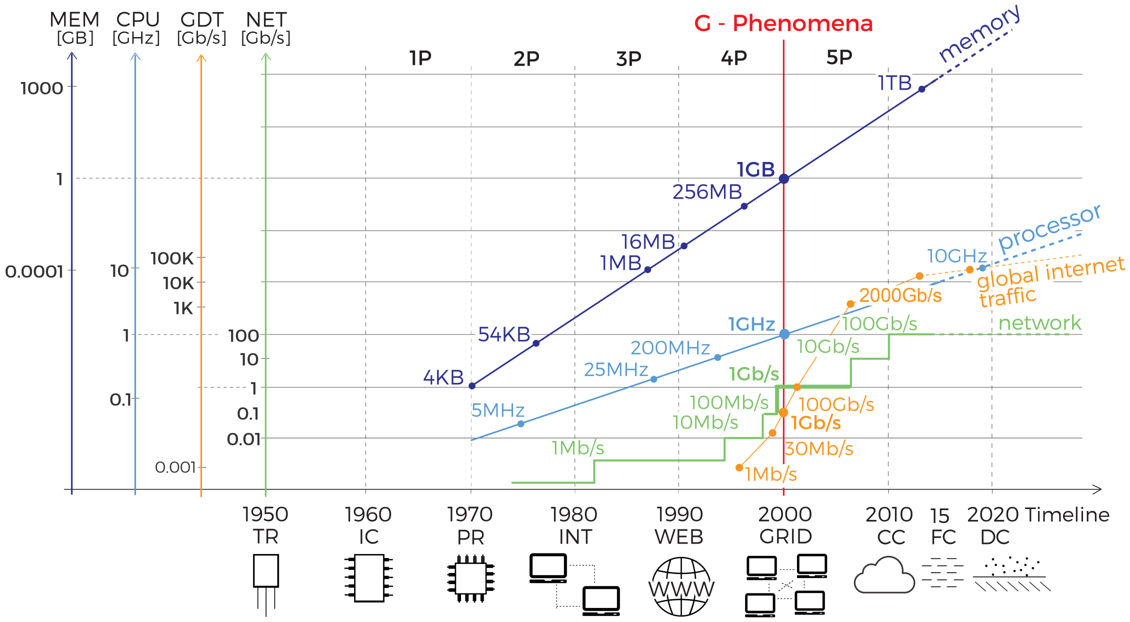
Following the G-phenomena (Figure 1) over the past decade, an increasing number of companies, especially from telecommunication and IT sector are moving from static, centralized cluster environments to more elastic, scalable and potentially cheaper Cloud distributed platforms. Moreover, Cloud computing is becoming attractive to the fast growing Small and Medium-sized Enterprises (SMEs) [22] as it allows them to allocate, increase and decrease the required resource on-demand depending on a rise in service demands. In other words, by moving from the capital upfront investment model to an operational expense, Cloud computing promises, especially to SMEs and entrepreneurs, to accelerate the development and adoption of innovative solutions and lower the operational costs.

Similarly, in many branches of modern research such as genomics, climate change simulations, drugs discovery, and medical research, computational and data-intensive problems have arisen. These problems encompass the generation, analysis and interpretation of vast amounts of data and their comparisons against catalogues of existing knowledge through complex multi-stage workflows. These workflows, or analyses, are enabled by a combination of analysis platforms and computational infrastructures and can be provided as three main Cloud services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

However, in order to meet the needs of the current as well as future research problems, for example real-time human brain simulation, one of the main challenges in the computer science, the scalability, has to be solved. It is expected that the future computing systems will be highly heterogeneous, consisting of various devices such as mobile devices, traditional computers, embedded systems, and sensors, that the present-day models and paradigms could not efficiently solve. Thus, new challenges will require new computing models and paradigms. As an example, a relatively new computing model that can efficiently solve data-intensive problems with reduced power consumption is based on the Field Programmable Gate Arrays (FPGAs) using a data-flow approach to processing [13].

2 HIERARCHY OF SCALABLE DISTRIBUTED COMPUTING

Scalability is the ability of a computer system, network or application to handle a growing amount of work, both in terms of processing power as well as storage resources, or its potential to be easily enlarged in order to accommodate that growth. Today, we are witnessing the exponential growth of data (Big Data) and processing (application) needs which lead to the necessary scalability of resources at multiple levels. In this way we came to a new hierarchical structure, consisting of three



G (GIGA) - Phenomena, TR - Transistor, IC - Integrated Circuit, PR - Processor, INT - Internet, BD - Big Data,
CC - Cloud Computing, FC - Fog Computing, DC - Dew Computing, 1P...5P - Development Phases

Figure 1: The decade phases of the development of ICT and its relations to the G-phenomena and Distributed Computing Hierarchy: Cloud Computing (CC), Fog Computing (FC) and Dew Computing (DC).

layers: Cloud, Fog, and Dew Computing [20]; scalable platforms that represent a new computing paradigm, as described in Figure 2. These hierarchical layers are implemented to facilitate the rapidly developing complex distributed computer systems and meet the following requirements:

- **Performance:** optimized for fast responses, processing and low latency;
- **Availability:** requires redundancy, rapid recovery in the case of system failures, and graceful degradation when problems occur;
- **Reliability:** system needs to be reliable in data and function;
- **Manageability:** scalable system that is easy to operate;
- **Cost:** includes hardware and software costs but it is also important to consider other facets needed to deploy and maintain a scalable computing system.

Figure 2 shows the hierarchical division of CC / FC / DC with the structural, resource and application aspects. Vertical development of scalable virtualisation in a distributed computing system has a role to extend the application domain to the comprehensiveness of applications across a variety of categories and types in the linked chain integrity and overall application functionality.

2.1 Cloud Computing

The term Cloud Computing (CC) was first introduced by professor Ramnath Chellappa in 1997 during a lecture in which he defined it as a new computing paradigm where the boundaries of computing will be determined by economic rationale rather than technical limits [9]. Cloud computing enables efficient management of data centers, time-sharing, and virtualization of resources with a special emphasis on the business model [3][4][8].

Through this paradigm, users and businesses are able to consume Infrastructure-as-a-Service, Platform-as-a-Service and Software-as-a-Service models on demand from anywhere in the world. When Cloud's services are made available by cloud provider in a pay-as-you-go manner to the general public, it is referred to as Public Cloud, while Private Cloud refers to internal Cloud platforms owned and utilized only by particular organizations. Unlike Grids, Clouds can scale up or down according to the requirements of users and at the same time enable more flexible and isolated execution environments for applications.

In Grid computing the logical choice was to keep the

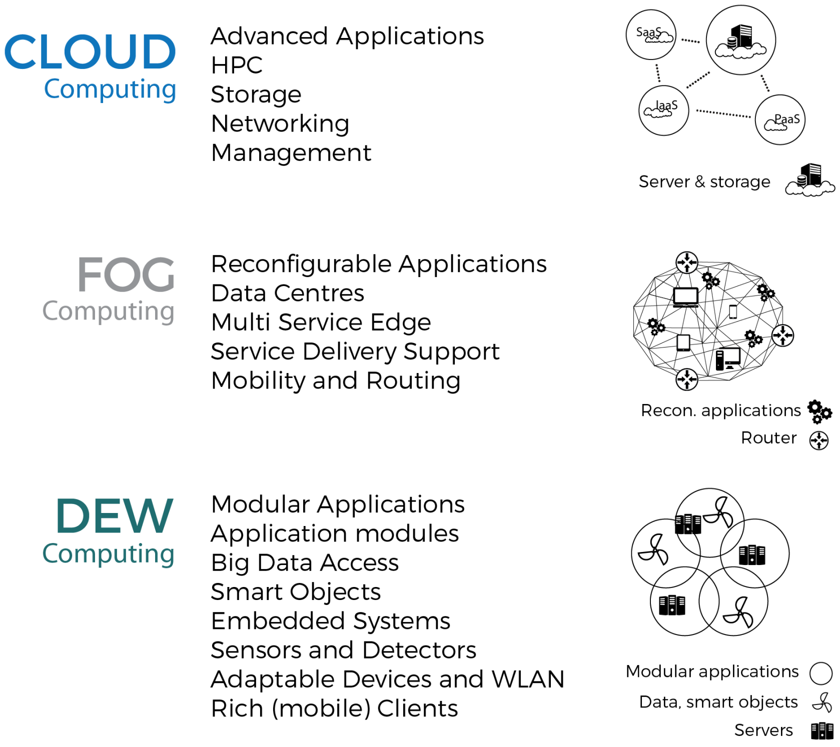


Figure 2: Scalable Distributed Computing Hierarchy.

heterogeneity of the computing equipment to the minimum possible (same operating system, processor type, software libraries etc., particularly at a specific Grid site, but also across the entire Grid infrastructure), whereas the Cloud, via virtualization, allows much wider possibility of choices. From the end-users perspective, Cloud delivers a service (IaaS, PaaS or SaaS) that hides the underlying complexity and heterogeneity which is perceived by the user as a single, homogeneous service. That means that the users needs will be processed by an already existing hardware/software combination, with no inherent restriction in the CC paradigm, which would disallow (or at least make hard) the usage of very heterogeneous computing equipment.

The user is not actually aware of the underlying hardware, e.g. operating system, processor, memory, or disk, used by a particular part of the Cloud. With the maturity of the Cloud technologies more companies, especially SMEs, are moving from self-managed, self-provided and self-acquired PaaS and SaaS solutions to more flexible, time-sharing oriented Cloud solutions which facilitate the development and make their business processes and products more competitive.

2.2 Fog Computing

The new generation of smart devices, capable of processing data locally rather than sending them to the Cloud or any other computational system for processing, enables a new area of possibility and applicability for distributed systems. The Fog Computing (FC) [1] paradigm, first introduced by CISCO Systems, Inc., is a distributed computing paradigm that provides data, compute, storage and application services closer to client or near-user edge devices, such as network routers. Furthermore, FC handles data at the network level, on smart devices and on the end-user client side (e.g. mobile devices), instead of sending data to a remote location for processing.

The goal of the FC is to improve the efficiency by directly processing data on the network (e.g. on network routers or smart devices) and thus reducing the amount of data that has to be transferred for processing but also keeping data closer to the user thus enhancing the security which is one of the main problems in CC [19]. The distinguishing characteristics of FC are its proximity to specific end-users, its dense geographical distribution, the ability to reconfigure itself and its support for mobility as well as its potential to improve security. Services in FC are hosted at the network edge, which reduces service latency and improves the overall Quality of Service (QoS), resulting in a superior level of distributed computing functionality.

Today, with the overwhelming abundance of sensors and devices that generate enormous amounts of data, the FC paradigm is becoming increasingly popular and attractive. Moreover, in numerous applications, transferring a large amount of data from sensors to distant processing systems can be inefficient and possibly leads to an unnecessary network overhead that can result in performance and security drops (e.g. real-time vehicle-to-vehicle communication). Following this example, the FC paradigm is well positioned for Big Data and real-time analytics and knowledge extraction.

FC is a new conceptual solution to dealing with the demands of the increasing number of Internet-oriented (or connected) devices sometimes referred to as the Internet of Things (IoT) [6][16]. In the IoT scenario, the term thing refers to any natural or artificial data-oriented object that an Internet Protocol (IP) address can be assigned to, thus providing it with the ability to acquire and transfer data over a network to a remote (distributed) computing environment [7]. Moreover, today, a large number of Things connected to the Internet are capable of producing vast amounts of data, for which FC will play a key role in reducing the latency in dynamical network-based applications.

2.3 Dew Computing

Dew Computing (DC) goes beyond the concept of a network/storage/service, to a sub-platform - it is based on a micro-service concept in vertically distributed computing hierarchy. Compared to Fog Computing, which supports emerging IoT applications that demand real-time and predictable latency and the dynamic network reconfigurability, DC pushes the frontiers to computing applications, data, and low level services away from centralized virtual nodes to the end users.

In [23] the author proposes a cloud-dew type of architecture that scatters the information around the end-users devices (e.g. laptops or smart-phones) thus enabling data access even when no Internet connection is available. The DC approach also leverages resources that are continuously connected to a network, such as smart phones, tablets and sensors. As a result, DC covers a wide range of technologies including wireless sensor networks, mobile data acquisition, cooperative applications on distributed peer-to-peer ad hoc networking and processing environments. These extend further to autonomic self-healing networks, data intensive and augmented reality applications that open everyday, general purpose usage in working and living environments, down to the level of the simple and special purpose equipment and micro-services.

One of the main advantages of the Dew Computing is in its extreme scalability. The DC model is based on a large number of heterogeneous devices and different types of equipment, ranging from smart phones to intelligent sensors, joined in peer-to-peer ad hoc virtual processing environments consisting of numerous micro-services. Although highly heterogeneous, the active equipment in DC is able to perform complex tasks and effectively run a large variety of applications and tools. In order to provide such functionality, the devices in DC are ad hoc programmable and self-adaptive, capable of running applications in a distributed manner without requiring a central communication point (e.g. a master or central device).

3 DISCUSSION ON VISIONARY ASPECTS

In contrast to the CC and FC paradigms, DC is based on Information-oriented processing rather than Data-oriented, which requires new, more flexible protocols and more productive programming models. In this sense, (raw) data are context-free, while information is data with accompanying meta-data. The meta-data places the data in a specific context enabling more comprehensive analyses and knowledge discovery. The existing computing paradigms, such as Cloud and Fog Computing, operate on huge amounts of raw data generated by specific Things, via predefined services.

Since the raw data is out of context, the services need to be tailored and application specific, requiring data driven decisions. Building an integrated scalable heterogeneous distributed computing environment from the level of Cloud or Fog is currently not plausible (or viable), as it disallows the generic integration of all processing elements. On the other hand, in the Dew Computing scenario, individual Things are responsible for colleting/generating the raw data, and are thus the only components of the computing ecosystem which are completely aware of the context the data were generated in.

As the Dew Computing is placed at the lowest level of the computing hierarchy, the context therefore needs to propagate from the bottom-level up using well-defined, standardized and information-oriented protocols of communication. Context-aware communication services have already been proposed before for application-specific cases such as enhanced IP telephony [15]. We suggest that one of the possible directions for the development of the Dew Computing could be through development of an extended OSI model (Open Systems Interconnection), where a new (eight) Context layer would be added on top of the existing Application layer. The context information could then be propagated through the remaining layers and automatically communicated by any network-enabled device, making any device which uses this extended model a Dew-enabled device.

However, not only protocols of communication need to change. The main programming principles of DC have to be of quite a different paradigm, as common programming models cannot cope efficiently with the necessities of the Internet of Everything. Cloud Computing should be further extended to seamlessly cope with the new presence of context in the data (the information), while still providing legacy for existing services. For example, Cloud Computing could then also be extended with a new layer which provides Context-as-a-Service (CaaS) on top of the existing Software-as-a-Service (SaaS) layer.

On the CaaS layer, programming must be completely oriented on processing and exchanging of information, not only the raw data. New programming models must be developed which will allow interplay amongst CaaS services. Recently, ideas of context graphs and context management systems were proposed [11]which might present a possible direction in which the development of the new Dew-oriented programming models should be steered.

Another important issues that DC has to solve is: Personal High Productivity. To be able to define the necessities of high productivity in the sense of the aim of CC/FC/DC, the exploration of the concept of High Productivity Computing, or, maybe better to say, High Productivity Information Processing and Computing has to be undertaken. The concept of Productivity is changing through time to encompass some aspects and put less significance on other aspects of the processes involved. The objective of further development is to explore the development of necessities in the area of high productivity scalable distributed computing. The extended visionary aspect and strict definition of the concept of High Productivity in Information and Computing Sciences will be a valuable effort in the next computer science development phase.

An important aspect of the Personal High Productivity Computing, which may be well exploited inside the principles of DC, is High Productivity of the equipment, or something probably best called service efficiency in the overall distributed computing hierarchy. Although the scientific use of computing equipment was partially gained from the Giga/Tera/Peta/Exa drive, the expansion was not primarily driven from the side of scientific needs, but from the common consumers. This can be observed over the past several decades through scientific efforts put in to find alternate computer architectures and programming models, specifically adapted to dedicated needs or generic types of programming principles.

As a growth of two or more orders of magnitude of todays computing systems is expected in the near-by future, including systems with unprecedented amounts of heterogeneous hardware, numbers of users, and volumes of data, sustainability is critical to ensure the feasibility of those systems. Due to those needs, currently there is an emerging cross-domain interaction between high-performance in Clouds and the adoption of distributed programming paradigms. The central objective of the programmability challenge is the development, definition and standardization of a universal environment from the level of hardware up to the level of expression of complex algorithms involving a wide spectrum of available information processing and computing equipment.

This kind of approach will enable wide usage of information processors to solve a myriad of ad-hoc problems by common people, i.e. solving problems by not involving specialized computer system programming knowledge, but by enabling context and information aware ”things” and ”processors” in the immediate or wider surrounding (environment) of the common user (depending on the request/algorithm) to provide, based on the universal information and algorithmic environment, proper solutions.

On a practical level, the Dew Computing paradigm will be very useful in everyday life. So, for example, a typical part of the Dew could be an integrated traffic control system of a town, where individual simple data collection/processing units located at all street segments between traffic lights would exchange information about traffic heaviness in their area of responsibility, allowing a collective computerized overview of the whole situation, and enabling auto-adaptive traffic control behavior. As part of such a system individual cars and drivers could also be involved, as information generators, information consumers and information processors. So it would be possible for example to inform hybrid cars approaching a congested area to switch to conventional fuel until entering the congestion, as to collect as much electrical energy as possible for moving through the congestion, therefore significantly reducing the exhaust smoke densities.

The information collected and processed on this level would be, through the hierarchy described, available for many advanced studies and analyses. And finally, it is important to note that the DC paradigm will have to also involve, through the Personal High Productivity Computing Environment, bilateral co-operation of the Person and the Information System, by enabling not only the Human user to pose ”questions” to the computing environment, but also the environment to pose specific questions about possible missing information or solution algorithms to the user, as to be able to intelligently process the user requests.

4 CONCLUSION

Cloud, Fog and Dew distributed computing systems are the result of the exponential development rate of computing and related technologies over the past 50 years. This development is the most prominent driving force of the human society. It is expected that new computing technologies will continue to emerge, such as today's researched photonic, nanocomputing and quantum computing paradigms, that will continue to make distributed computer systems more powerful than any standalone computer. This predicted development, and the ever increasing heterogeneity span shows that many of the present day notions of programmability, user-interaction and ad-hoc definition of user needs will have to be heavily adapted, defined and/or re-defined to enable the user-experience of a simple and integrated living environment.

The new computing paradigm, Dew Computing, will focus on the three major points: Information processing (raw data and metadata describing those data), High Productivity of user-requests (programmability / reconfigurability) and High Efficiency of the equipment (complexities of everyday human information environment).

ACKNOWLEDGEMENTS

This work was co-funded by the European Commission through the AIS DC (EU FP7) project, EGI-Engage project (Horizon 2020) under Grant number 654142 and INDIGO DataCloud projects (Horizon 2020) under Grant number 653549. This paper was submited to Open Journal Of Cloud Computing (OJCC) on September 17, 2015 and accepted on December 14, 2015. It is a OpenAir FP7 Pilot granted paper (https://www.openaire.eu/postgrantoapilot).

REFERENCES

[1] M. Abdelshkour, "IoT, from Cloud to Fog Computing," http://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing, accessed 28th Sep 2015.

[2] E. Afgan, P. Bangalore, and K. Skala, "Application Information Services for distributed computing environments." *Future Generation Comp. Syst.*, vol. 27, no. 2, pp. 173–181, 2011.

[3] E. Afgan, P. Bangalore, and T. Skala, "Scheduling and planning job execution of loosely coupled applications." *The Journal of Supercomputing*, vol. 59, no. 3, pp. 1431–1454, 2012.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[5] F. Berman, G. Fox, and A. J. Hey, *Grid computing: making the global infrastructure a reality.* John Wiley and sons, 2003, vol. 2.

[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16.

[7] F. Bonomi, R. A. Milito, P. Natarajan, and J. Zhu, "Fog Computing: A Platform for Internet of Things and Analytics." in *Big Data and Internet of Things*, ser. Studies in Computational Intelligence, N. Bessis and C. Dobre, Eds. Springer, 2014, vol. 546, pp. 169–186.

[8] R. Buyya, C. Yeo, S. Venugopal, and J. B. I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[9] R. K. Chellappa, "Intermediaries in Cloud-Computing: A New Computing Paradigm," *INFORMS Annual Meeting*, Oct. 1997.

[10] 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," 2001.

[11] B. Chihani, "Enterprise context-awareness : empowering service users and developers," Software Engineering [cs.SE]. Institut National des Telecommunications, Tech. Rep., 2013.

[12] ETHW, "IBM System/360," http://ethw.org/IBM_System/360, accessed 18th November 2015.

[13] M. J. Flynn, O. Mencer, V. Milutinovic, G. Rakocevic, P. Stenstrom, R. Trobec, and M. Valero, "Moving from Petaflops to Petadata," *Communications of the ACM*, vol. 56, no. 5, pp. 39–42, May 2013.

[14] I. T. Foster and C. Kesselman, *The Grid, Blueprint for a New Computing Infrastructure*. San Francisco: Morgan Kaufmann Publishers, 1999. [Online]. Available: http://www.bibsonomy.org/bibtex/283302c972e4e27f2333c31fc21beaca9/buzz

[15] M. Gortz, R. Ackermann, J. Schmitt, and R. Steinmetz, "Context-aware Communication Services: A Framework for Building Enhanced IP Telephony Services." in *ICCCN*. IEEE, 2004, pp. 535–540.

[16] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A

Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[17] IBM, "IBM, System 360 Annoucement," https://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_PR360.html, accessed 18th November 2015.

[18] D. Kranzlmüller, J. Marco de Lucas, and P. Őster, "The European Grid Initiative (EGI): Towards a Sustainable Grid Infrastructure," in *Remote Instrumentation and Virtual Laboratories*. Springer, 2010, pp. 61–66.

[19] K. Shenoy, P. Bhokare, and U. Pai, "Fog Computing Future of Cloud Computing," *International Journal of Science and Research*, vol. 4, no. 6, pp. 55–56, Jun. 2015.

[20] K. Skala, D. Davidović, E. Afgan, I. Sović, T. Lipić, and E. Cetinić, "Scalable Distributed Computing Common Hierarchy," https://indico.egi.eu/indico/contributionDisplay.py?contribId=121&confId=2544, EGI Community Forum, unpublished materials.

[21] K. Skala, D. Davidović, T. Lipić, and I. Sović, "G-Phenomena as a Base of Scalable Distributed Computing —G-Phenomena in Moore's Law," *International Journal of Internet and Distributed Systems*, vol. 2, no. 1, pp. 1–4, 2014.

[22] N. A. Sultan, "Reaching for the "cloud": How SMEs can manage." *Int J. Information Management*, vol. 31, no. 3, pp. 272–278, 2011.

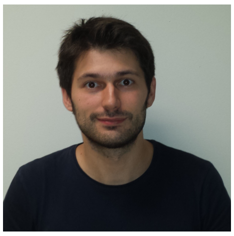
[23] Y. Wang, "Cloud-dew architecture," *International Journal of Cloud Computing*, vol. 4, no. 3, pp. 199–210, 2015.

AUTHOR BIOGRAPHIES

Prof. Karolj Skala is a scientific advisor at the Rudjer Bošković Institute and full professor University of Zagreb. Initiator of the CRO GRID national program from which emerged CRO NGI. He participated in the EU projects: COST 254, COST 276, COST IC0805 and COST IC1305 from Croatia. He successfully completed five EU FP 6, seven EU FP7 projects and provide 3 Horizon 2020 Cloud computing projects. Member of the Croatian Academy of Engineering and an associate member of the Hungarian Academy of Sciences.

Dr. Davor Davidović is a research assistant at Rudjer Bošković Institute. He obtained his Ph.D. in 2014 from the Faculty of Electrical Engineering and Computing at the University of Zagreb. His research interests are in parallel and distributed computing, with the emphasis on the high-performance and GPU computing, heterogeneous computing, and numerical linear algebra. He participated in 4 EU projects: EGEE-III, SEE-GRID-SCI, SCI-BUS, and COST Action IC0805, and is actively involved in 3 projects: Cost Action IC1305, EGI-Engage, Indigo-Datacloud, and SESAME-Net. He has published 15 original scientific papers in international journals and conference proceedings.

Dr. Enis Afgan is a research scientist at the Rudjer Bošković Institute in Zagreb, Croatia. He obtained his Ph.D. in 2009 from the Department of Computer and Information Sciences at the University of Alabama at Birmingham (UAB). Since 2009 he has been a member of the Galaxy Project team and since 2012 a member of the Genomics Virtual Lab project. He is a recipient of the FP7 Marie-Curie Reintegration Grant, AIS-DC. He leads on the CloudMan project. His research interests focus around distributed computing, with the current emphasis on application- and user-level accessibility of cloud computing resources.



Ivan Sović is a research assistant at the RBI and a PhD student at the University of Zagreb Faculty of Electrical Engineering and Computing. He received his M.Sc. in Electrical Engineering and Information Technology in 2010, and enrolled in a PhD programme in Computer Science at the same University in 2011. His research interests include algorithms for sequence alignment and de novo genome assembly with a focus on third generation sequencing data, machine learning, signal processing and reconfigurable computing platforms.

Prof. Zorislav Šojat is a professor of General Linguistics and Socio-Humanistic. His main field of interest is Cybernetics, and he applies Cybernetic principles in scientific research in the area of computer science and intelligence stimulation. From 1991 till 2000 he was Assistant Professor for APL, Lisp and Robot Languages on the Department of Information Science at the Faculty of Philosophy University of Zagreb. He studied Linguistics and Socio-Informatics. He has published hundreds of papers in the field of linguistics, ethics, informatics, philology, computing, economics, philosophy, as well as several books. He is co-author of several world-renowned patents. He speaks fluently Croatian, German, English, Esperanto and actively Italian, Dutch and French.