Paper 1 Report

1 Cloud Computing

The paper gives insights into the evolution of cloud computing services. It mentions that in future, people will move to an increased usage of centralized facilities operated by third party compute and storage facilities. Looking into the current scenario, this prediction is indeed correct as more and more cloud based solutions have come up since this paper was written. There are cloud services like Google Stadia (gaming), Databricks (Big-Data Analytics), AWS (Web Services) etc.

2 Grid Computing

The term grid was coined to describe computing power on demand (non-service applications) to users approach. An example of this can be the Lisa clusters of supercomputer.

3 Cloud vs. Grid

Business Model Cloud-based models provide consumption based service cost to users. Scalability of services makes it easier to bring in more users, drive prices down and increase profitability. This trend is still in view as of today, as high performance computations are becoming cheaper day by day and scalability increasing, further making it easier to add more performance intrinsic services like Deep Neural Networks and machine learning algorithms. For grids, the business is concerned more on merging and sharing of resources by collaboration. As more organisations join with the grid for usage, they also provide power, computation resources to the grid making the grid much more diverse and large. Most of the cluster computing resources in the education field are done this way by collaboration between universities' resources. An example can be the DAS-4 clusters in the Netherlands.

Architecture Grids are more focused on integration of different and existing computing resources with their hardware, operating systems, local resource management and security infrastructure to form grids for clustered computations. These grids can interact with each other for message transfers, and resource allocations. The issue with this approach is the security risks, interoperability and administrative constraints with usage of two different usage policy resources. Clouds are built on top of many protocols (REST APIs, etc) to provide services over the application to the compute and storage resources given, can be a grid too. The scaling factor is not needed to known by the user, the cloud software takes care of the scalabilties.

Resource Management As grids involves batch-scheduling, the jobs have to be stored in a queue till the required resources are released. This prevents grids from supporting interactive applications unlike clouds. Although low latency multi-scheduling jobs are now possible like executing in GPUs to increase throughput. Cloud computing focuses on sharing resources rather than locking it to a user and hence is much more QoS for end users. In the data model level, cloud computing is deemed to be shifting to a more cloud-client computing system. This prediction is true to this day, as client computing facilities are becoming more and more faster with multi-core features. Technology is limited now with number of transistor in chips whereas, new architecture solutions are paving way to provide integration of more cores to a desktop or client computer. Grids provide an abstraction layer for data with virtual data. In terms of data locality issues, there are new technologies which helps to lock data spatiality for faster access. One such example is of Apache Spark that can be run in-memory data pipelines for faster map-reduce and standard analytical functions. Clouds uses more virtualization compared to that of grid which uses almost no virtualization to provide service applications on top of the distributed architecture of the cloud system.An example of virtualization can be docker containers to run micro-services over the cloud.

Programming Model Reliability, Fault tolerance are important factors to be considered in Grid programming model. These are nowadays taken care by multiple replication factors for file within nodes in the clusters. A common model is MPI for message passing between parallel computing clusters. Linda is another such interface for retrieval and setting up of tuple spaces. These systems are pretty much implemented with a master-worker configuration. Apache Kafka is one such programming model for implementation of high throughput data pipeline between the nodes. Map-Reduce is another landmark programming model which kick-started big data and real time computations in cluster computing. Apache Spark makes use of this technique with in-memory access to make efficient and faster data fetch and computations.

Application Model Grid system supports high performance as well as high throughput computing. The MTMD architecture model of grids support this. These applications can be coupled into tasks with different resources, coalescing to one larger application. Clouds perform poorly in the field of HPC, as it requires efficient and low latency networks and resources. Since clouds have shared resource access, performance is limited here. But nowadays, even dedicated hpc services are utilised in cloud, like the AWS neural network instances. Although, acquiring a free instance is time consuming. Gateways provide community-developed applications for integration. This gateways have today linked Web2.0 with the cloud systems to provide web based services.

Security Model Grids, since they have heterogeneity and interoperable functionality is designed to be more secure than a cloud based system. Grids have hierarchical authorisation over levels of clusters depending on location, user role, etc. The recent AWS and iCloud hacks defend this position strongly, as grids are not well connected with Web2.0 as like cloud systems.