COMMUNITY CLOUD
Cloud Computing for the Community

Marte K. Skadsem
Northern Research Institute, P.O. Box 6434, TromsøScience Park, N-9294 Tromsø, Norway

Randi Karlsen
Dep. of Computer Science, University of Tromsø, N-9294 Tromsø, Norway

Gordon Blair, Keith Mitchell
School of Computing and Communications, InfoLab21, Lancaster University, Lancaster, LA1 4WA, U.K.

Keywords: Cloud computing, Community cloud.

Abstract: Commercial clouds have gained a lot of popularity because of easy access to high quality services. Private clouds and community clouds have emerged as an answer to the security concerns raised by commercial clouds. A community cloud is build up by resources made available by the community members and provides services adapted to the needs of the community. In this paper we present a community cloud design that focus on the properties of the social bindings within a community and look at how the characteristics of a community can effect the cloud.

1 INTRODUCTION

Clouds and Cloud Computing have gained a lot of popularity among enterprises due to their easy use, low administration costs, and an illusion of access to endless amounts of resources. But moving data and computations to commercial cloud services raises many questions around location and ownership of your personal data. Where is your data stored? Who owns your data when it is not stored on your own computer? What happens if the vendor you buy or use services from closes down? Will you have a chance to get your data back? Will it be sold to other cloud providers?

In this paper we are proposing a community cloud design that aims to alleviate some of these concerns. In a community cloud environment, small communities share their existing resources to form a cloud that provide services for user generated data. We define a community as a group of people connected together through a common interest, like a village, family or interest group. Resources that a community cloud service is able to offer include storage, processing power and application sharing.

Within a community cloud scenario, the cloud itself is owned, operated and controlled by the community themselves who may be regarded as a type of social network, and the properties of the social relationships within the network can be utilized in the cloud design and operation. To achieve this a community cloud design is open for its members and a low barrier to entry must be adopted.

The outline of this paper is as follows: In section 2 related work is presented. Our approach to a community cloud is presented in section 3. Section 4 describes the architecture and design of the community cloud, and the paper is concluded in section 5.

2 RELATED WORK

The main motivations for moving computation to the cloud are cost and the promise of easily accessible, almost endless amount of resources. Users of cloud computing do not need to buy, administrate and maintain large data centers, clusters or grids themselves as this is provided by the cloud. Most cloud providers offer their services on a pay-per-use model in which

K. Skadsem M., Karlsen R., Blair G. and Mitchell K..
COMMUNITY CLOUD - Cloud Computing for the Community.
DOI: 10.5220/0003389404180423
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (CLOSER-2011), pages 418-423
ISBN: 978-989-8425-52-2

you only pay for the resources you use at any given time. This gives good elasticity as clouds will automatically expand when more resources are needed and contract when the demand is lower. A good overview of cloud computing can be found in (Armbrust et al., 2009).

Moving data and computation to the cloud is convenient, however, this convenience comes at the cost of control. Control over the network, implementation, interfaces etc. stays with the cloud provider and the customer is bound to the terms of services with the provider. Many are pointing out the danger of data lock-in in clouds, among them the founder of the Free Software foundation, Richard Stallmann (Johnson, 2008). Since, in general, most of the APIs for cloud computing services are proprietary, moving from one cloud service provider to another is not trivial (Armbrust et al., 2009) and getting data out is not easy, if possible at all. This prevents many potential customers from moving to the cloud. Another problem with the clouds is that they are public and do not offer any security for ensuring privacy (Wood et al., 2009). For enterprises it is crucial to know that their data is not easily attacked or leaked to third parties.

Because of the security concerns, private clouds, sometimes called internal clouds, have emerged as clouds behind a firewall that are not globally accessible. Private clouds are by some not recognized as "real clouds" as it is not possible for the enterprise who owns the private cloud to outsource the heavy computation and maintenance of a large data center. Hence private clouds may not offer as good and elastic services as a commercial cloud. The CloudNet project (Wood et al., 2009) is one attempt to merge private clouds with commercial vendor provided clouds by utilizing Virtual Private Networks.

In (Marinos and Briscoe, 2009) a community cloud is presented as a virtual data center made up of available resources of networked personal computers which are self configurable and adaptable. The basic architecture comprises of three layers where the core infrastructure is a Peer-to-Peer (P2P) network of virtual machines. As the nodes participate both as contributors and consumers, a "community currency" is needed that ensures that a node has to contribute in order to consume (Marinos and Briscoe, 2009). The application areas the authors envision are community hosted versions of Wikipedia and YouTube.

From a user's point of view the community cloud does not look any different from an ordinary cloud. The distribution and underlying organization of the cloud is not visible outside the cloud. The user will have to know about the community currency and that contribution of resources to the cloud is necessary for consumption, but except for that the community cloud looks just like any other cloud.

The authors of (Marinos and Briscoe, 2009) are concerned about the control that cloud computing providers have and this is the main motivation for proposing the community cloud computing approach. While we share the same security and privacy concerns we are also interested in supporting cloud based services within communities that are poorly connected to the Internet. In many European countries there are a significant proportion of the population whom do not have high-speed access to the Internet, especially in rural communities. In the U.K, for example, the term 'final third' has recently been coined and relates to the fact the almost 30 percent of the population are unable to achieve speeds of 2Mbps. Therefore, our approach to the community cloud differs in that we want a more open version of the cloud with more focus on the properties of the community and inclusion of the community.

3 COMMUNITY CLOUD

3.1 Community Clouds vs Commercial Clouds

We use the term "community cloud" to describe *private clouds adapted for smaller environments like a local community, a school or a home*. The difference between the commercial vendor clouds and community clouds are shown in figure 1 and figure 2. In a commercial cloud users connect to the cloud, put data in it and get data and/or results back. The users do not take part in the cloud in any other way, and do not know anything about how the cloud is constructed or where it actually is located (figure 1). In the community cloud a group of users connects their computers to form the basic cloud infrastructure. The users connect and use the community cloud much in the same way as they do with a commercial cloud, but the cloud infrastructure is open and controlled by the participating users (figure 2).

In a community cloud each member of the community contribute with resources to the cloud infrastructure, forming a Peer-to-Peer (P2P) network. Information about the type and amount of available resources is open and known to all participants at all time. The community cloud will provide services for user generated data based on the needs of the community. Because the infrastructure is open it is easily controlled by the community, and the community cloud will be adaptable to offer new services as

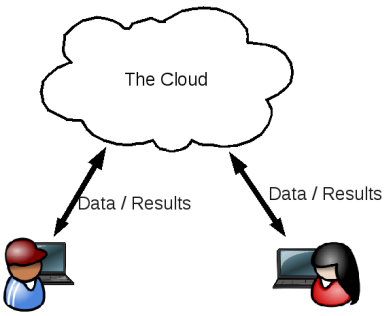

Figure 1: The commercial cloud. The users put data to the cloud and get data and/or results back.

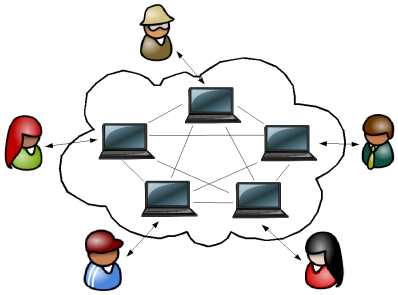

Figure 2: Members of a community connect their computers to form a cloud.

needed. The social relationships within a community will set the premises for the community cloud properties. We will look more at this in section 3.3

3.2 Illustrative Applications

Before we go into a more detailed description we would like to give two examples of how a community cloud may be used.

3.2.1 Editing Video in Low Resolutions

Consider the situation where a group of people are making a film together. After shooting the film they have lots of raw material to work on. All this raw material could be terabytes in size, so editing it requires lots of processing power and storage.

The members of the group do not live in the same town. They have normal workstations, and their Internet connections vary in speed and quality. The cutting and editing of the film is a collaboration project, so each of the group members must be able to download and edit the film as a whole or in parts. In order to do this efficiently, they form a community cloud.

The raw material is distributed and stored among the group members' computers by adding it to the community cloud. Then it is transcoded to a file format with lower resolution. The group members will then work on this smaller file to edit and finish the film. At some point, for instance when the whole film is finished in the low resolution file format, the changes are propagated to the raw film.

In this example the community cloud will offer storage and processing services. The film is replicated and stored on the computers of the different group members. The processing tasks, i.e., the transcoding and editing, is done on the computers that are best suited. A computer with a slow CPU will for instance not be assigned heavy processing. The community cloud will take care of the distribution of replicated data and processing tasks, and it will make sure all group members are offered the same services despite different quality in network connections.

3.2.2 Online Message Board for a Village

A small village wants to share an online message board where local activities can be announced. They also want a photo sharing service for photos from these activities and the daily life in the village. The village has a slow connection to the Internet, and not all villagers have proper Internet access in their homes. Because of this, existing services on the Internet cannot be used, so the villagers set up a local network in which they all contribute to form a community cloud.

There are a few people in the village who commit themselves to never turn off their computers so it is possible to always access the data in the cloud. In addition the local school dedicates one of their computers to the cloud. The rest of the villagers will contribute with resources when their computers are online the local network.

Because of the committed computers the community cloud is always accessible. All data in the cloud will be stored on these computers. Data is also replicated on the computers that are not always connected. When users connect to the community cloud from their computers, the computers will become part of the cloud thus making the cloud able to offer better accessibility of the replicated data. When more users connect, the accessibility to the community cloud is not reduced since every connected user contribute with more resources and replicas.

While the villagers wants to contribute with resources for the cloud, they are concerned about abuse and protection of their private data. The community cloud will have to run isolated on each participating node and there must exist controlling systems for handling abuse of the cloud.

In this example the community cloud will offer a local network with storage services. The community cloud will also offer application services like photo

sharing and a message board. The services will be available from all computers in the village that are connected to the local network even if they do not have an Internet connection.

3.3 Communities

As we can see from the examples above, quite different communities can make use of a community cloud environment and we will now take a closer look at the properties of the communities.

As mentioned in the introduction, we define a community as *a group of people that are connected through a common interest*. This implies that a community cloud can consist of *heterogeneous resources* that are *geographically spread* and have a *variable network connection* between them. It also implies that the people forming a community to a certain degree know and *trust* each other.

The community is a social network where all the users participate with whatever resources they have available. The underlying distribution and heterogeneity among resources are visible for the users. The users will have access to a larger amount of resources than what they have with only their own computer, like in commercial clouds, but at the same time they know the resources are available with the properties and limits of the community.

When the community is your social network, or parts of your social network, the trust and nature of the relationships between the members will make the community currency used in (Marinos and Briscoe, 2009) redundant. The members trust that everyone participate with as much resources as they can because they all have a common goal, or interest, in making the community cloud as good as possible. Participating in a community cloud puts expectations and obligations on the users. If a user consumes a lot of resources but provides little to the community, the other users will cut the "free rider" off the community cloud. Hence everyone should feel obligated to participate and will act to the best of the community.

3.4 Properties and Challenges

We will now look at the properties and challenges for the community cloud based on the requirements from the application examples and the properties of the communities.

Services the community cloud will offer are *storage, processing and applications*, depending of the needs of the community. The community cloud will hence offer Software as a Service (SaaS) to the users, but it will also offer Infrastructure as a Service (IaaS) for those wanting to implement and adapt their own services.

The community cloud must be made from the heterogeneous resources that the community already has available. Also, the community cloud architecture must be able to handle varying network speed between the participating machines and also that machines will become unavailable for periods of time because users turn them off.

As we have seen, communities differ greatly in "personality". Although the services used by different communities are the same, the communities will differ in available resources and so the services will have to be adaptable for this. Building this kind of infrastructure is challenging.

The key for solving these challenges lies in the opportunity given by the social relationships and the natural trust within a community. Different communities will have different levels of trust. An *open infrastructure* where knowledge about the environment is available for all participants at all time is strengthening the trust. The trust is also strengthened by the common interest, or the common goal, shared in the community.

An important property for the community cloud will therefore be openness and common knowledge of the environment that is accessible at any time for all participants. This knowledge include what kind of resources that exist in the community cloud, how much is currently available, but also statistics of each user's usage of the cloud. The usage statistics will help detect if there are any "free riders" using the community cloud.

4 ARCHITECTURE AND DESIGN

4.1 Community Cloud Design

Figure 3 shows the overall design of the community cloud architecture. At the bottom of the figure we find the existing resources of the community. Depending on the community this can be everything from ordinary workstations, to PlayStation. How much and what kind of resources, i.e. how much storage and processing power, these can offer to the community cloud are reported to a community resource manager. The community resource manager will coordinate the available resources to operate as a cloud. Services can be built on top of the community cloud based on the resource information available through the community resource manager's service interface. Users will then be able to use the services their community cloud offer.

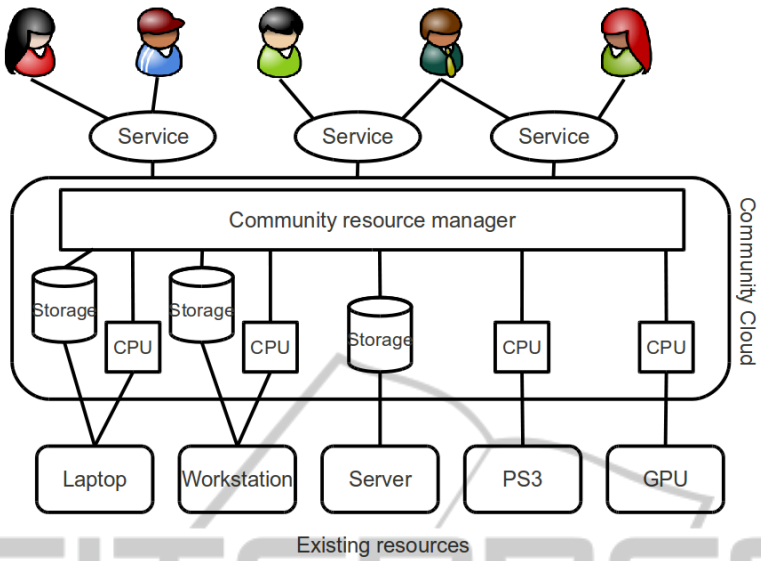

Figure 3: The community cloud design. The existing resources reports their available resources to a community resource manager that coordinates the available resources and offer an interface which services can use.

Grid and P2P networks are possible alternatives for the community cloud design. Because of the dynamic and heterogeneous environment a community represents we will base the cloud on a P2P architecture.

4.2 The Community Resource Manager

The community resource manager keeps control over cloud participants, available resources and the properties of these resources. It also keeps an overview of the applications the cloud offers to its users, and it knows the current status of the whole cloud. Based on this information the community resource manager is able to map the applications' requested resources to the available resources.

In order to do this mapping the community cloud needs to be able to describe requirements and available resources. I.e. we need to describe applications, data, storage and processing power. For this we will use JSON (JavaScript Object Notation), a lightweight, text-based, language-independent data interchange format (Crockford, 2006).

A P2P architecture implies a fully distributed system, however the community resource manager's tasks are better handled centralized on a single node. To make sure the community resource manager does not represent a single point of failure, we will make it soft state, similar to the master in Google File System (Ghemawat et al., 2003). This means that any participating node can be elected as community resource manager as it will rebuild the required state when selected. It also means that the community resource manager will not represent a single point of failure as the cloud will not collapse if the node running the community resource manager fails.

In order for the community resource manager to be soft state, its state must be re-produceable by any participating node. This means that the state is distributed in the cloud, and that each node knows a part of the cloud's current state. The community cloud will use round robin DNS. When someone contacts the system, they do not know which specific node they are talking to. The node that establish contact with a user is caching all data and information it gets while responding to the user's request. When it gets a request it cannot handle, it asks the community resource manager for help.

4.3 Architecture

The community cloud architecture is based on an existing storage and processing system made of a P2P based storage network, called DeStore, and its processing interface, DeLight.

In DeStore several heterogeneous computers collaborate to provide a replicated storage area that can be exported as a single storage service (Borch and Heggelund, 2007).

Key features of the storage network are:

- Designed to run on commodity hardware. It is possible to run DeStore on hardware already existing in a community.
- Easy to extend the storage capacity on demand. When more storage is needed, new nodes are just added to the network. This is possible because

DeStore is a self-organizing P2P network.

- Policy based replication and replication level is recursively configurable for each directory.

DeStore is composed of two different types of nodes; slaves and masters. The master performs synchronized jobs within the network. This includes locking/unlocking operations, caching of authentication and giving replication jobs to slaves. A slave, the basic DeStore node, stores and replicates data and can provide HTTP or WebDAV access to the data. Requested data that is not available locally on a slave will be fetched from a replica and provided to the requester. Hence the requester (a WebDAV enabled client) does not need to be aware of the underlying distribution.

The community resource manager in the community cloud corresponds to DeStore's master node.

DeLight (Skadsem et al., 2009) extends DeStore to include processing on the stored data. DeLight takes advantage of DeStore's support for heterogeneity by building a network of nodes able to either store data, process data, or both, offering a storage network with integrated processing.

Key features of an integrated storage and processing network:

- Utilization of each node's individual properties so that the nodes are used to what they are good at.
- Data is processed at the same place or close to where it is stored.This way bandwidth is saved and latency can be avoided.
- Adaption to individual user demands by offering applications an interface for adding and running processing tasks on data stored in DeStore. This is done by making DeStore able to handle plug-ins. A plug-in manager running on each node is responsible for registering and enrolling new plug-ins in the system. Plug-ins can be added and removed by user applications during run-time, and the changes are migrated to all participating nodes.

The integrated storage and processing system is a good base to build a community cloud from since it posses many of properties needed. By extending the master node it will be able to fill the tasks of the community resource manager in the community cloud design. The community resource manager will then do all the work of the master node, but also keep definitions on how to describe the community cloud resources and how to represent information about the infrastructure so that openness and common knowledge about the environment can be achieved.

5 CONCLUSIONS

In this paper we have presented a community cloud that focus on the properties the social relationships within a community give to the cloud design. The community cloud can consist of heterogeneous resources that are geographically spread and have different network connections. The community cloud offer storage, processing and application services adapted to the needs of the specific community. The cloud design is based on openness to the knowledge of the environment, meaning that information about the infrastructure and its resources are available to all the community members at all time. An integrated storage and processing network serve as the starting point for the community cloud architecture.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2009). Above the Clouds: A Berkeley View of Cloud Computing. White paper, UC Berkeley Reliable Adaptive Distributed Systems Laboratory.

Borch, N. T. and Heggelund, A. (2007). A read/write distributed web server. In *Proceedings of the Second International Conference on Internet Technologies and Applications (ITA07)*, Wrexham, Wales.

Crockford, D. (2006). The application/json Media Type for JavaScript Object Notation (JSON). RFC 4627 (Informational).

Ghemawat, S., Gobiof, H., and Leung, S.-T. (2003). The google file system. In *SOSP'03: Proceedings of the nineteenth ACM Symposium on Operating Systems Principles*, pages 29–43, New York, NY, USA. ACM Press.

Johnson, B. (2008). Cloud computing is a trap, warns GNU founder Richard Stallman. *The Guardian, guardian.co.uk*. webpage, http://www.guardian.co.uk/technology/2008/sep/29/cloud.computing.richard.stallman.

Marinos, A. and Briscoe, G. (2009). Community cloud computing. In Jaatun, M. G., Zhao, G., and Rong, C., editors, *Cloud Computing, First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings*, volume 5931 of *Lecture Notes in Computer Science*, pages 472–484. Springer.

Skadsem, M. K., Borch, N., and Karlsen, R. (2009). DeLight: A Peer-to-Peer storage and processing system. *Internet and Web Applications and Services, International Conference on*, 0:97–101.

Wood, T., Shenoy, P., Gerber, A., Ramakrishnan, K. K., and der Merwe, J. V. (2009). The case for enterprise-ready virtual private clouds. In *Proceedings of the Workshop on Hot Topics in Cloud Computing (HotCloud'09)*, San Diego, CA. USENIX.