1 Introduction

In recent years, the amount of data generated has been increasing steadily as it becomes more viable for companies to gather large quantities of data. This has had the effect of ever increasing demand for large computational tasks required to analyze the date and provide useful information. At the same time, expectations of digital based services have been raising and it is not uncommon to expect availability rate of close to perfect. Such recent developments have led to the increased use of the 'cloud' platform.

The aim of this essay is to explore the underlying operating system used to support the cloud. The idea of a cloud and its meaning will be explored and linked to how distributed networks fit in. Additionally, requirements for an operating system for the cloud will be explored and the main aspects such as virtualization and load management will be discussed. Furthermore, focus will be given to existing cloud frameworks and platforms. Finally, a recommendation for future cloud framework providers will be given.

2 Cloud Computing

Based on the National Institute of Standards and Technology, cloud is defined as *"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources [] that can be rapidly provisioned and released with minimal management effort or service provider interaction."* This provides an indirect specification of features a cloud service should have. Firstly, we would expect it to be easy to use with a well defined API, ideally common for a variety of cloud services to allow clients to transfer data effectively. Secondly, it should have a 'on-demand' nature where the cloud is able to be used when needed and be able to scale as needed when more computation power is required. Thirdly, it should be able to terminate the service as needed without the need for extensive management.

3 Distributed Computing

The term 'cloud' is often used as a synonym for a distributed computing network. Such network is designed to provide improved computing capability as opposed to a single server unit. George Coulouris defines distributed system as *"one in which components located at networked computers communicate and coordinate their actions only by passing messages"*. In essence, a distributed network is composed of computers designed to work together and make computationally heavy tasks faster. Generally, such network is designed to appear as a single unit on the outside in order to simplify access. For example, sending an HTTP request to the

Twitter API will be done using only a single end point while the distributed network will decide which of the network components will do the computation. This helps reduce overload as tasks can be scheduled to optimize computation time and make future requests respond immediately rather than wait for the previous task to finish. Therefore, distributed computing is essential for effective cloud systems as they may require multiple users working at the same time.

4 Cloud Operating System Requirements

Given the discussion about the features of a cloud operating system, it becomes clearer that there is a large number of similarities to a standalone server. For example, it is required to be able to schedule processes and manage them such then no takes over the computation time and/or starves. On the other hand, we require client tasks to be cross platform, easily portable and completely secluded from other users in the system. However, the complexity of the scheduler is much greater for distributed networks. *"Task reallocation involves more than simply moving depended data. Clouds rely heavily on virtualization and thus sometimes in orders to execute tasks VMs with certain characteristics need to be created"* (Handbook on Cloud Computing, 2010). One way to achieve portability is through virtualization.

4.1 Full virtualization

In a full virtualization scenario, a *hypervisor* - a version of an operating system - is responsible for the creation and termination of virtualized workspace as well as monitoring of the resource utilization. The hypervisor is aware of all computers in the distributed network as well as all currently running virtualized instances and uses this information to determine the *load* of each server. The load is essential in the cloud process as it provides information on how much each virtual machine is able to expand or contract to accommodate for scaling. In the case of a very high load the hypervisor can ensure the virtual machine is transferred onto a different server in the network with a lower load and provide extra room for scaling. This is a form of scheduling the distribution in the network and is similar to scheduling processing in a machine with non-distributed network. Additionally, if a Linux based hypervisor is used, the maturity of the linux system and support for a large variety of drivers can be leveraged.

Full virtualization method is used by Amazon's AWS and has been very successful. It is highly independent of the underlying hardware and a seemingly unbound access to resources. Despite its success, full virtualization suffers from large overheads. LinuxJournal, 2013, claims full virtualization requires between five to ten minutes to get up and running. This is a vast improvement compared to a full server deployment which is somewhere between eight to twenty-four hours of set up time.

4.2 Para-virtualization

In the para-virtualization scenario, the hypervisor is responsible for creation and termination of virtual machines but does not manage monitoring of the virtual machines. Instead, the virtual machines themselves are made aware of other virtual machines and resource management is done by each one. The potential benefit of para-virtualization is a reduced resource usage for the hypervisor. In practice, this approach is not widely used is said to be gaining focus *(HowStuffWorks, 2013)*.

However, both para and full virtualization fail to utilize resources allocated to a virtual machine but currently not required, either in idle state or the computation requires less resources. *"Often, the resources consumed by that application are dwarfed by the size of the operating system in terms of memory, disk space and CPU utilization"* (InformationWeek, 2013), this suggests that the overheads may exceed the real computation cost for a task. The hypervisor does not have the ability to observe resources inside the virtual machine as it is a completely separate system from the rest. This is crucial as virtual machines get a slice of cpu and memory allocated to them and they are both unavailable to the rest of the system even when they are not utilized.

4.3 Containers

Containers aim to solve the issue of under utilization of virtual machines and overheads of running a full blown operating system in a virtual machine. Container is a thin layer of separation in the Linux operating system with shared kernel and libraries while keeping all containers running separate from each other. This reduces overhead significantly. According to LinuxJournal, 2013, the boot up time of a linux container is between five and fifteen seconds, considerably less than a virtualized solution. As InformationWeek points out Heroku, dotCloud and OpenShift are already using containers for cloud services. Furthermore, Docker has been attempting to popularize container solutions and make deployment of web applications a more manageable task.

4.4 Interfaces

Regardless of whether containers or virtualization is used, it is required for a cloud service to be successful to have a well defined interfaces. Ideally, the interfaces of a cloud service would be part of a standard to allow clients to switch from one cloud provider to another and/or utilize multiple providers at once. This applies particularly to data storage. *"Data itself is treated as a separate set of services so that the information can be shared across applications and across customers, suppliers, and partners"* (The Role of the OS in the Cloud, 2011). In order to improve portability of applications, each service should be designed to be independent.

This is particularly true of database management systems which will require access from various different applications. This is, of course, more of a software design principle rather than being OS specific, however, this also applies to the OS being used as the underlying system for the cloud. It should be portable with relatively easy deployment and should provide support for a wide range of I/O devices and hardware drivers.

4.5 Failover

Cloud services are generally advertised as always online. What this means in practice is when a server fails, the session needs to transferred onto a different server in order to provide seamless uptime. This can be relatively straight forward in the case of virtual machines. Virtual machines can be stored on a different server with a replication model and in the case of server failure the virtual machine is instantiated on a different server in the distributed network. This can be achieved automatically through the hypervisor and requires little human intervention.

Failover becomes more difficult to handle in the case of containers since user tasks are running as containers only and the state of the machine is bound to the particular server. Replication techniques can be used to ensure data is preserved but this scenario requires additional resources. Research in the area of failover of containers is limited as containers only becoming more popular.

5 Existing cloud frameworks

To explore what cloud frameworks are currently used in the industry, focus will be given to both virtual machine solutions and containers. One of the most popular virtual machine solutions is Amazon AWS. On the other side of the spectrum there is Docker with container focused approach. Of course, hybrid solutions are in place as well.

The large success of Amazon AWS can be attributed to mostly very well defined API and the infrastructure provided by Amazon. It provides the Elastic Compute Cloud service which is designed for scalability both up and down. The services is based on virtual machines, each instance is running its own operating system and can be used on demand. The ability to use virtual machines makes modularity very feasible as the underlying operating system is independent of the client operating system running. Additionally, data management services are provided as separate components of the system to encourage reusability and modularity. The Amazon distributed network is hosted in various locations all across the globe and allows for faster response time due to reduced location spatiality.

Containers are gaining increased attention as they provide reduced overheads for deployment. Docker Inc. has is one of many companies attempting to make the container solution more feasible. The main aim is to provide reusable and highly modular components running on the linux kernel. The kernel itself is shared with all the containers on a single machine while providing a thin wrapper around the client processing. It uses virtual memory like techniques to reduce dependency on the underlying hardware and additional system requirements.

Based on the evolution of cloud computation, it is reasonable to expect more and more hybrid solutions for the cloud with both hypervisors and simple container based linux systems. Containers still appear to a be a long away from being a simpler and more viable option for customers but with sufficient research and further development it has the potential to become more popular than virtual machines.

6 Advice for Future Cloud Providers

The cloud is booming and demand for cloud services is expected to keep increasing in the future. There are a number of possible solutions for the cloud infrastructure and the underlying operating systems used. At the moment, the ideal course of development appear to be a hybrid solution with support for both virtual machines and containers. Majority of the industry utilizes the Linux operating system as a basis for distributed cloud frameworks due to its maturity and large community. It provides a cost efficient, secure and highly customizable scenario.

The reason for hybrid solutions is the expected increase in demand for container cloud based solutions while virtual machines help maintain clientele and provide accessibility for other operating systems than Linux. Additionally, reasonable management software needs to be provided for balancing nodes in the distributed network. Hypervisor is one of the options for virtual machines, however, other solutions with a hypervisor like solution where the main server monitors the distributed network and handles actions and failovers can be used.

Additionally, sufficient security of the operating system will need to be ensured. Client instances should be running separately from each other and there should be a high level of modularity to ensure instances can be moved from a server to a server in the case of failure. Contingency plans should be made to ensure data consistency and impact mitigation is done.

Furthermore, future cloud service providers should ensure that each service provided is in fact independent of any other service and can be thus used in a wider range of applications and accessed from other points in the cloud network. It should work as a single unit rather than part of a bigger application.

Moreover, access to the cloud should be through well defined and industry standard methods. These methods should be easy to use and should provide a basis for custom applications to be built on top in order to allow clients to write automated tools for interaction with the system.

7 Conclusion

Requirements for the cloud computation platform have been presented and evaluated with respect to their complexity and solutions used in the industry. A hypervisor is used to manage virtual machines in the distributed network. It is responsible for load balancing of the physical machines and handles creation and termination of virtual machines in the network. The topic of virtual machines and the modularity they provide has been explored and linked to practice in the industry. The alternative approach of containers has been explored and its benefits and drawbacks presented. Based on the research it appears that containers may be the future of the industry, however, current level of development does not make them so widely used in practice. They do, however, seem to have the potential to increase utilization and provide a more fine grained level of monitoring for the server unit responsible for allocating resources to tasks.

Main requirements for future cloud service providers were presented. A hybrid solution appears to be the most viable option at the moment as it provides compatibility with current service providers yet has the potential to attract demand likely to be created in the future. The concept of containers is still in development and it is difficult to predict how successful it will end up being. In the time being, virtual machines are to be the solution seeked by many clients.

References:

- **George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair**
  [Book] *Distributed Systems: Concepts and Design (5th Edition), 2011*,
  ISBN-13: 978-0132143011
- **NIST - The NIST Definition of Cloud Computing**
  [PDF] *http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf*
  [Visited November 21, 2013]
- **Borko Furht, Armando Escalante**
  [Book] *Handbook of Cloud Computing*
  ISBN 978-1-4419-6523-3
- **HowStuffWorks.com - How Server Virtualization Works**
  [Web] *http://computer.howstuffworks.com/server-virtualization2.htm*
  [Visited November 22, 2013]
- **InformationWeek.com - Containers Add New Efficiency To Cloud Computing**
  [Web] *http://www.informationweek.com/cloud*
  /containers-add-new-efficiency-to-cloud-computing/d/d-id/1112037?
  [Visited November 22, 2013]
- **The Role of the OS in the Cloud**
  [Web] *www.redhat.com/f/pdf/The_Role_of_the_OS_in_the_Cloud.pdf*
  [Visited November 22, 2013[