Cloud Services & Multi site Workflows

Sonali J.Rathod
M.E. 1st year student, JCET Arni Road, Yayatmal, Maharashtra

***Abstract*: Cloud computing is the long dreamed vision of computing as a utility, where data owners can remotely store their data in the cloud to enjoy on-demand high-quality applications and services from a shared pool of configurable computing resources. The global deployment of cloud datacenters is enabling large scale scientific workflows to improve performance and deliver fast responses**.

I. INTRODUCTION

Cloud Computing is the delivery of service, which is an Internet-based development and use of computer technology. The ever chapter and more powerful process, together with the ”Software as service”(saas) computing architecture, are transforming data centers into pools of computing services on a huge scale

Meanwhile, the increasing network bandwidth and reliable yet flexible network connections make it even possible that clients can now subscribe high-quality services from data and software that reside solely on remote data centers. Although envisioned as a promising service platform for the internet, this new data storage paradigm in “cloud” bring about many challenging design issues which have profound influence on the security and performance of the overall systems.

II. LITERATURE REVIEW

2.1 Cloud Computing

Cloud Computing is a technology that uses the internet and central remote servers to maintain data and applications.

There are many type of public cloud computing:
1. Infrastructure as a service (IaaS)
2. Platform as a service (PaaS)
3. Software as a service (SaaS)

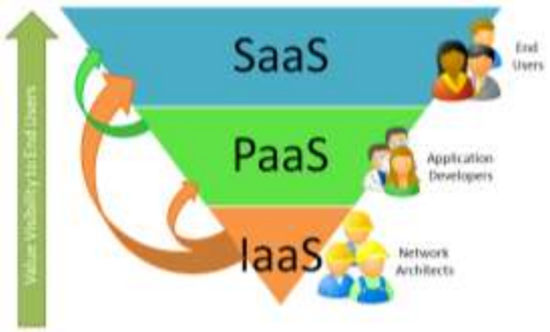

Fig. 2.1.1: Cloud Services

- **Platform as a service (PaaS):**

In the PaaS model, cloud providers deliver a computing platform typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. With some PaaS offers, the underlying computer and storage resources scale automatically to match application demand such that cloud user does not have to allocate resources manually.

Fig. No. 2.1.2: Cloud Platform as a service (PaaS)

Examples of PaaS include: Amazon Elastic Beanstalk, Cloud Foundry, Heroku, Force.com, EngineYard, Mendix, Google App Engine, Microsoft Azure.

In 1999, Salesforce.com was established by Marc Benioff, Parker Harris, and his fellows. They applied many technologies of consumer web sites like Google and Yahoo! to business applications. Amazon.com played a key role in the development of cloud computing by modernizing their data centres after the dot-com bubble and, having found that the new cloud architecture resulted in significant internal efficiency improvements, providing access to their systems by way of Amazon Web Services in 2002 on a utility computing basis. 2007 saw increased activity, with Google, IB Mand a number of universities embarking on a large scale cloud computing research project, around the time the term started gaining popularity in the mainstream press.

2.3 Cloud clients

Users access cloud computing using networked client devices, such as desktop computers, laptops, tablets and smart phones. Some of these devices - cloud clients - rely on cloud computing for all or a majority of their applications so as to be essentially useless without it. In computing, client is a system that accesses (remote) service on another computer by some kind of network. The term was first applied to devices that was not capable of running there ownstand alone programs, but could interact with remote computers via a network.

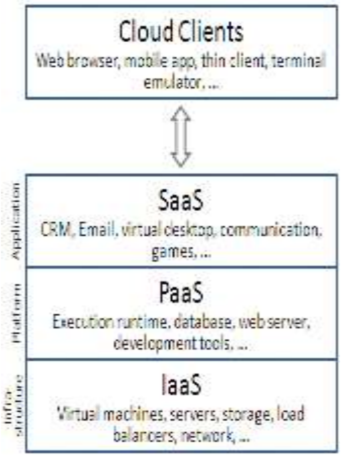
Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
SaaS
CRM, Email, virtual desktop, communication, games, ...
PaaS
Execution runtime, database, web server, development tools, ...
IaaS
Virtual machines, servers, storage, load balancers, network, ...
Application
Platform
Infra-structure

Fig. No.2.3.1: Cloud Clients

2.4 Benefits of Cloud Computing

Data as a Service brings the notion that data quality can happen in a centralized place, cleansing and enriching data and offering it to different systems, applications or users, irrespective of where they were in the organization or on the network. As such, Data as Service solutions provide the following advantages:

- **Agility –** Customers can move quickly due to the simplicity of the data access and the fact that they don't need extensive knowledge of the underlying data. If customers require a slightly different data structure or has location specific requirements, the implementation is easy because the changes are minimal.
- **Cost-effectiveness –** Providers can build the base with the data experts and outsource the presentation layer, which makes for very cost effective user interfaces and makes change requests at the presentation layer much more feasible.

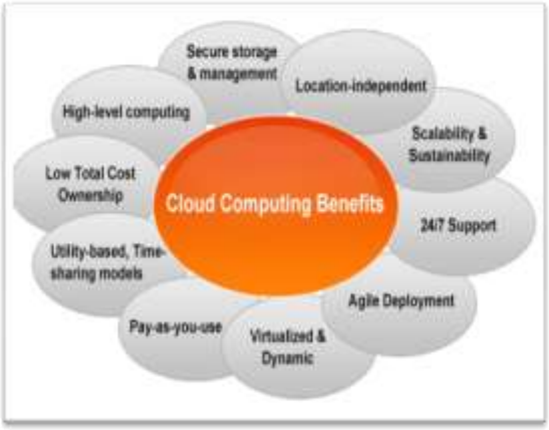

Fig. No.2.4.1: Cloud Computing Benefits

Data quality – Access to the data is controlled through the data services, which tends to improve data quality because there is a single point for updates. Once those services are tested thoroughly, they only need to be regression tested if they remain unchanged for the next deployment.

2.5. Features of Publically Auditable Service:

1. Multitenant:

Multitenant refers to a principle in software architecture where a single instance of the software runs on a server, serving multiple client organizations. Multitenancy is contrasted with a multi-instance architecture where separate software instance are set up for different client organizations. With a multitenant architecture, a software application is designed a virtually partition its data and configuration, and each client organizations works with a customized virtual application instance. Multitenancy is also regarded as one of the essential attributes of cloud computing.

2.Self-Service and On-demand Services:

Cloud computing is based on self-service and on-demand service models. It should allow the user to interact with the cloud to perform tasks like building ,deploying, managing and scheduling .The user should be able to access computing capabilities an and when they are needed and without any interaction from the cloud-service provider. This would help users to be in control, bringing agility in their work, and to make better decision on the current and future needs.

3.Pay Per Use:

Consumer are charge fees based on their usage of a combination of computing power , bandwidth use and/or storage .There are different services are hosted on application Then the clients are used which services they pay the charges for it.

4.High return on investment :

Cloud computing does not have any upfront cost. It is completely based on usage. The user is build based on amount of resources they use. This helps the user to track their usage and ultimately help to reduce cost. Cloud computing must provide means to capture, monitor, and control usage information for accurate billing. The information gathered should be transparent and readily available to the customer. This is necessary to make the customer realize the cost benefits that clouds computing bring.

5. Centralized Storage:

The use of central disk storage also makes more efficient use f disk storage. This can cut storage costs, freeing up capital to invest in more reliable, modern storage technologies, such as RAID arrays which support redundant operation and storage are networks which allow hot-adding of storage without any interruption. Further, it means that losses of disk drives to mechanical or electrical failure which are statically highly probable events over a timeframe of years, with a large number of disks involved are often both less likely to happen and less likely to cause interruption..

III.ADAPTIVEFILE MANAGEMENT ACROSS CLOUD

In this section we show how the main design ideas behind the architecture of OverFlow are leveraged to support fast data movements both within a single site and across multiple datacenters.

3.1 Core Design Principles

Our proposal relies on the following ideas:

_ Exploiting the network parallelism. Building on the observations that a workflow typically runs on multiple VMs and that communication between datacenters follows different physical routes, we rely on multi-route transfer strategies. Such schemes exploit the intra-site low-latency bandwidth to copy data to intermediate nodes within the source deployment (site). Next, this data is forwarded towards the destination across multiple routes, aggregating additional bandwidth between sites.

_ Modeling the cloud performance. The complexity of the datacenters architecture, topology and network infrastructure make simplistic approaches for dealing with transfer performance (e.g., exploiting system parallelism) less appealing. In a virtualized environment such techniques are at odds with the goal of reducing costs through efficient resource utilization. Accurate performance models are then needed, leveraging the online observations of the cloud behavior. Our goal is to monitor the virtualized environment and to predict performance metrics (e.g., transfer time, costs). As such, we argue for a model that provides enough accuracy for automating the distributed data management tasks.

_ Exploiting the data locality. The cumulative storage capacity of the VMs leased in one's deployment easily reaches the TBs order. Although tasks store their input and output files on the local disks, most of the storage remains unused. Meanwhile, workflows typically use remote cloud storage (e.g.,Amazon S3, Azure Blobs) for sharing data [8]. This is costly and highly inefficient, due to high latencies, especially for temporary files that don't require persistent storage. Instead, we propose aggregating parts of the virtual disks in a shared common pool, managed in a distributed fashion, in order to optimize data sharing.

_Cost effectiveness. As expected, the cost closely follows performance.

Different transfer plans of the same data may result in significantly different costs. In this paper we ask the question: given the clouds interconnect offerings, how can an application use them in a way that strikes the right balance between cost and performance?

_ No modification of the cloud middleware. Data processing in public clouds is done at user level, which restricts the application permissions to the virtualized space. Our solution is suitable for both public and private clouds, as no additional privileges are required.

3.2 Architecture

The conceptual scheme of the layered architecture of OverFlow is presented in Fig. 1. The system is built to support at any level a seamless integration of new, user defined modules, transfer methods and services. To achieve this extensibility, we opted for the Management Extensibility Framework,1 which allows the creation of lightweight extensible applications, by discovering and loading at runtime new specialized services with no prior configuration. We designed the layered architecture of OverFlow starting from the observation that Big Data application requires more functionality than the existing put/get primitives. Therefore, each layer is designed to offer a simple API, on top of which the layer above builds new functionality. The bottom layer provides the default "cloudified" API for communication. The middle (management) layer builds on it apattern aware, high performance transfer service the top (server) layer exposes a set of functionalities as services The services leverage information such as data placement, performance estimation for specific operations or cost of data management, which are made available by the middle layer. This information is delivered to users/applications, in order to plan and to optimize costs and performance while gaining awareness on the cloud environment. The interaction of OverFlow system with the workflow management systems is done based on its public API. For example, we have integrated our solution with the Microsoft Generic Worker [12] by replacing it's default Azure Blobs data management backed with OverFlow. We did this by simply mapping the I/O calls of the workflow to our API, with OverFlow leveraging the data access pattern awareness as fuhrer detailed in Sections. The next step is to leverage OverFlow for multiple (and ideally generic) workflow engines across multiple sites. We are currently working jointly with Microsoft in the context of the Z-CloudFlow[14] project, on using OverFlow and its numerous dataFig. 1. The extendible, server-based architecture of the OverFlow System.

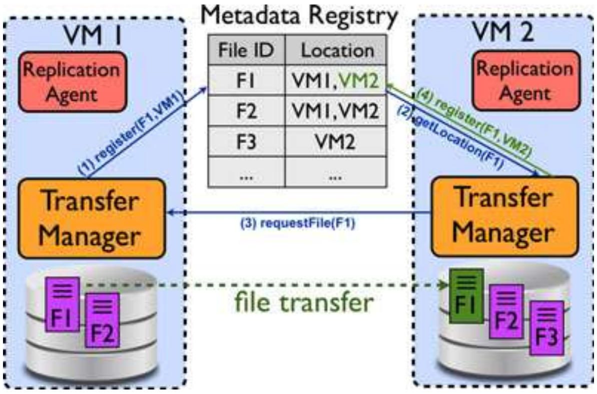

Fig. No.4.2.1. Architecture of the adaptive protocol-switching file management system

3.3 Inter-Site Data Transfers via Multi-Routes

In a second step, we move to the more complicated case of inter-site data transfers. Sending large amounts of data between two datacenters can rapidly saturate the small interconnecting bandwidth. Moreover, due to the high latency between sites, switching the transfer protocol as for intra-site communication is not enough. To make things worse, our empirical observations showed that the direct connections between the datacenters are not always the fastest ones. This is due to the different ISP grids that connect the datacenters (the interconnecting network is not the property of the cloud provider). Considering that many Big

Data applications are executed on multiple nodes across several sites, an interesting option is to use these nodes and sites as intermediate hops between source and destination. The proposed multi-route approach that enables such transfers is shown in Fig. 3. Data to be transferred is first replicated within the same site (green links in Fig. 3) and it is then forwarded to the destination using nodes from other intermediate datacenters (red arrows in the Figure). Fig. 2. Architecture of the adaptive protocol-switching files management system. Operations for transferring files. Instances of the inter-site transfer module are deployed on the nodes across the datacenters and used to route data packages from one node to another, forwarding data towards the destination. In this way, the transfers leverage multiple paths, aggregating extra bandwidth between sites. This approach builds on the fact that virtual routes of different nodes are mapped to different physical paths, which cross distinct links and switches (e.g., datacenters are interconnected with the ISP grids by multiple layer 2 switches Therefore, by replicating data within the site, which is up to 10 times faster than the inter-site transfers, we are able to increase the performance when moving data across sites. OverFlow also supports other optimizations: data fragmentation and re-composition using chunks of variable sizes, hashing, and acknowledgement for avoiding data losses orpackets duplication. One might consider the acknowledgement-based mechanism redundant at application level, as similar functionality is provided by the underlying TCP protocol. We argue that this can be used to efficiently handleand recover from possible cloud nodes failures. Our key idea for selecting the paths is to consider the cloud as a two-layer graph. At the top layer, a vertex represents a datacenter. Considering the small number of data centers in a public cloud (i.e., less than 20), any computation on this graph (e.g., determine the shortest path or second shortest path) is done very fast. On the second layer, each vertex corresponds to a VM in a datacenter. The number of such nodes depends on application deployments and can be scaled dynamically, in line with the elasticity principle of the clouds. These nodes are used for the fast local replication with the purpose of transferring data in parallel streams between the sites. OverFlow selects the best path, direct or across several sites, and maximizes its throughput by adding nodes to it. Nodes are added on the bottom layer of the graph to increase the inter-site throughput, giving the edges based on which the paths are selected on the top layer of the graph. When the nodes allocated on a path fail to bring performance gains, OverFlow switches to new paths, converging towards the optimal topology.

CONCLUSION

As per above we conclude that the cloud computing system is very beneficial for finance system. This application also provides the branch of financial company to store their client related data on the cloud serve. This service will provide new inventive ways to use computers and provide services. Client data sources are managed under the cloud resources.

This paper introduces OverFlow, a data management system for scientific workflows running in large, geographically distributed and highly dynamic environments. Our system is able to effectively use the high-speed networks connecting the cloud datacenters through optimized protocol tuning and bottleneck avoidance, while remaining non-intrusive and easy to deploy. Currently, OverFlow is used in production on the Azure Cloud, as a data management backend for the Microsoft Generic Worker workflow engine.

REFERENCES

[I]http://www.reference.com/browse/what+is+module?www.sei.cmu.edu/.../CloudComputingArchtecture
[II]http://www.answers.com/topic/cloud-computing
[III]http://en.wikipedia.org/wiki/Financial_system [IV]"Software as a Service (SaaS)". Cloud Taxonomy. http://cloudtaxonomy.opencrowd.com/taxonomy/software-as-a-service/.Retrieved 24 April 2011. [V]Azure Succesful Stories [Online]. Available: http://www. windowsazure.com/en-us/case-studies/archive/, 2015.

AUTHOR(S) PROFILE

Sonali Janardhan Rathod received the B.E degree in Information Technology and persuing M.E degrees in computer science and engineering from Jagadamba college of engineering & technology Yavatmal , india. respectively.