Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 97 (2016) 131 – 134

Procedia Computer Science

CLOUD FORWARD: From Distributed to Complete Computing, CF2016, 18-20 October 2016, Madrid, Spain

A holistic approach for high-level programming of next-generation data-intensive applications targeting distributed heterogeneous computing environment

Emanuele Carlini[a], Patrizio Dazzi[a,]*, Matteo Mordacchini[b]

[a]CNR-ISTI, Area della Ricerca di Pisa, 56124 Pisa, Italy
[b]CNR-IIT, Area della Ricerca di Pisa, 56124 Pisa, Italy

Abstract

The intrinsic richness and heterogeneity of large amount of data is paired with the extreme complexity in its storing and processing, as well as with the heterogeneity of their processing environments, ranging from super computers to federations of Cloud data-centres. This makes the conception, definition and implementation of software tools for programming applications dealing with very large amount of data really challenging from different perspectives, ranging from technological issues to economic concerns. We propose an approach focused on data-intensive applications that goes beyond the state of the art allowing a seamless exploitation of heterogeneous and distributed resources and satisfying users' needs on data processing providing a dynamically determined set of features, depending on the running environment, the application, the user requirements.

Peer-review under responsibility of organizing committee of the international conference on cloud forward: From Distributed to Complete Computing

Keywords: Cloud Computing; Cloud Federation; Resource Management; Data-intensive Applications; High-level Programming Models

* Corresponding author. Tel.: +39-050-315-3074; fax: +39-050-315-2040.
E-mail address: patrizio.dazzi@isti.cnr.it

Peer-review under responsibility of organizing committee of the international conference on cloud forward: From Distributed to Complete Computing
doi:10.1016/j.procs.2016.08.290

1. Background

Data-intensive applications are one of the largest customers of high throughput computing environments. As clearly shown by the BigData raise, there is a widely recognised urge for processing large amount of data, universally considered the "new oil", from which to extract wisdom, knowledge and, more in general, information of very different kinds, ranging from commercial to financial, from medical to political hints. This extreme intrinsic richness and heterogeneity of data is paired with the extreme complexity in its storing and processing. In this light, today's storage and processing platforms are highly heterogeneous and specially characterised in terms of deployment environments, which include super computers, clusters of commodity or specialized hardware, private and public Cloud data-centres, up to federations of Cloud data-centre. Even more, the heterogeneity is not limited to the deployment at-large but also impacts on the type of resources exploited within a single machine: multicore processors, GPU, FPGA, etc. Cloud technology demonstrated to be an effective solution as it is able to elastically adapt the computing capacity with respect to the (dynamic) needs of applications and customers. In any case, this wide offers in platforms and technologies needs to be properly mastered and leveraged to allow an effective and efficient exploitation of data and resources. As a consequence, the conception, definition and implementation of software tools for programming applications dealing with very large amount of data still represent a hot research topic. This is due to several factors, including technological issues, economic concerns and legal constraints. One of the main aspects is that data is more and more decentralised and localised[1,2], thereby moving it is not always efficient in terms of cost and performance, and cannot even be possible, for example for legal aspects. Further, modern processing platforms are characterised by high levels of dynamicity, in which there is no a static set of resources available for the computation, but rather their kind and numbers depends on the context of a particular application and relative data. As matter of fact orchestrating a computation in such environments is complex, time-consuming, error-prone. Even more, the dynamic nature of users[3], applications and computational resources makes this task extremely complex and almost unfeasible to be conducted without the support of automatic tools. In the last years many approaches, models and solutions have been proposed to tackle these issues.

The problem has been faced from very different perspectives. Some solutions focused to the creation of infrastructures enabling a seamless exploitation of very different kind of resources, both in terms of their hardware heterogeneity[4], and in terms of deployment (infrastructures enabling a transparent exploitation of data-centers, cloudlets as well as resources located at the extreme edge of the network)[5,6]. Other approaches focused on smart brokerage solutions aimed at easing the task of finding the most suitable resources to run an application, depending on several factors including: user and application requirements, location, volume of requests, etc. In addition, have been also proposed approaches aiming at simplify the exploitation of heterogeneous and dynamic resources by leveraging proper programming model abstractions[7]. More recently, both the research and industrial cloud communities are trying to define holistic approaches[8] aimed at providing vertical solutions that, on the one hand ease the programming of applications targeting heterogeneous environments and on the other hand simplify the management of large computational infrastructures. In these approaches users specify their requested service level (e.g. a given throughput), and the supporting environment ensure their satisfaction by adopting the proper resources.

In this position paper we propose our vision for a programming ecosystem for organizing the computation of data-intensive applications in heterogeneous platform that goes beyond the state of the art and try to solve the aforementioned issues. In particular, we are envisioning an ecosystem whose ability is not limited to a seamless exploitation of a set of heterogeneous and distributed resources but it is able to address users' needs about data processing by adopting solutions providing a dynamically differentiated set of features, depending on the actual running environment, the hosted application, the user requirements.

2. Our envisioned Ecosystem

What we envision is an ecosystem in which the functional logic realising the applications self-adapts with respect to the available resources, properly defined user requirements and the actual context. Basically, our idea is to go beyond the traditional application development approach. In fact, a general data-intensive application is realised according to a well-defined process: problem definition, algorithm selection, software implementation, data preparation and application deployment. According to this schema, the application management support, provided by the target execution environments is involved only in the last stage, ensuring that the application is deployed on a proper (set of) resource(s). Some advanced systems are involved also in a few further stages, either by providing ad-hoc programming approach or by leveraging solutions (like Aspect-Oriented or Meta-Programming[9,10]) to customise at runtime non-functional features characterising the application (e.g., logging, encryption, network configuration). However, these systems are usually isolated components whose interoperability is complicated to obtain. Even more, these tools are not designed to adapt the *semantics* of the application to the actual context and situation.

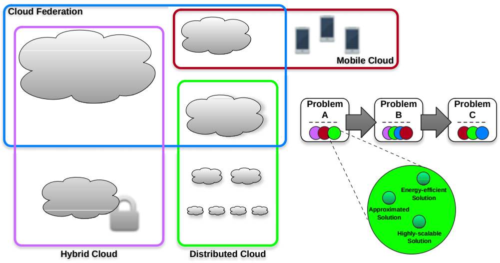

Fig. 1. Our envisioned ecosystem.

Our proposal is the definition of an ecosystem able to affect the realization of next-generation data-intensive application as a whole, rely to application developers mainly for the problem definition and the high-level orchestration logic. According to this vision, application programmers are no longer requested to directly select and implement the specific algorithms realising the application logic, or to explicitly define data movements among resources. In our proposed perspective they define their applications as a composition of "problems", for which exist different solutions, that are eventually adopted and composed. Such enactment and composition results from a co-operative work jointly conducted by the infrastructure run-time management, and the application run-time support.

The infrastructural manager is entitled to determine the best computational, network and storage resources among the available ones, by detecting and exploiting the context of the user and the application. The run-time support of the application analyses the user requirements on: the expected quality on the results of the computation, the available "solutions" for the target "problem", the amount and nature of the data to be processed.

From the interplay of such *active detection entities* is derived the actual, material, logic of the application. Basically, at this stage the applications are effectively composed by software modules specialised with respect to the context, the running environment, the available resources, the input data and the quality expected by the users/customers.

A direct consequence of the proposed vision is a re-identification of the roles characterising the entities involved in this scenario. We already defined the role of "new deal" application developers, no longer involved in the actual implementation of (most of) their own applications, but mainly concerned with a detailed identification of the problems to solve to realise their applications, and with a clear definition of their expectation (quality, cost, performance, etc.) regarding the output of their applications. Each solution solving a problem, among the one identified by application developers are selected from a collection of pre-defined solutions. Such solutions are provided by solution designers, that are responsible for providing highly-efficient solutions, tailored to specific execution environment. Besides their tailoring to a certain architecture/environment, the available solutions can differentiate one each other with respect to the qualitative features provided, e.g., the provided level of privacy preservation, the required network bandwidth, the quality of being able to deliver either exact or approximated results. As matter of fact, the goal for the infrastructure is no longer limited to find the most suitable resources, given an application and its requirements, instead, the intent here is far more challenging and complex.

Thus, the aim is to manage applications defined as a composition of problems which are characterised both by quantitative and qualitative requirements that definitely affect the selection of the solutions that will be adopted for their materialisation. As aforementioned, such selection is performed by an advanced execution environment, resulting from the interplay of both the application runtime and the management environment. In a nutshell, by analysing the user and application requirements, the nature and distribution of the input data, the actual context as well as the available resources, taking into account the desiderata of users describing their desired trade-off on qualitative aspects (e.g., performance, cost, security, precision, etc.) such support decides about the solutions to instantiate to address the problems identified by the application developers. Figure 1 sketches our proposed ecosystem. Basically, it includes some of the most diffused architectures composing the actual landscape of cloud infrastructures: Cloud Federations, Hybrid Clouds, Mobile Clouds and Distributed Clouds. Applications are represented in the figure by a workflow of problems defined by an application developer. Each “problem” can be properly deployed on different cloud architectures by means of specific classes of solutions made available by solution designers. Every class of solutions is in turn composed by specific instances of that solution, each designed for a specific aim, e.g., an implementation delivering approximated results. to be exploited either when the available resources are not enough to adopt an exact solution or its exploitation would be too costly.

3. Conclusion and future work

The valuable information that can be extracted from big amounts of data that are available in many crucial and relevant fields push the need for a new generation of effective and efficient computing ecosystems that support data-intensive applications devoted to the analysis and the knowledge-extraction tasks. In this context, these new ecosystems should face issues related to the intrinsic complexity of the problems to be solved, technological, economic and security and privacy concerns. In addition, the high dynamicity of modern computing platforms and the relationship between the context of a user and the data and computing resource that can be exploited, make the problem even more complex. In order to tackle these issues, in this position paper we proposed a novel computing ecosystem. In the proposed approach, developers of next generation data-intensive applications will work at high-level, by identifying and defining the set of “problems” that should be solved by their applications. The ecosystem will provide intelligent *active detection entities*, that will be in charge of selecting the most suitable implementation for every application “problem”, considering the overall environment, ranging from the context of the user and that of the data to be used and of the available resources, leading to final deployment and orchestration of the various components of the application. We expect that such a new computing ecosystem will allow to reach a more effective and efficient solution for the challenges posed by modern data-intensive application by providing a new, simpler way to program them and, at the same time, by allowing to seamlessly (for both the users and the developer) deploy them in the most effective way, with respect to the actual context of data and computing resources.

Acknowledgements

This work has been partly supported by BASMATI project H2020 G.A. no. 723131 and ICT R&D program of Korean Ministry of Science, ICT and Future Planning no. R0115-16-0001

References

1. Dazzi P, Felber P, Leonini L, Mordacchini M, Perego R, Rajman M, Rivière É. Peer-to-peer clustering of web-browsing users. In *Proceedings of LSDS-IR*; 2009.
2. Baraglia R, Dazzi P, Mordacchini M, Ricci L, Alessi L. Group: A gossip based building community protocol. In Balandin S, Koucheryavy Y, Hu H, editors. *Smart Spaces and Next Generation Wired/Wireless Networking*, Heidelberg, 2011. Lecture Notes in Computer Science, Volume 6869. Springer; p. 496-507.
3. Conti M, Mordacchini M, Passarella A. Design and performance evaluation of data dissemination systems for opportunistic networks based on cognitive heuristics. ACM *Transactions on Autonomous and Adaptive Systems*, 2013;**8**:3, p. 12:1-12:32.
4. Coutinho J, Pell O, ONeill E, Sanders E, McGlone J, Grigoras P, Luc W, Ragusa C. Harness project: Managing heterogeneous compute resources for a cloud platform. In Goehringer D, Santambrogio M D, Cardoso J M P, Bertels K, editors. *Reconfigurable Computing: Architectures, Tools, and Applications*, Heidelberg, 2014. Lecture Notes in Computer Science, Volume 8405. Springer; p. 324-329.
5. Satyanarayanan M, Bahl P, Caceres R, Davies N. The case for vm-based cloudlets in mobile computing. *Pervasive Computing* 2009;**8**:4.
6. Stojmenovic I. Fog computing: a cloud to the ground support for smart things and machine-to-machine networks. In the *2014 Australasian Conference on Telecommunication Networks and Applications Conference*. Southbank. IEEE; p. 117-122.
7. Aldinucci M, Campa S, Danelutto M, Kilpatrick P, Torquati M. Pool evolution: a domain specific parallel pattern. In the *7th International Symposium on High-level Parallel Programming and Applications*. Amsterdam. 2014.
8. Carlini E, Coppola M, Dazzi P, Ricci L, Righetti G. Cloud federations in contrail. In Alexander M et al., editors. *Euro-Par 2011: Parallel Processing Workshops*, 2011. Lecture Notes in Computer Science, Volume 7155. Springer; p. 159-168.
9. Danelutto M, Pasin M, Vanneschi M, Dazzi P, Laforenza D, Presti L. Pal: exploiting java annotations for parallelism. In Gorlatch S, Bubak M, Priol T, editors. *Achievements in European Research on Grid Systems*, 2008. Springer; p. 83-96.
10. Danelutto M, Dazzi P. A Java/Jini framework supporting stream parallel computations. *ParCo 2005*. Central Institute for Applied Mathematics, Jülich: Málaga, 2005; p. 681-688.