From Volunteer to Cloud Computing: Cloud@Home

Vincenzo D. Cunsolo, Salvatore Distefano, Antonio Puliafito and Marco Scarpa
University of Messina
Contrada di Dio, S. Agata, 98166 Messina, Italy.
vdcunsolo,sdistefano,apuliafito,mscarpa@unime.it

ABSTRACT

Even though mainly commercial Cloud solutions have been implemented so far, the Cloud computing approach is quickly and widely spreading in open contexts such as scientific and academic communities. Two main research directions can be identified in such context: to provide an *open* Cloud infrastructure able to share resources to the community; and to implement an *interoperable* framework, allowing commercial and open Cloud infrastructures to interact. In this paper we present the *Cloud@Home* paradigm as an effective solution to the problem of building open and interoperable Clouds. In this new paradigm, users' hosts are not passive interface to Cloud services anymore, but they can interact (for free or by charge) with other Clouds, that therefore must be able to interoperate.

Categories and Subject Descriptors: C.2.4 [Computer-Communication Networks]: Distributed Systems; D.2.12 [Software Engineering]: Interoperability; H.3.4 Information Storage and Retrieval: Systems and Software Distributed systems.

General Terms: Design, Management, Standardization.

Keywords: Infrastructure as a service, Cloud, Interoperability, Volunteer Computing.

1. INTRODUCTION AND MOTIVATION

Cloud computing is derived from the *service-centric perspective* that is quickly and widely spreading on the IT world. From this perspective, all capabilities and resources of a Cloud (usually geographically distributed) are provided to users *as a service*, to be accessed through the Internet without any specific knowledge of, expertise with, or control over the underlying technology infrastructure that supports them.

In order to achieve such goals it is necessary to implement a level of abstraction of physical resources, uniforming their interfaces and providing means for their management, adaptively to user requirements. This is done through *virtualizations* [8], *service mashups* (Web 2.0) [7] and *service oriented architectures* (SOA) [4].

A great interest on Cloud computing has been manifested from both academic and private research centers, and numerous projects from industry and academia have been proposed, such as: Amazon EC^2 [1], Microsoft Azure Services Platform [3], Reservoir [6], Eucalyptus [5], etc.

CF'10, May 17–19, 2010, Bertinoro, Italy.
ACM 978-1-4503-0044-5/10/05.

We propose a more "democratic" form of Cloud computing, in which the computing resources of single users accessing the Cloud can be shared with the others, in order to contribute to the elaboration of complex problems. Since this paradigm is very similar to the Volunteer computing one [2], it can be named *Cloud@Home*. Both hardware and software compatibility limitations and restrictions of Volunteer computing can be solved in Cloud computing environments, allowing to share both hardware and software resources. The Cloud@Home paradigm could be also applied to commercial Clouds, establishing an *open computing-utility market* where users can both buy and sell their services.

2. CLOUD@HOME OVERVIEW

The most important issues that should be taken into account in order to implement Cloud@Home can be synthesized as follows:

- *Resources and Services management* - a mechanism for managing resources and services offered by Clouds is mandatory.
- *Frontend* - abstraction is needed in order to provide users with a high level service oriented point of view of the computing system.
- *Security* - effective mechanisms are required to provide: authentication, resources and data protection, data confidentiality and integrity.
- *Interoperability among Clouds* - it should be possible for Clouds to interoperate each other.
- *Business models* - it is mandatory to provide QoS and SLA management for both commercial and open volunteer Clouds (traditionally best effort).

2.1 Basic Architecture

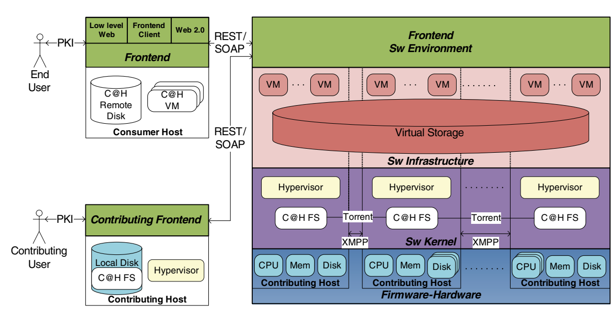

Figure 1: Basic architecture of Cloud@Home.

A possible Cloud@Home architecture that could accomplish the issues above listed is shown in Fig. 1, adapted to the ontology provided in [9]. Two types of users are distinguished in such architecture according to the role they assume in the Cloud: *end users*, if they only interface the Cloud for submitting requests, and/or *contributing users* if they make available their resources and services for building up and supporting the Cloud.

Software Environment - The Cloud@Home software environment implements the user-infrastructure frontend interface. It is responsible for the resources and services management (enrolling, discovery, allocation, coordination, monitoring, scheduling, etc.) from the global Cloud system's perspective. It also provides tools, libraries and API for translating user requirements into physical resources' demand. Moreover, in commercial Clouds, it must be able to negotiate the QoS policy to be applied (SLA), therefore monitoring for its fulfillment and, in case of unsatisfactory results, adapting the computing workflow to such QoS requirements. If the Cloud's available resources can not satisfy the requirements, the frontend provides mechanisms for requesting further resources and services to other Clouds, both open and/or commercial. In other words, the Cloud@Home frontend implements the interoperability among Clouds, also checking for services' reliability and availability.

Software Infrastructure - Two basic services are provided by the software infrastructure to the software environment and, consequently, to end users: *execution* and *storage* services. The execution service allows to create and manage virtual machines. A user, sharing his/her resources within a Cloud@Home, allows the other users of the Cloud to execute and manage virtual machines locally at his/her node, according to policies and constraints negotiated and monitored through the software environment. The storage service implements a storage system distributed across the storage hardware resources composing the Cloud, highly independent of them since data and files are replicated according to QoS policies and requirements to be satisfied.

Software Kernel - The software kernel provides to the software infrastructure, mechanisms and tools for locally managing the Cloud physical resources in order to implement execution and storage services. Cloud@Home negotiates with users that want to join a Cloud about his/her contribution. This mechanism involves the software kernel that provides tools for reserving execution and/or storage resources for the Cloud, and monitors these resources, such that constraints, requirements and policies thus specified are not violated.

Firmware/Hardware - The Cloud@Home firmware/hardware layer is composed of a *"cloud"* of generic contributing nodes and/or devices geographically distributed across the Internet. They provide to the upper layers the physical-hardware resources for implementing the execution and storage services.

3. CLOUD@HOME CORE STRUCTURE

The blocks implementing the functions above identified are pictorially depicted in the layered model of Fig. 2, that reports the core structure of the Cloud@Home server-side middleware. It is subdivided into two subsystems: *management* subsystem implementing the upper layer of the functional architecture, and *resource subsystem* implementing lower level functionalities.

Fig. 3 pictorially depicts the deployment of the Cloud@Home core structure into the physical infrastructure. On top of

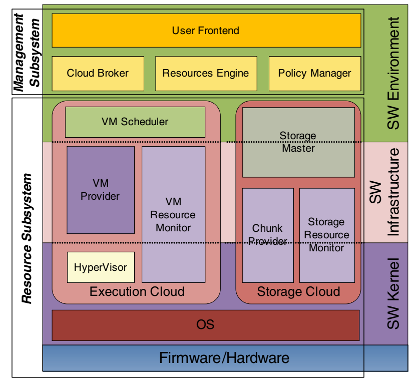

Figure 2: Cloud@home Core Structure Organization

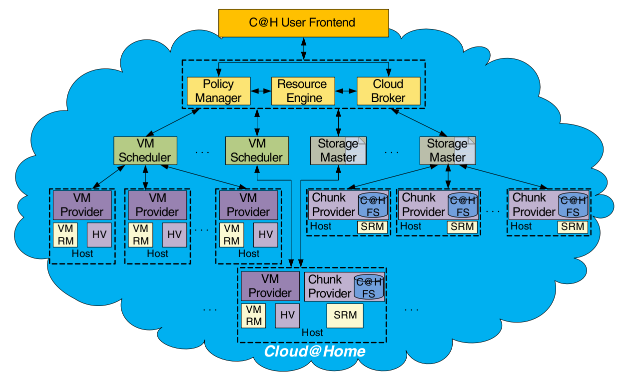

Figure 3: Cloud@home Core Structure Infrastructure Deployment.

such hierarchy there are the blocks implementing the management subsystem, that can be deployed into different servers/nodes, one for each block, or can be grouped into the same node. Below them, VM schedulers and storage masters manage smaller groups (grid, clusters, multi-core nodes, etc) of resources. At the bottom of the hierarchy there are the contributing hosts. Each of such leaves contains the blocks, the software for supporting the specific service for what was enrolled into the Cloud.

4. REFERENCES

[1] Amazon Inc. Elastic Compute Cloud [URL]. Amazon, Nov. 2008. http://aws.amazon.com/ec2.

[2] D. P. Anderson and G. Fedak. The computational and storage potential of volunteer computing. In *CCGRID '06*, pages 73–80.

[3] M. Co. Azure services platform. http://www.microsoft.com/azure/default.mspx.

[4] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz, and B. A. Hamilton. Reference Model for Service Oriented Architecture 1.0. OASIS SOA Reference Model Technical Committee, 2006. http://docs.oasis-open.org/soa-rm/v1.0/.

[5] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The eucalyptus open-source cloud-computing system. In *Proceedings of Cloud Computing and Its Applications*, October 2008.

[6] Reservoir Consortium. Reservoir Project [URL], 2009. http://www-03.ibm.com/press/us/en/pressrelease/23448.wss/.

[7] Tim O'Reilly. What is WEB 2.0, Sep. 2005. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

[8] X. I. VMWare Inc. The open virtual machine format whitepaper for ovf specification, 2007. http://www.vmware.com/appliances/learn/ovf.html.

[9] L. Youseff, M. Butrico, and D. Da Silva. Toward a unified ontology of cloud computing. In *Grid Computing Environments Workshop, 2008. GCE '08*, pages 1–10, Nov. 2008.