23

ADVANCES IN PARALLEL COMPUTING

Cloud Computing and Big Data

Edited by
Charlie Catlett
Wolfgang Gentzsch
Lucio Grandinetti
Gerhard R. Joubert
José Luis Vazquez-Poletti

IOS Press

Cloud Computing and Big Data

Edited by

Charlie Catlett

USA

Wolfgang Gentzsch

Germany

Lucio Grandinetti

Italy

Gerhard Joubert

Netherlands/Germany

and

José Luis Vazquez-Poletti

Spain

IOS Press

Amsterdam • Berlin • Tokyo • Washington, DC

ISBN 978-1-61499-321-6 (print)
ISBN 978-1-61499-322-3 (online)
Library of Congress Control Number: 2013949618

Publisher
IOS Press BV
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

PRINTED IN THE NETHERLANDS

CLOUD COMPUTING AND BIG DATA

Advances in Parallel Computing

This book series publishes research and development results on all aspects of parallel computing. Topics may include one or more of the following: high-speed computing architectures (Grids, clusters, Service Oriented Architectures, etc.), network technology, performance measurement, system software, middleware, algorithm design, development tools, software engineering, services and applications.

Series Editor:
Professor Dr. Gerhard R. Joubert

Volume 23

Recently published in this series

Vol. 22. K. De Bosschere, E.H. D'Hollander, G.R. Joubert, D. Padua and F. Peters (Eds.), Applications, Tools and Techniques on the Road to Exascale Computing
Vol. 21. J. Kowalik and T. Puźniakowski, Using OpenCL – Programming Massively Parallel Computers
Vol. 20. I. Foster, W. Gentzsch, L. Grandinetti and G.R. Joubert (Eds.), High Performance Computing: From Grids and Clouds to Exascale
Vol. 19. B. Chapman, F. Desprez, G.R. Joubert, A. Lichnewsky, F. Peters and T. Priol (Eds.), Parallel Computing: From Multicores and GPU's to Petascale
Vol. 18. W. Gentzsch, L. Grandinetti and G. Joubert (Eds.), High Speed and Large Scale Scientific Computing
Vol. 17. F. Xhafa (Ed.), Parallel Programming, Models and Applications in Grid and P2P Systems
Vol. 16. L. Grandinetti (Ed.), High Performance Computing and Grids in Action
Vol. 15. C. Bischof, M. Bücker, P. Gibbon, G.R. Joubert, T. Lippert, B. Mohr and F. Peters (Eds.), Parallel Computing: Architectures, Algorithms and Applications

Volumes 1–14 published by Elsevier Science.

ISSN 0927-5452 (print)
ISSN 1879-808X (online)

Cloud Computing and Big Data
C. Catlett et al. (Eds.)
IOS Press, 2013

Preface

At the International Research Workshop on Advanced High Performance Computing Systems in Cetraro in June 2012, two of the main workshop topics were High Performance Computing (HPC) in the Cloud and Big Data.

Cloud computing offers many advantages to researchers and engineers who need access to high performance computing facilities for solving particular compute intensive and/or large scale problems, but whose overall HPC needs do not justify the acquisition and operation of dedicated HPC facilities. The questions surrounding the efficient and effective utilization of HPC cloud facilities are, however, numerous, with perhaps the most fundamental issues being the limitations imposed by accessibility, security and communication speeds.

Therefore, in order to mobilize the full potential of cloud computing as an HPC platform a number of fundamental problems must be addressed. On the one hand it must be identified which classes of problems are amenable to the cloud computing paradigm with its limitations. On the other hand it must be clarified which technologies, techniques and tools are needed to enable a widely acceptable use of cloud computing for HPC.

The second topic, big data, is nothing new. Large scientific organizations have been collecting large amounts of data for decades. What is new, however, is that the focus has now broadened to almost all sectors – be it business analytics in enterprises, financial analyses, Internet service providers, oil and gas, medicine, automotive, and a long list of others.

This book presents three chapters with together 14 contributions, selected from the International Research Workshop on Advanced High Performance Computing Systems in Cetraro in June 2012. The five contributions of the first chapter on "Cloud Infrastructures" discuss several important topics of High Performance Computing in the cloud, covering automatic clouds with an open-source and deployable Platform-as-a-Service; QoS-aware cloud application management; building secure and transparent inter-cloud infrastructure for scientific applications; cloud adoption issues such as interoperability and security; and semantic technology for supporting software portability and interoperability in the cloud.

Chapter two discusses "Cloud Applications", with a focus on using clouds for technical computing; dynamic job scheduling of parametric computational mechanics studies; the bulk synchronous parallel model; and executing multi-workflow simulations on clouds.

Finally, the articles in chapter three are dealing with "Big Data" problems such as ephemeral materialization points in stratosphere data management on the cloud; a cloud framework for big data analytics workflows; high performance big data clustering; scalable visualization and interactive analysis using massive data streams; and mammoth data in the cloud from clustering social images. The editors wish to thank all

the authors for preparing their contributions as well as the many reviewers who supported this effort with their constructive recommendations.

Charlie Catlett, USA
Wolfgang Gentzsch, Germany
Lucio Grandinetti, Italy
Gerhard Joubert, Netherlands/Germany
José Luis Vazquez-Poletti, Spain

14 July 2013

Reviewers

Charlie Cattlet	USA
Erik D'Hollander	Belgium
Sudip Dosanjh	USA
Wolfgang Gentzsch	Germany
Sergei Gorlatch	Germany
Lucio Grandinetti	Italy
Gerhard Joubert	Germany
Odej Kao	Germany
Janusz Kowalik	USA
Marcel Kunze	Germany
Erwin Laure	Sweden
Thijs Metsch	Germany
Domenico Talia	Italy
Jose Luis Vazquez-Poletti	Spain

Contents

Chapter 3. Big Data

Chapter 1

Cloud Infrastructures

Cloud Computing and Big Data
C. Catlett et al. (Eds.)
IOS Press, 2013

doi:10.3233/978-1-61499-322-3-3

Building Automatic Clouds with an Open-source and Deployable Platform-as-a-service

Dana PETCU [1]
Computer Science Department, West University of Timişoara

Abstract. Cloud computing is making a new step towards the already old vision of utility computing of seamless delivery of computing, storage or networks as measurable consumables. However, completely automated processes are not yet in place, and human intervention is still required. This paper intends to provide a snapshot of the current status in the automated processes happening in the Cloud. Moreover, a special attention is given to a recent developed platform-as-a-service that was designed to be an open-source and deployable middleware: ensuring the portability of applications based on elastic components and consuming infrastructure services, it is a good example of the potential of automated procedures in a Cloud environment.

Keywords. Automatic Clouds, Platform as a Service, Open source

1. Introduction

The main characteristics of Cloud computing that make it appealing as a new paradigm for distributed computing are the elasticity in resource usage and the on-demand self-service. While the need for human intervention in new software services deployment is considerable reduced and the influence of the location of the resources is diminishing in comparison with previous emerged distributed systems, still the programmers are not completely free from the low level issues as operating system patches, configuration updates, or booting machines, especially when they are dealing with infrastructure services. Further steps should be undertaken to reduce the human intervention at a minimum, and to achieve this aim the essential element is the automation of everything from server allocation, virtual machine deployment to application life-cycle management, as well as fault tolerance.

The automation of resource management has received large interest in the past decade mostly under the name of autonomic computing [1]. The concepts are inspired from biology: the autonomic nervous system takes care of low-level functions of the human body such as temperature regulation. Autonomic computing was launched with the goal to build information systems that are capable to self-manage according to the goals set by human administrators. It is an inter-disciplinary field situated at the cross-

[1] West University of Timişoara, 4 B-dul V. Parvân, 300223 Timisoara, Romania, E-mail: petcu@info.uvt.ro

roads of several well-known branches of computer science: distributed systems, artificial intelligence, bio-inspired computing, software engineering and control systems.

Autonomic approaches are particularly suited for use in Cloud environments, where rapid scalability of the pool of resources is requested to support unpredictable number of demands, and where the system should automatically adapt to avoid failures in the hardware resources which can impact the service level agreements. However, the Cloud computing community has not taken up yet the full benefit of the rich results that have been obtained in the field of autonomic computing. In this paper we intend to identify the latest efforts for introducing the autonomic techniques in building an autonomic platform for Cloud computing (the next section is dedicated to this aim).

A platform-as-a-service (PaaS) is meant to deal with the different stages of the application lifecycle, as well as hiding the complexity of the mechanisms to support application deployment and execution. Therefore a high degree of automation is expected to be reached at this level of Cloud services. However, current PaaS are only partially automated due to different reasons. One reason pointed in [2] is the lack of an architectural model for describing a distributed application in terms of its software stacks (operating system, middleware, application), their instantiation as virtual machines, and their configuration interdependencies. Another reason is the vendor lock-in problem due to the non-portability of applications – this is a considerable threat for the production costs. Therefore we focus our attention to one of most recent platforms that was build to ensure the portability of the applications consuming Cloud resources and using configurable software stacks, namely mOSAIC. Building it in the last two and half years through a collaborative European effort[2] has required the implementation of several automated procedures that will be revealed partially in this paper.

Summarizing, the contributions of this paper can be focused in two statements: (1) estimation of the state of the art in Automatic Clouds; (2) reveal the degree of fulfillment of the characteristics of an Automated Cloud middleware by mOSAIC's platform and its futures improvement needed in order to support Automated Clouds.

2. Research and Implementation Issues in Automatic Clouds

2.1. Concept Definition

In what follows, we consider that the Autonomic Computing refers to the self-managing characteristics of a computing system, while the Cloud computing refers to a distributed system that is distinguished from other systems through elasticity and business model.

An autonomic system is expected to take decisions on its own using certain policies. It should also check and optimize its status and automatically adapt itself to changing conditions.

Consequently, the main characteristics of an Automatic Cloud resulting from applying autonomic computing techniques to Cloud Computing should be:

1. involves distributed computing resources and software services which instance number varies by adapting to unpredictable changes;

[2]Details at the project web site: www.mosaic-cloud.eu

2. performs a contextual behaviour through methods of self-management, self-tuning, self-configuration, self-diagnosis, and self-healing;
3. hides intrinsic complexity to operators and users, as implementing techniques for designing, building, deploying and managing computing systems with minimal human involvement and presents itself as a robust, fault tolerant and easy to manage and operate architecture and deployment.

2.2. *Topics of Current Research*

In what follows we describe shortly the approaches of interest for Automatic Clouds reported in the literature until recently and which we consider relevant.

Architectural Styles. The Cloud optimization architecture proposed in [3] describes the autonomic behavior that can be provided at IaaS, PaaS and SaaS layers and represents a valuable framework for classifying the architectural contributions to Cloud computing. To match the classification requirements at PaaS layer, for example, we need to ignore the programming environment (i.e., the tools for the development of applications) in order to focus on the execution environment with the goal to find an optimum deployment of application components on Cloud resources. The authors of [4] proposed also a framework for evaluating the degree of adaptability supported by an architectural style and classified the most known architectural styles (e.g., pipe and filter, publish/subscribe, SOA, peer-to-peer) according to this framework – the most adaptable architectural styles for Autonomic Clouds should follow such adaptability evaluation. Following these recommendations, we have proposed in [5] a robust and scalable autonomic architecture, but its adaptation to the context of Cloud is an on-going work.

Reactive and proactive techniques for self-healing. The reactive techniques such as the policy-based, goal-based, or utility-based approaches, enable the system (Automatic Cloud) to respond to problems only when they occur [6]. For example, in [7] a policy based management is used to evaluate the state of the system against predefined rules and actuate self-healing to return the system to the desired state – focus is put on investigating the capability of the system to recognize a fault and react to it.

Alternatively, proactive techniques like the ones proposed in [8,9] predict a set of environmental conditions before they actually happen.

Reactive and proactive techniques are usually implemented using multi-agent systems. In the context of Autonomic Clouds we can distinguish between two types of solutions. The first type of solutions exploits the idea of building multi-agent controllers for autonomic systems, which are capable, not only to manage the system, but also to manage themselves, as in [5,10]. The second idea is to allow virtual machines and services to behave like agents and to make decisions based on local policies and local knowledge as in [11,12].

Auto-scaling, scheduling and adaptive resource provisioning. In order to take advantage of the elasticity characteristic of the Cloud, the deployed applications need to be automatically scaled in a way that makes the most efficient use of resources. Several frameworks have been introduced in the last years to support the application development taking into account the need of scalability. For example, SmartScale [13] is an automated scaling framework that uses a combination of vertical (adding more resources to existing VM instances) and horizontal (adding more VM instances) scaling to ensure

that the application is scaled in a manner that optimizes both resource usage and the reconfiguration cost incurred due to scaling. Such scaling strategies are encountered also in [14] where different scalability patterns for a PaaS and an approach to performance monitoring allowing automatic scalability management are presented.

The auto-scaling of application components is useless without techniques for load balancing the requests among the scaled components. Usually load balancing is considered among physical machines, like in [15]. In the context of building Cloud applications from components, a load balancing among software components or services need to be considered.

The scheduling problem is as a multi-objective problem where the transfer, deployment and energy consumption costs need to be simultaneously minimized. Several approaches used in heterogeneous environments (e.g. those presented in [10,16,17,18]) can be applied. However, none of the current approaches are considering simultaneously the transfer, deployment and energy consumption costs.

The problem of finding the mapping which minimizes the cost is NP complete and as a result scheduling (meta-)heuristics should be used to find sub-optimal solutions. Meta-heuristics such as those based on neural networks or evolutionary algorithms or linear programming proved efficient in solving cost problems found in scheduling problems [19].

Service selection, discovery and composition. Due to the tremendous number of Cloud services and the lack of a standard for their specification, manual service selection is a time costly task. Automatic methods for matching the user needs with the offers are therefore needed. In [20] for example a method for finding semantically equal SLA elements from differing SLAs by utilizing several machine learning algorithms is presented, together with a framework for automatic SLA management. Themis [21] is a recent implementation of a proportional-share auction that maximizes resource utilization while considering virtual machine migration costs; it uses a set of feedback-based control policies to adapt the application bid and resource demand to fluctuations in price.

Cloud service description languages should allow the automatically composition of Cloud service to achieve a common shared business goal. The paper [22] formalizes the issue of automatic combination of Cloud services. Moreover a proof-of-the-concept implementation is revealed to leverage a batch process for automatically constructing possible combinations of Cloud services, followed by a search for the best fit solution.

In [23] is presented an approach – named Café (from Composite Application Framework) – to describe configurable composite service-oriented applications and to automatically provision them across different providers. Components can be internal or external to the application and can be deployed in any of the delivery models present in the Cloud. The components are annotated with requirements for the infrastructure they later need to be run on. A component graph is used to capture the dependencies between components and to match against the infrastructures offered by different providers.

Software agents have been successfully used in the recent years for service discovery, brokering or composition. Cloudle [24] is such an agent-based search engine for Cloud service discovery proving that agent-based cooperative problem-solving techniques can be effectively adopted for automating Cloud service composition. As reported in [25], agents can be used also in the mechanism for service migration between Clouds.

Self-configuration. The authors of [2] have recently propose an automated line for deploying distributed application composed of a set of virtual appliances, which includes a decentralized protocol for self-configuring the virtual application machines. The solution named VAMP (Virtual Applications Management Platform) relies upon a formalism for describing an application as a set of interconnected virtual machines and, on the other hand, on an engine for interpreting this formalism and automating the application deployment on an IaaS platform. The formalism offers a global view of the application to be deployed in terms of components with the associated configuration- and interconnection constraints and with their distribution within virtual machines; it extends OVF language, dedicated to virtual machines description, with an architecture description language that allows describing a distributed application software architecture.

The above described approach can be complemented by the one from [26] where is exposed a mechanism that requires zero manual intervention during the configurations on the IP addresses of the resources from a Cloud center.

An automated approach to deploy pre-configured and ready-to-run virtual appliances on the most suitable Cloud infrastructure is still missing. However, in [27] is proposed an architectural approach using ontology-based discovery to provide QoS aware deployment of appliances on Cloud service providers.

Costs versus reliability. A current challenge for Cloud providers is to automate the management of virtual servers while taking into account both high-level quality of service requirements of hosted applications as well as the resource management costs. In this context, the paper [28] proposes an autonomic resource manager to control the virtualized environment which decouples the provisioning of resources from the dynamic placement of virtual machines and which aims to optimize a global utility function which integrates both the degree of SLA fulfillment and the operating costs (a constraint programming approach to formulate and solve the optimization problem).

Probabilistic models were extensively used to assess the reliability of software systems at the architectural level, like in [29,30], and these should to applied also in the particular case of Cloud systems. Moreover, the idea of [31] to reason at run-time about the non-functional attributes of the system and to perform accordingly some adaptations is particularly interesting in the context of Autonomic Clouds.

Most of the Cloud service providers charge their clients for metered usage based on fixed prices. In [32] were exposed pros and cons of charging fixed prices as compared to variable prices. Deploying an autonomic pricing mechanism that self-adjusts pricing parameters to consider application and service requirements of users is shown to achieve higher revenue than various other common fixed and variable pricing mechanisms.

Adaptive resource provisioning. The problem of adaptive virtualized CPU provisioning has received a lot of attention (for example, in [33,34,35]). However an automated adaptive resource provisioning system as proposed on [36], based on feedback controllers (customer add-on outside of the Cloud service itself), was not reported yet.

In [37] an automated framework for resource allocation is presented: it can adapt the adaptive parameters to meet the specific accuracy goal, and then dynamically converge to near-optimal resource allocation (optimal in terms of minimum costs). The proposed solution can handle unexpected changes in the data distribution characteristics and/or rates of the streaming application. Resource allocation for streaming processing was considered also in [38] which proposed an elastic scaling of data parallel operators.