



An ontology for heterogeneous resources management interoperability and HPC in the cloud

Gabriel G. Castañé*, Huanhuan Xiong, Dapeng Dong, John P. Morrison

Department of Computer Science, University College Cork, Ireland

HIGHLIGHTS

- An ontology for interoperability in heterogeneous cloud infrastructures is proposed.
- Enable the adoption of heterogeneous physical resources in self managed clouds-Support for HPC-in-Cloud, hardware accelerators, resource abstraction methods.
- A proposed architecture to exploit the semantic and syntactic benefits.
- Included into the CloudLightning project for large scale Cloud Computing environments.

ARTICLE INFO

Article history:

Received 31 December 2017

Received in revised form 19 April 2018

Accepted 30 May 2018

Keywords:

Cloud interoperability

HPC in cloud

Resource management

Ontology

Self-management clouds

ABSTRACT

The ever-increasing number of customers that have been using cloud computing environments is driving heterogeneity in the cloud infrastructures. The incorporation of heterogeneous resources to traditional homogeneous infrastructures is supported by specific resource managers cohabiting with traditional resource managers. This blend of resource managers raises interoperability issues in the Cloud management domain as customer services are exposed to disjoint mechanisms and incompatibilities between APIs and interfaces. In addition, deploying and configuring HPC workloads in such environments makes porting HPC applications, from traditional cluster environments to the Cloud, complex and ineffectual.

Many efforts have been taken to create solutions and standards for ameliorating interoperability issues in inter-cloud and multi-cloud environments and parallels exist between these efforts and the current drive for the adoption of heterogeneity in the Cloud. The work described in this paper attempts to exploit these parallels; managing interoperability issues in Cloud from a unified perspective. In this paper the mOSAIC ontology, pillar of the IEEE 2302 – Standard for Intercloud Interoperability and Federation, is extended towards creating the CloudLightning Ontology (CL-Ontology), in which the incorporation of heterogeneous resources and HPC environments in the Cloud are considered. To support the CL-Ontology, a generic architecture is presented as a driver to manage heterogeneity in the Cloud and, as a use case example of the proposed architecture, the internal architecture of the CloudLightning system is redesigned and presented to show the feasibility of incorporating a semantic engine to alleviate interoperability issues to facilitate the incorporation of HPC in Cloud.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Cloud computing environments offer customers a wide diversity of services through loosely coupled instances, and storage systems, guaranteeing certain levels of services. Features such as availability on demand, large capacity, elasticity, and service-level performance have been attracting end-users to migrate (to Cloudify) their applications from traditional cluster environments.

It is estimated that by the year 2019 more than 85% of workloads will be processed by Cloud environments [1].

The Cloud market is expanding [2,3] and this growth is attracting specific users demanding specific services that are driving the traditional homogeneous Cloud infrastructure to become a heterogeneous ecosystem. In particular, supporting High Performance Computing (HPC) services in the Cloud requires an evolution of the traditional homogeneous cloud [4]. HPC as-a-service (HPCaaS) is leading the availability of services that, from a cloud management perspective, are challenging to support. Cloudifying HPC applications while maintaining performance, while desirable,

* Corresponding author.

E-mail addresses: gabriel.gonzalezcastane@ucc.ie (G.G. Castañé), h.xiong@ucc.ie (H. Xiong), d.dong@ucc.ie (D. Dong), j.morrison@ucc.ie (J.P. Morrison).

presents significant challenges. Virtualization management, virtual machine monitoring, communications, and processing overheads, detract from the bare metal performance of applications [5]. Currently, only a few such have been Cloudify-ed, since they have been optimized over decades for specific hardware architectures. Early forays into this process can be seen in the attempts to create models for determining the effectiveness of cloud environments in supporting such HPC workloads [6–8].

Cloud-based HPC solutions are offered by several commercial vendors. They include dedicated servers and containers used to access hardware accelerators in cloud computing infrastructures. For example, Amazon EC2 provides instances based on Xeon servers with GPU accelerators and FPGA instances as dedicated servers [9,10]. To support diverse resource abstraction methods, these providers offer multiple frameworks, as required. Foremost among these are OpenStack Nova [11] and OpenNebula [12] to manage virtualized environments; Kubernetes [13], Mesos [14], and Docker Swarm [15] to manage container environments; OpenStack Magnum [16] to manage containers within virtualized environments; OpenStack Ironic [17] to bare-metal servers; and finally, traditional cluster management frameworks for HPC/HTC such as SLURM [18] and ROCKS [19]. However, in spite of the maturity of these frameworks, none supports all resource abstraction mechanisms. Thus, in a heterogeneous environment multiple interacting frameworks are required to adequately support the diversity of resources in that environment. This complexity is exacerbated in the HPC domain where services running on heterogeneous resources may co-operate to deliver the HPC application. Combining resources in this manner is currently state-of-the-art and requires expert low-level configuration. Customers need to be able to configure, deploy, and link services associated with diverse instances and address the interoperability issues associated with using different resource abstractions under the control of multiple resource managers. The deployments of services interacting across a number of management domains are tailored to specific resources managers and are currently supported by specific-to-vendor interfaces potentially using different, and often incompatible, APIs.

Fig. 1 shows an idealized extensible heterogeneous cloud architecture containing multiple resources. This architecture would require sophisticated mechanisms to enable services hosted on different heterogeneous resources to interact within the workflow.

Physical resources (labeled from R_1 to R_5 in the figure) represent different sets of heterogeneous hardware available in the cloud infrastructure. For example, commodity hardware, servers with hardware accelerators, and servers with enhanced capabilities such as low latency networks. In the figure, four coexisting local resource managers are deployed in the system, partitioning the Cloud in support of multiple hardware types and resource abstraction methods: *LocalRM1* uses a *VirtualizationLayer* for providing virtual machines on R_1 and R_5 resources, e.g., OpenStack Nova, Eucalyptus, or OpenNebula; *LocalRM2* and *LocalRM4* uses a *ContainerizationLayer* to provide containers on R_3 , R_4 , and part of the R_5 physical resources, e.g., Marathon, Kubernetes, or Docker-Swarm; and finally *LocalRM3* accesses R_5 hardware resources in a dedicated server fashion, e.g. ROCKS, SLURM, or specific proprietary resource managers. On the left hand side, Fig. 1 shows how the incorporation of a new resource manager into the resources fabric will increase the number of resource management partitions within the management domain of the Cloud. For example, Amazon EC2 F1 and G2 instance types [20] offer virtual machines instances based on FPGA and GPU hardware that have to be manually configured and linked to reflect the composition of services within an application using this hardware.

A consequence of the work proposed in this paper is the elimination of this manual configuration step. The approach taken is to define and use an ontology as a mechanism to achieve semantic interoperability [21] in heterogeneous systems.

The main goals of this work are: to create an ontology, known as the CloudLightning Ontology (CL-Ontology) that supports heterogeneous and high performance resources in the Cloud; to ameliorate interoperability issues between existing resource managers and resource abstractions; to maintain compatibility with previous standards for interoperability.

The CL-Ontology extends the IEEE-2302 (SIIF) – Standard for Intercloud Interoperability and Federation [22], and the IEEE-2301, Guide for Cloud Portability and Interoperability Profiles (CPIP) [23]. This extensions incorporate support for resource management, specific hardware accelerators, and different resource abstraction methods, such as virtual machines and containers into the Cloud; matching service requests to specific heterogeneous infrastructures, and enabling intelligent resource discovery. To the best of our knowledge this is the first attempt to explicitly address an Ontology supporting HPC in Cloud.

In addition to the CL-Ontology, in an effort to demonstrate its utility, this paper presents a conceptual Service Oriented Architecture (SOA) for autonomic resource management. In the proposed architecture, the CL-Ontology is used as part of a semantic engine to dynamically incorporate resources into the cloud resource fabric, and to support decisions making for targeting service requests to appropriate resources. Finally, the architectural design of the CloudLightning [24,25] (EU H2020-ICT programme under grant #643946) system is presented as use case into which the proposed semantic engine is incorporated, and where the management of heterogeneous resources at scale is undertaken.

2. Related work

Many efforts have been made towards addressing interoperability issues in the Cloud by creating common interfaces and APIs that enhance the compatibility between deployment models and public vendors. The Open Cloud Manifesto [26] is an initiative by industry for supporting open standards in cloud computing. The main targets are grouped into five categories: security, data application interoperability, data application portability, monitoring and portability between clouds. However, the Manifesto does not incorporate current cloud technologies nor support for hardware accelerators within any of the above mentioned categories. Similarly, the Universal Cloud Interface (UCI) [27,28] was proposed to solve inter-cloud interoperability avoiding lock-in issues with proprietary solutions by unifying the representation of all cloud resources in a common interface. Its evolution has been very limited and UCI does not incorporate concepts emerging from cloud technologies.

The Guide for Cloud Portability and Interoperability Profiles (CPIP) [23] assist cloud computing vendors and users in developing, building, and using standards-based cloud computing products and services. For each element, multiple options are proposed regarding interfaces, file formats and operational conventions. However, these are grouped in the “Standard profiles”, as drivers for interoperability from a user perspective, does not support concepts on specialized hardware nor emerging resource abstraction methods in Cloud.

Another initiative, specifically targeting the IaaS service model, is the Open Cloud Computing Interface (OCCI) [29]. OCCI defines an interface to support the creation of hybrid cloud environments independently of cloud providers and frameworks. It specifies, in UML, real-world resources and their links, expressed in a similar manner to an OWL [30] ontology definition. However, OCCI targets a user-view perspective of the Cloud, in which resource managers manage traditional homogeneous resources, and hence, do not address interoperability issues caused when multiple services coexist and use diverse abstraction methods and specialized hardware.

By using ontologies, it is possible to generate intelligent decision support mechanisms for addressing interoperability issues in

Download English Version:

<https://daneshyari.com/en/article/6872877>

Download Persian Version:

<https://daneshyari.com/article/6872877>

[Daneshyari.com](https://daneshyari.com)