Constructing Community Clouds for Disaster Management in Smart Cities

Mohsen Amini Salehi
Center of Advanced Computer Studies (CACS)
University of Louisiana Lafayette
{amini}@cacs.louisiana.edu

***Index Terms*—Community Clouds, Disaster Management, Smart Cities.**

I. Background and motivation

Forward looking cities across the globe must be robust against natural disasters. Such disasters demand immediate, targeted, and large-scale response from government and decision makers so that loss of life and the disruption to the community is minimized.

Applications, such as evacuation traffic modeling [6] and evacuation fuel demand prediction modeling [7], that provide real-time analytical information for decision makers in emergency situation generally utilize data from traffic sensors, traffic cameras, and disaster management systems. Therefore, such applications are often resource- and data-intensive.

Cloud computing infrastructure, in a smart city, plays a critical role in providing resources for disaster and emergency management. However, in the emergency situation, Cloud infrastructure is frequently overwhelmed by the load of disaster management applications. In addition, it would be cost-prohibitive for many cities to maintain a large Cloud infrastructure that is over-provisioned to respond to rare disaster events. Several organizations within a smart city, on the other hand, have plenty of underutilized resources, and are willing to share their resources in disastrous situation.

The data-intensive, dynamic, and real-time nature of disaster management applications motivates organizations in smart cities to create community Cloud models that enable them to share hardware, software, network, and storage resources on demand. A community Cloud is a deployment model where the Cloud infrastructure is distributed across several organizations (i.e., datacenter providers) where a high bandwidth and robust network is available within a local jurisdiction. In a community Cloud the resources are offered based on donation, mutual trust, and shared interests. Such model makes it possible to implement an infrastructure for disaster management that can scale in response to increasing demands of the relief operations and then shrink back in times of normal operation. Any such solution has to deal with the common challenges of ensuring scalability, transparency, performance, and security. Being motivated by this scenario, this project aims to develop techniques and tools for efficient use of community Cloud features to solve the challenges of disaster management in a smart city.

It is noteworthy that, the community Cloud differs from a public cloud (such as, Amazon AWS, Google Cloud) that offers infrastructure, platform, and software as service. This also differs from a hybrid cloud model where a private cloud is supported by resources from another public Cloud during peak demand periods. There is a mutual trust between the shareholders of a community Cloud. Besides, there is high-bandwidth and robust interconnect networks between the communities which makes the large-scale communication feasible.

There are several ongoing commercial and open cloud platform development efforts in the United States, such as SCOPE from Boston University, IBM's pilot in Dubuque, Iowa [4], and several others outside the U.S. [5] that provide analytics within the smart city context. The key differentiating aspect of this proposal from existing efforts is being tailored for disaster management and its ability to deliver real-time distributed data-intensive (Big data) analytics services across multiple datacenters.

II. Proposed research work

In the community Cloud project, our aim is to harness the capacity of several datacenters exist within a city to enable processing of resource- and data-intensive disaster management applications. Users of this system expect to execute their analytical applications given their Quality of Service (QoS) requirements. Receiving a certain amount of bandwidth or meeting the application deadline for the application completion are examples of such QoS requirements. In addition, users can potentially have privacy constraints attached to their applications. For example, users may have restrictions on the datacenters that cannot be utilized.

To offer analytical services on a community Cloud, users need a middleware that given the user requirements and constraints, discovers appropriate providers, prepares them for executing user applications, schedules the application on the acquired resources optimally, and finally returns the results back to the users. More specifically, such middleware has the following responsibilities:

- Receiving users' applications along with their QoS requirements and privacy constraints;
- Seamless resource acquisition from various datacenters in the community Cloud based on the application QoS requirements and constraints;
- Preparing the acquired resources for the users' application;

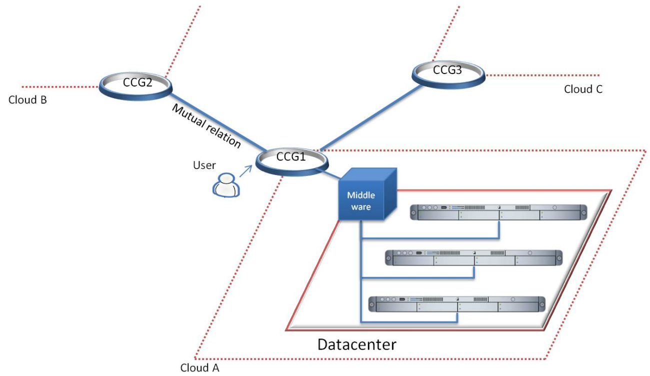

Fig. 1: Proposed architecture for community cloud for disaster management.

- Scheduling the users' applications on the acquired resources efficiently, based on the characteristics of the applications (e.g., MapReduce, stream processing) and to meet their QoS demands;
- Monitoring the execution of the applications, recovering possible faults, and returning the results to the users.

In the community Cloud, we consider that a user has access to a cloud resources. However, she requires contribution of other Cloud providers in the community to be able to meet the QoS requirements of her applications. To encourage participation of providers in a community Cloud, we need to ensure that the emerging community Cloud would not interfere with the normal operations of participant Clouds.

Considering these restrictions, we propose an architecture that does not interfere the normal operations of other Cloud provider. Figure 1, provides a bird's eye view of the architecture for the community Cloud. The architecture introduces a gateway component, called Community Cloud Gateway (CCG), that communicates with the Cloud middleware in each provider and acquire resources from them. CCG includes the arrangement of a Cloud provider with its peers. In fact, CCGs form an overlay network on top of the robust underlying network in a smart city. As we can see in the architecture, the CCG is the entry point to the community Cloud where users can submit their applications, QoS requirements, and privacy constraints. Upon receiving the application, CCG verifies if the application can be executed with the local Cloud resources or more resources should be acquired from the community. In the latter case, CCG works with its peers at different Cloud providers based on their predefined agreement for resource sharing. To bridge any configuration heterogeneity at the local middleware of participating Clouds (e.g., OpenStack [3], Open Nebula [2]), the CCG is designed to be pluggable and can be extended to support any variation of local Cloud middleware. Once the CCG obtains required resources from different Clouds, it allocates the user application on them by considering the application QoS constraints (e.g., deadline). In case of data-intensive applications, the allocation should consider optimization issues regarding data locality as well.

III. Potential Impact in Advancing the Smart Cities

The subject matter of the project helps in enabling smart cities to be robust against natural disasters. It provides a computational infrastructure for smart cities that can manage peak computational demand periods during natural disasters. The proposed architecture will enable real-time or near real-time processing of resource- and data-intensive applications that are common in disaster and emergency management.

The National Incident Management Systems and Advanced Technologies (NIMSAT) Institute [1] in conjunction with University of Louisiana at Lafayette works closely with several local, state, and federal homeland security and emergency management agencies on various data management and analytics projects. Several of these projects use real-time data sources. The institution has developed an evacuation traffic and evacuation fuel demand prediction model for the state of Louisiana's regional evacuation that uses data from traffic sensors, traffic cameras, and disaster management systems. The proposed community Cloud system enables execution of such applications by providing sufficient resources with high-bandwidth and low failure rate.

References

[1] Nimsat. http://www.nimsat.org/, accessed Feb 25, 2015.

[2] Open nebula. http://opennebula.org/, accessed Feb 25, 2015.

[3] Openstack. https://www.openstack.org/, accessed Feb 25, 2015.

[4] Smart water pilot study report. http://www.cityofdubuque.org/DocumentCenter/Home/View/3116, accessed Feb 25, 2015.

[5] Smartsantander project. http://smartsantander.eu/, accessed Feb 25, 2015.

[6] V. Dixit and B. Wolshon. Evacuation traffic dynamics. *Transportation research part C: emerging technologies*, 49:114–125, 2014.

[7] R. Gottumukkala, J. Zachary, B. Kearfott, and R. Kolluru. Real-time information driven decision support system for evacuation planning. In *In Proceedings of 2nd IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pages 206–209, 2012.

Mohsen Amini Salehi received his Ph.D. in Computing and Information Systems from Melbourne University, Australia, in 2012. He is a visiting Assistant Professor at the Center of Advanced Computer Studies (CACS) at University of Louisiana Lafayette, LA, USA. Since 2004, he has worked on different aspects of Cloud computing, such as load balancing, virtualization, energy-efficiency, Cloud federation, and Cloud security. His PhD thesis has been on supporting contention between requests in federated Cloud environments. Currently, he is working on user-friendly and secure authentication for Cloud services. You can find more detailed information about his research interests and background in his web site: http://dataengineering.org/aminis/