Cirrus: A Disruption-Tolerant Cloud

Eleftheria Katsiri[1,2,*]

[1] Department of Electrical Engineering and Computer Engineering
Democritus University of Thrace,
Xanthi, 67100, Greece

[2] Institute for the Management of Information Systems,
Research and Innovation Centre in Information, Communication and Knowledge Technologies, "Athena",
Artemidos 6 and Epidavrou,
Maroussi, 15125, Greece
eli@imis.athena-innovation.gr

Abstract. *Cloud computing* is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, virtual machines, applications, and services) that can be rapidly and *elastically* provisioned, to quickly scale out, and rapidly released to quickly scale in. However, commercially available cloud services such as public grids target the needs for the broader customer base and do not meet the specialized requirements of *real-time, data-centric* applications, such as sensor data aggregation, messaging, media streaming and commodity exchange, that need to process very large volumes of diverse, streaming data in near real time. To make matters worse, end-to-end communication paths between real-time data providers and consumers are no longer guaranteed, due to either node unavailability or service unavailability. The DTN paradigm has shown to promote interoperable and reliable communications in the presence of disruptions, however, is not directly applicable to cloud computing. A new cloud computing model is therefore needed for the above scenarios.

This paper proposes a novel concept, that of a generalized cloud, *Cirrus*, defined as a computing cloud with the following characteristics: (i) abiding by the NIST Cloud Definition, (ii) providing specialized, core Cloud services targeted to real-time, data centric applications, (iii) allowing for the elastic use of Cirrus cloud resources by ad-hoc networks and (iv) allowing for the elastic incorporation of nomadic and/or severely resource constrained devices, in Cirrus. Cirrus is built on top of DTN application-layer extensions, such as the Bundle Protocol (BP). As a result, Cirrus behaves as an "overlay Cloud", elastically forming, expanding and shrinking over networks of dynamic topology that may contain both fixed and ad-hoc infrastructure, thus providing a more fair and de-centralized Cloud Computing solution that is not exclusive to "big players" in the field.

[*] The author is Assistant Professor-elect at the Department of Electrical and Computer Engineering at the Democritus University of Thrace.

Y. Koucheryavy et al. (Eds.): WWIC 2012, LNCS 7277, pp. 350–357, 2012.

1 Introduction

Real-time, data-centric applications such as environmental monitoring, traffic and transport monitoring and disaster management, but also social networking, file sharing and commodity exchange require specialized computational models that process very large amounts of (streaming) data in near-real time while at the same time trying to extract useful knowledge from the data, in order to best serve the user without compromising their privacy. In the data-centric paradigm, it is *timely and useful information* that is most highly valued while for energy-constrained devices, *prolonged device lifetime* also ranks high. This is a shift from the earlier computation-centric paradigm, such as scientific computing, where the focus is placed mainly on computational resources such as CPU and memory. The computational tasks involved in data-centric applications span from mathematical *aggregation* and *logical condition evaluation* to *fusion, data analytics, data mining* and *pattern classification.* Hybrid applications exist also, such as Smart-Grid [8] applications and certain scientific computing applications [11].

Playing a critical role in the world's computing and data storage requirements is *Cloud Computing* that has evolved into a model for enabling convenient, on-demand network access to a shared pool of configurable computing capabilities (e.g. networks, servers, storage, virtual machines, services, and applications) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This is known as *Infrastructure as a Service (IaaS).* Users involved in the study of real-time applications need to re-think their strategies to process, share and store large datasets with the advent of this technological advancement. However, commercially available cloud services such as public grids target the needs for the broader customer base and do not meet the specific requirements of real-time applications such near real-time response, large-scale stream processing, semantic interoperability and often science-class performance and capacity requirements. Furthermore, the available applications that are provided as cloud services, known as *Software as a Service (SaaS)* are general-purpose and are not directly applicable to the computational needs of this domain. Concerns for data security, data governance and reliability have also been expressed in the literature. *Mobile clouds* [2,18] are also emerging, where the user can access cloud services from their smart phone or laptop, while being on the go.

To make matters worse, the proliferation and diversity of wireless communication technology in combination with processor and sensor miniaturization has led to inexpensive sensor networks and mobile devices that can be easily deployed at large scale, from deep space to deep ocean. On the one hand, this has increased both the amount and the diversity of available data, allowing for richer, more useful knowledge and better reliability, on the other hand it has lifted the assumption that end-to-end communication paths between real-time data providers and consumers are guaranteed, thus making the deployment of data-centric applications over these devices, unreliable, ad-hoc and uncoordinated, thereby infeasible in a systematic way.

Delay-Tolerant-Networking (DTN) [9] is a set of protocols that act together to enable a standardized method of performing store and forward communications, in scenarios where end-to-end connectivity cannot be assumed. DTN operates in two basic environments: low-propagation delay and high-propagation delay. In a low-propagation environment such as may occur in near-planetary or planetary surface environments, DTN bundle agents can utilize underlying Internet protocols that negotiate connectivity in real-time. In high-propagation delay environments such as deep space, DTN bundle agents must use other methods, such as some form of scheduling, to enable connectivity between the two agents. The convergence layer protocols provide the standard methods for transferring the bundles over various communications paths. Examples of environments where DTN has made significant contributions include spacecraft [12], military/tactical, some forms of disaster response, underwater, and some forms of ad-hoc sensor/actuator networks. It may also include Internet connectivity in places where performance may suffer such as developing parts of the world.

It is therefore clear that a different model is needed. It should be noted that our approach is inline with NASA's Mobile cloud, as defined in [18] however, our goals are different: Cirrus is focused specifically on the requirements of (personalized) streaming applications, while providing in addition support for environmental applications, social applications and commodity exchange, while NASA's mobile cloud targets mainly mobile phone users and has a much broader scope in terms of integration with existing clouds and applications.

1.1 Aims and Objectives

We define *Cirrus*, a generalized, Cloud Computing model, with the following characteristics:

1. **Abiding by the NIST Cloud Computing model definition** [3] that advocates five essential characteristics (*on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service*) and three service models (*Cloud Software as a Service - SaaS, Cloud Platform as a Service - PaaS, Cloud Infrastructure as a Service - IaaS*).
2. **Providing specialized IaaS, SaaS and PaaS services for real-time, data centric applications.** Such services lay the foundations, i.e. the "plumbing" for data-centric applications that run on the cloud, as SaaS instances. They provide among others, a distributed Event Service, a distributed, federated Data Management Service, a Streaming Service with filtering and aggregation operators, a Personalization Service and a Semantic Interoperability Service. These IaaS services can be used by PaaS instances, in order to develop SaaS instances, i.e. applications, on Cirrus, such as a Personalized Twitter-like [17] Messaging Service, a location-aware Media Streaming service, a Commodity Exchange Service.
3. **Allowing for the elastic use of cloud services by ad-hoc networks.** This is the simpler interaction with the cloud. Here, ad hoc devices can use the above mentioned cloud services in order not to have to develop their

own. For example, a user equipped with a smart phone that can pick up data from ambient sensor networks, can store the data on the cloud. If the user is mobile, then the data should be stored in the nearest fixed-location on the cloud, in order to save battery resources.

4. **Allowing for the elastic incorporation of ad-hoc resources in the cloud.** This level of interaction allows ad-hoc and nomadic nodes to participate in the cloud resources by hosting a part of a Cloud service, resulting in cloud services that are distributed services over dynamic topologies of both fixed and mobile infrastructure. For example, by sharing a VM that is hosted on one mobile device, a second device's computational resources are pooled to those of the first one. The same applies to storage resources.
5. **Is built on top of DTN application-layer extensions, such as the Bundle Protocol.** However, several extensions are required at the Session and Application layers, to realize the above scenarios.

2 Research Challenges

The analysis of the application domain as well as the Cirrus characteristics of Section 1.1 raise a number of research challenges that are discussed in more detail in this section.

2.1 Abiding by the NIST Cloud Computing model definition

On-demand self-service *means that a consumer can unilaterally provision computing capabilities as needed, automatically without requiring human interaction with each service's provider.* We extend the NIST concept of cloud capabilities to include not only physical resources but also logical resources, i.e., Iaas instances, the "plumbing" that was mentioned earlier. We define a generalized virtual machine, the *Cirrus virtual Node - CN* that provides an abstraction over the above capabilities as well as the physical location where they are deployed and can be configured to provide any number of such capabilities on demand.

Broad network access *advocates that capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).* A promising approach towards providing broad network access in Cirrus is to adopt the *Service-Oriented-Architecture -SOA* [6] paradigm for both the supported IaaS, SaaS and PaaS services. SOA provides an interoperable interface for all devices via the state-of-the art XML standard. Other approaches include the development of *Thin Client* software or *Smart Client* software.

Resource pooling *has to do with the fact that the providers computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.* The Cirrus virtual machine serves this purpose well by being able to offer remote, often distributed capabilities, without disclosing their physical location. Another approach here would be to create a CN that supports parallel processing. This is an extended aim of this work.

Rapid elasticity *refers to the ability of rapidly and elastically provisioning capabilities, in some cases automatically, to quickly scale out, and rapidly releasing them to quickly scale in.* On one hand, elasticity is inherent in Cirrus that has the ability to scale on resources that were not previously available, such as mobile devices and sensor networks. On the other hand, in order to realize this feature, more research is required in order to investigate how to best implement elasticity at the VM (CN) level. One successful approach here is that of Amazon's Elastic Cloud (EC2) [5] where VMs can be composed in a flexible manner to form larger VMs and vice-versa. This in turn implies that the Cirrus VM (CN) also needs to be SOA-enabled, in order to be composable with other VMs.

Measured Service *implies an algorithm and its implementation for automatically controlling and optimizing resources use by leveraging a metering capability at some level of abstraction appropriate to the type of service.* In Cirrus, such a measuring service needs to be developed possibly as an embedded service inside the CN in order to monitor both physical and logical resources. One promising approach would be to develop an ontology of measurable resources that caters as much as possible for these differences. Furthermore, the Cirrus Measuring Service may have to aggregate individual results to produce an overall estimate.

2.2 Providing Specialized IaaS, SaaS and PaaS Services for Real-Time, Data Centric Applications.

Infrastructure as a Service - IaaS This is where the core, "plumbing" services belong to. In addition to the Cirrus virtual Node (CN) , IaaS services include but may not be restricted to a distributed Event Service, a distributed File Service, a distributed, federated Data Management service, a Streaming Service with filtering and aggregation operators, a Personalization Service [1] and a Semantic Interoperability service. ***Software as a Service - SaaS*** This category contains the cloud applications that aim to satisfy the specific computational requirements of data-centric applications, ranging from aggregation (of both numeric data, e.g. temperature values, and text data, e.g. *tweets*) to peer-to-peer commodity exchange. Other examples include a Twitter-like messaging application, a Map-creation visualization service, a Media Player service, a Commodity Exchange service. ***Platform as a Service - PaaS*** This category includes tools such as work flow editors that allow for the composition of both IaaS and SaaS services to form other applications, also available as SaaS instances. Examples of such work flow applications include a Personalized Twitter-like, Messaging Service, a location-aware Media Streaming service, a Commodity Exchange service and a privacy preserving Map service. [2]

[1] Personalization technology enables the dynamic insertion, customization or suggestion of content in any format that is relevant to the individual user, based on the users implicit behavior and preferences, and explicitly given details

[2] Such a service can process sensor data gathered by a mobile user by means of their smart-phone in a way that they can still be visualized in a map but without disclosing the identification and associated location of the users that gathered them, thus breaching user privacy.

Here as well, the SOA paradigm appears to be a promising solution for the creation of these services, their discovery and their composition in work-flows. The research issues therefore relate to the design and implementation of the above services so that they are *cloud-enabled*, i.e. service-oriented, providing an external, searchable interface and be distributable over more than one VMs or operate over distributed data. They also may need to be virtualized so that they can be used by multiple users or migrated closer to the data or the user, if required. Furthermore, the above services need to be composable with application services, thus providing added value on existing services.

2.3 Allowing for the Elastic Use of Cloud Services by Ad-Hoc Networks

In this level of integration user devices behave as Thin (or Smart Clients) that connect to the fixed cloud infrastructure either directly, or through a multi-hop path, using an appropriate routing scheme (e.g. epidemic routing). This model allows them to use Cirrus Cloud Services instead of developing their own custom solutions, thus allowing for correctness, standardization and flexibility. Clients can instantiate, use in an elastic manner and later deallocate VMs and associated capabilities on the fixed cloud. Client mobility and isolation both play a significant role in the way cloud services are designed. For example, a user with a laptop in a car that picks up ambient pollution data, can store the data continually on the cloud, each time at the current closest available location in order to save laptop battery. Similar arrangements may need to be made in the case of data streaming from isolated sensor networks that are only connected to the cloud periodically by means of satellite connection or when a user streams video clips or movies from optimal locations in terms of QoS, while traveling on a train. Note that these scenarios do not necessarily mean that other resources are in a data center other mobile devices could serve as resources (e.g. streaming video from nearest passenger) (see next Section).

In order to realize these scenarios, a federated Data Management needs to be designed, possibly using existing solutions such as RODs [1] or Hadoop [10]. Furthermore, nomadic and resource constrained devices should be DTN-enabled, while fixed infrastructure components need to be either DTN enabled (leading to a cloud where the interconnection is DTN-based rather than IP-based allowing for large scale integration with remote data centres) or connected to a DTN switch.

2.4 Allowing for the Elastic Incorporation of Ad-Hoc Resources in the Cloud

This level of interaction involves the hosting a part of a Cloud service by nomadic devices resulting in cloud services that are fully distributed over dynamic topologies of DTN-enabled nodes. For example, the Cloud's Distributed Storage Service can be built on top of mobile nodes allowing for the construction of distributed applications that use the MAP-REDUCE [15] paradigm. The Data

Aggregation Service as well as the Twitter-like Messaging Service can also be distributed. Aggregation and filtering algorithms can be integrated with the publish-subscribe[7] event paradigm and the performance of this "marriage" over an infrastructure-less DTN network will be investigated. Commodity Exchange involves the development of distributed algorithms that enable the exchange of goods in a fair-trade manner.

Inverting the problem, it is also interesting to investigate the placement of online stream processing and filtering tasks on special devices that behave like "data mules" running on a scheduled trajectory and providing connectivity between otherwise isolated components, i.e. UAV, satellites, submarines. Such devices are also relatively secure from threat, being physically airborn (or emerged).

In terms of research challenges, these include: a light-weight version of the CN hypervisor node (LCN) that can run on the above devices; an integration of the LCN and the DTN Bundle Protocol; investigating how the SOA paradigm can be applied to bandwidth-deprived nomadic devices scenarios (some work has already been done in the area of sensor networks, in the scope of an EU project [13]); An appropriate ontological approach trying to cater for semantic differences among data and services; novel privacy-preserving, forensic algorithms for the extended Cirrus cloud.

3 Conclusions and Future Work

Summarizing the above, mobile users of modern data-centric applications, expect computable, useful answers and they expect them "now"! At the same time, current technology suffers from resource poverty and lack of maturity. Although the Cloud industry is emerging it is only at the beginning of standardization. There exist too many choices of mobile devices and sensor networks, each supporting different operating systems and all prone to disrupted service. Cirrus aims to provide a solution for the above issues by investigating both middleware and algorithmic approaches. In terms of middleware we are considering: the design and implementation of the virtual Cirrus Node (CN) and the lightweight Cirrus Node (LCN), the integration of CN with a SOA technology, such as OSGI [4], the integration of CN with an event technology such as JMS [16], and an integration of all the above with the Bundle Protocol. Also we plan to investigate Thin or Smart Clients, and the composition/decomposition of VMs to allow for elasticity. In terms of algorithms, we are considering optimisations that related to performing distributed aggregation, personalized filtering and media streaming integrated with publish-subscribe over the DTN routing protocols, as appropriate, Map-Reduce processing algorithms over the same protocols, elasticity and service measurement algorithms, and interoperability mechanisms, privacy-preserving forensic algorithms. Cirrus's potential impact is significant. Apart from promoting quality and richness of computed knowledge, it enables functionality similar to that of the Internet of Things [14], such as Machine-to-Machine intelligence, augmented reality, location-based and personalized services.

Acknowledgments. This work was carried out under the auspices of the Space Internetworking Centre (SPICE) Project, which is led by Prof. V. Tsaoussidis of the Democritus University of Thrace.

References

1. IRODS:Data Grids, Digital Libraries, Persistent Archives, and Real-time Data Systems,https://www.irods.org/index.php/IRODS:Data_Grids,_Digital_Libraries,_Persistent_Archives,_and_Real-time_Data_Systems
2. Mobile Cloud Computing: Devices, trends, issues, and the enabling technologies, http://www.ibm.com/developerworks/cloud/library/cl-mobilecloudcomputing/
3. The NIST Definition of Cloud Computing. National Institute of Standards and Technology 53(6), 50 (2009)
4. OSGI Alliance, http://www.osgi.org/About/Technology
5. Amazon Elastic Compute Cloud, http://aws.amazon.com/ec2/
6. Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: Web Services Architecture. Technical report, W3C (2004)
7. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M.: The Many Faces of Publish/Subscribe. ACM Computing Surveys 35(2), 114–131 (2003)
8. Smart Grid, http://energy.gov/oe/technology-development/smart-grid
9. Delay Tolerant Networking Research Group, http://www.dtnrg.org/wiki/home
10. Hadoop, http://hadoop.apache.org/
11. NASA Nebula in Action: Cloud Computing Case Examples, http://nebula.nasa.gov/media/uploads/nasa-nebula-in-action.pdf
12. Nichols, K., Holbrook, M., Pitts, R.L., Gifford, K., Jenkins, A., Kumzinsky, S.: Dtn implementation and utilization options on the international space station. In: SpaceOps 2010 Conference "Delivering on the dream", Huntsville, Alabama, Springer (April 2010)
13. Leguay, J., Lopez-Ramos, M., Jean-Marie, K., Conan, V.: An efficient service oriented architecture for heterogeneous and dynamic wireless sensor networks. In: 33rd IEEE Conference on Local Computer Networks, LCN 2008, pp. 740–747 (October 2008)
14. The Internet of Things. Executive Summary. Itu internet reports (2005), http://www.itu.int/osg/spu/publications/internetofthings/InternetofThings_summary.pdf
15. MapReduce: Simplified Data Processing on Large Clusters, http://research.google.com/archive/mapreduce.html
16. Java Messaging Service Tutorial, http://docs.oracle.com/javaee/1.3/jms/tutorial/1_3_1-fcs/doc/jms_tutorialTOC.html
17. Twitter, https://twitter.com/
18. Warner, S.A., Karman, A.F.: Defining the mobile cloud. NASA IT Summit 2010 (August 2010)