An Generic Cloud Framework for Cloud Based Applications

Arvind Kumar M, Mr T.Auntin Jose
Computer Science & Engineering, RRCE, Bengaluru, India
kmarvind@gmail.com, Auntin123@yahoo.com

***Abstract*—**This paper focuses on solving the problem of cloud application developers facing while using the available Clouds in their cloud application development process. There are multiple cloud infrastructures available in the market. The Cloud Application developers can't be familiar with all the clouds available in the market. Our paper addresses this issue, and talks to facilitate the cloud application developer to provide an generic abstract layer which hides the back end cloud infrastructure and with a uniform API layer to make use in their cloud applications. Generic Cloud Frame work will provide the facility to common developers to have a minimum knowledge about the cloud and solve the various problems. With the help of Generic Cloud Framework, the developers can provision and develop various applications like provisioning for DB management systems, Medical Systems, Educational systems, Agriculture systems, using an ―Infrastructure as a Service‖ (IaaS) cloud. To solve this problem in a generic way by providing the domain specific API Libraries with uniform interface to back end cloud infrastructure.

Keywords *— Cloud, IaaS; PaaS; Library; Analytics;*

I. Introduction

Cloud computing is not an technology now a days. But it is becoming and a valuable and important which is becoming an fundamentally changing the way we use and develop on-demand applications. As you we know, Linux & open source provides the foundation for the cloud (either for public or private infrastructures). We explored the cloud anatomy, its architecture, and the open source technologies used to build these platforms. Cloud computing is commonly considered as ―the greatest sliceable thing in IT infrastructure ―, as it allows the computing power can be utilized to behave similar to a generic utility that is always available to user on their need basis from anywhere. This utility-orientation about the hardware and software usage makes the cloud computing as an potential transformative technology. The model of pay-as-you-go service makes it an essential economic savvy by reducing the capital expenses and reducing the cost of ownership over time.

Most of the applications are good candidates for adapting to the cloud — if the applications are difficult to configure and manage, then they are the definite candidates for hosting them in the cloud. Many applications have been already ported on virtualized hardware provided by IaaS, SaaS or PaaS providers. The key factor need to be keep in mind is, cost-effectiveness in deployment of cloud applications remain open. One such key factor involves application optimization, which basically assumes the availability of Hardware, Software resources, configuration of them which will result in performance optimization. This ignores the OPEX (operational cost) of running the applications. The pay-as-you-go model used in cloud computing, however, attracts the large number of users to consider hosting the applications on the Cloud, to reduce operational costs and consider increasing the performance.

In this paper, we discuss how any kind of users can easily implement their cloud applications by minimizing the detail knowledge of the backend cloud. This reduces the operational & development cost of a Cloud Applications. That is to say, the Libraries provided by the Generic Cloud Framework will be tapping all the requests from the Cloud applications and translating to back end cloud specific calls to use the specific resource types from physical available cloud at their disposal. We propose a resource provisioning framework with common generic API libraries which identifies the heterogeneous cloud environment, resources, virtual machines etc. that can collectively abstracts the user from Cloud specific knowledge. The requirement for the developer is to have detailed knowledge about the APIs provided by the library. At run time these calls will be translated to make best use of resources by intelligently routing the incoming queries to specific cloud environment.

The work is still in progress. In this work in progress paper, we describe one domain and few set of API libraries to use limited cloud resource for provisioning and utilizing them. We can be termed it as a black box provisioning as the user no need to have any insight or the resource usage of the back-end cloud. An sample application on desktop and handheld device is developed using the Generic Cloud Frame work APIs to demonstrate the end to end flow of the data from domain user to cloud.

We formulated these solutions to help the cloud application developer for packaging and tackling their applications with greater ease and efficiency.

II. SYSTEM STRUCTURE

Cloud Infrastructure Services. Our model system assumes an cloud infrastructure which is similar to the offerings made from various IaaS providers, Cloud vendors provide access to the pre-configured computing power they could be virtual machines or cloud servers on which users can remotely install and run their software. Available server types are categorized by the resources they provide to the end user, such as CPU, I/O Band Width, and Physical disk space and system memory size. Cloud resources are usually offered through a –pay-as-go‖ price model, they can be rentable for a certain predefined period, probably for an hour, and this period cost is fixed. The cost depends only on the server configuration and not on the server utilization over the time. IaaS clouds are also will provide the services for data storage and transfer. Usually the additional request charges will cost the user, For example, Amazon‘s EC2, S3, & EBS services offers persistent storage space and charged based on the per request basis for each I/O operation. This work considers the charges only for the Servers and I/O operations. Regardless of pricing model used, it is the responsibility of the IaaS user to provision appropriately to use its resources. Based on our estimation the workload expected to be by hosting the application.

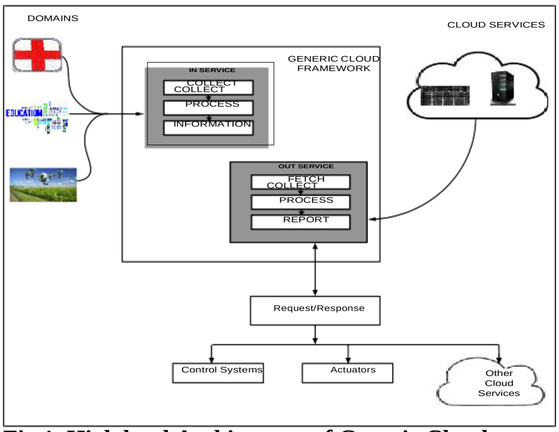

Fig.1. High level Architecture of Generic Cloud Framework

Domain Specific Components. Cloud-applications for specific domain need specific services. Based on the Hardware, Software required for that domain need to have customized APIs to configure Hardware, APIs for domain specific operations, APIs for Domain specific analytics etc. We pre-assume that the application writer is a domain expert rather a cloud expert. Each domain will have a specific library and the application developer need to understand the library instead of the cloud infrastructure. The APIs reside on API server with wrapper routines to the back-end cloud infrastructure. The (Figure 1) shows over a high level architecture of the system. The application development for specific domain needs appropriate library.

In-Service Module. This module will have the interface to cater the Domain specific community and Cloud service. It will receive the Client request and translate it into the back-end cloud specific call. The calls received from the client are un-wrapped; the parameters are analysed and forwarded to specific cloud utilities and infrastructure modules provided by the cloud infrastructure provider calls. For meaningful, the in-service module receives the request, process the request and at the end, it will send it to the cloud for servicing the request. Based on the service requested the expected distribution of requests across the different service handlers also happens based on this module. This module provides facilities for methodical representation and facilitates to generate reports for various purposes.

Out-Service Module. Provisioning the data collected in the cloud from In-Service module to end users on need basis for reference, for predictions, etc. The Out-Service module is responsible for representing the data to the users. Again to fetch the data from cloud, the user will have domain Specific APIs . The client uses these set of APIs and read the data for representation, dash boarding, Analytics, decision making purposes. The out-service module provides the APIs to fetch the data either on the desktops or on any hand held devices.

The APIs interactions could be represented as shown below picture.

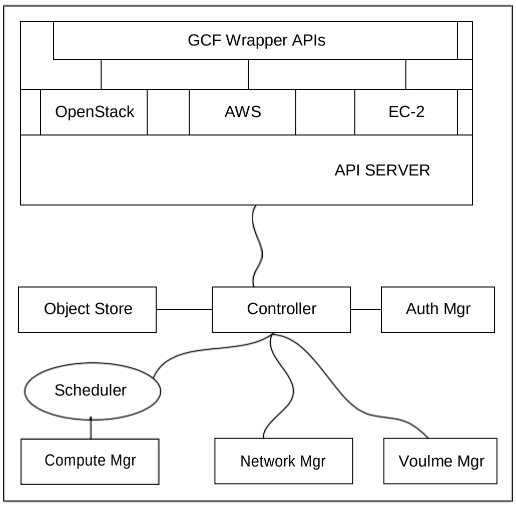

Fig. 2. The Interfaces interaction

III. EXPERIMENT CONFIGURATIONS

Throughout this paper we reference the results using the proof-of-concept for our framework deployed for an application in an agriculture environment, which involves Hardware modules, Software modules, Google services, Amazon Web Services [1]. We deployed the Hardware to read the vegetative index of a particular geographical area‘s crop and upload the data using In-Services of Generic cloud framework. In-turn this data could help to analyse the health of a particular crop in a specific geo-locations. It can be used

to advice the agriculture community for predictive analysis, like what could be the proactive health care of the crops by using proper ratio of Urea, Nitrogen fixes and appropriate pesticides at right time and right composition. The hardware module used for reading the crop health, and is uploaded to the cloud along with the crop‘s vegetative index, and geo co-ordinates. In this Proof Of Concept the cloud environment used is Amazon Web Services. The HW used are, PIC Microcontroller, Vegetative index reading sensor, GSM and GPRS 908 module 908. All these components are interconnected as below,

Geolocation based Vegetative Index					
Longitude	***Latitude***	***Humidity***	***Temperature***	***Vegetative Index***	***City / Feedback***
12.91N	77.92 E	28	34	5F	B‘lore
13.08N	80.27 E	70	30	3B	Chennai
17.38N	78.41 E	23	36	2C	Hyderabad
19.07N	72.87 E	62	31	1A	Mumbai
22.51N	88.36 E	66	31	32	Kolkota

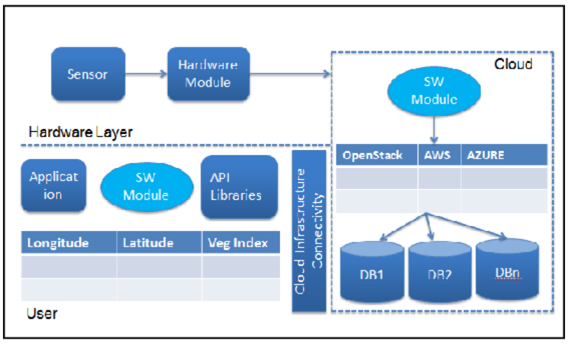

Fig. 3. Generic Cloud Frame work, POC configuration

This POC generates the reports in the form of crop‘s geo location, it is health indicator and environment parameters are also can be captured, if the necessary sensors are used to collect the data from the field. We also can extend this and optimize to produce estimates for any specific crops, analytics modules will give the usage of each micro components and ratio analysis for demographic related parameters can be defined. We correlated this with actual resources consumed without using these modules in real field. The consistency and high availability concepts can also be supported and achieved.

IV. GENERIC CLOUD FRAMEWORK REPORTS

The report generated by this proof of concept is as mentioned

Table .1. Sample Report structure

In the above given table. The sensor information is captured and could be used for the analytics purpose. This data set information will be will be uploaded into the cloud database. The user can fetch, process and analyse with the help of APIs provided by the GCF.

V. CONCLUSION

The outcome of the generic cloud framework is in multi-fold. It directly helps the user who would like to use the cloud environment for their specific application development activities. Thus Generic Cloud Framework gives users the speed; simplicity and control they need to develop and deploy applications easily and faster. It provides different specific API layers and infrastructure customized to their development environments.

Over all benefits of this framework are listed as below,

- Enable users to have real time data.
- Application development will be very quick since developer need not be aware about cloud services.
- Reuse of the frame work for multiple purposes.
- Data analytics can be done by understanding one single library on various domain data.
- Easy to maintain the application code.

Enable users to have real time data. User can gain flexibility & control over their choice of languages, frameworks, application services and clouds. Portability is ensured, and vendor lock-in can be eliminated. As the global industry standard for PaaS is open source technology, we operate under the open governance by contribution model. The customer who generates the data can be processed and stored in a defined common standard format, so that it can be used at later point of time by the future customer for their analysis purpose.

Quick Application development. By connecting human brainpower and computing power, we reduce development times and accelerate how applications are designed, developed and delivered. Empowering the companies to cut weeks and months durations once needed to develop and ship new applications to just days, hours, even minutes, Generic Cloud Framework makes faster easy.

Reuse of the frame work for multiple purposes. Built on hardened production infrastructure for global enterprises, the GCF platform ensures scalable micro services and continuous deployment, faster cycle time and higher reliability. Everything is fully scalable, including the platform itself. The framework developed can be reused for various purposes.

Data analytics can be done on various domain data. Built on hardened production infrastructure for global enterprises, the platform ensures scalable micro services and continuous deployment, faster cycle time and higher reliability. Everything is fully scalable, including the platform itself. The data generated can be tiered and utilized for future analytics purpose. Only data also can be rendered to the customers who are looking for the for R&D purposes in multiple domains.

Easy to maintain the application code: Built on hardened production infrastructure for global enterprises, the platform ensures scalable micro services and continuous deployment, faster cycle time and higher reliability. Everything is fully scalable, including the platform itself.

Acknowledgment

This work has been partially incorporated by the open stack group as a part open source and getting customised for commercial use under various brand names like Open Shift.

References

[9] Amazon Web Services, http://aws.amazon.com/.

[10] D. J. Abadi. Data management in the cloud: Limitations and opportunities. IEEE Bulletin on Data Engineering, 32(1), 2009.

[11] A. Aboulnaga, K. Salem, A. A. Soror, and U. F. Minhas. Deploying database appliance in the cloud. IEEE Bulletin on Data Engineering, 32(1), 2009.

[12] M. Ahmad, A. Aboulnaga, S. Babu, and K. Munagala. Modeling and exploiting query interactions in database systems. In CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management, pages 183–192, New York, NY, USA, 2008. ACM.

[13] M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska. Building a database on s3. In SIGMOD, 2008.

[14] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good. The cost of doing science on the cloud: the montage example. In SC '08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing, pages 1–12, Piscataway, NJ, USA, 2008. IEEE Press.

[15] R. P. Doyle, J. S. Chase, O. M. Asad, W. Jin, and A. M. Vahdat. Modelbased resource provisioning in a web service utility. In USITS, 2003.

[16] A Highly Scalable Transactional Multi-Tier Platform as a Service from Director Distributed Systems Lab (LSD), Univ. Politecnica de Madrid (UPM) Grant Agreement number: 257993 – 2013

[17] Efficient & scalable Para virtual I/O system – 2013 USENIX Annual Technical conference

[18] Sukhpal Singh and Inderveer Chana, ―QoS-aware Autonomic Resource Management in Cloud Computing: A Systematic Review‖,―ACM Computing Surveys‖ , Volume 48, Issue 6, pp. 1-39, 2015

[19] Rajkumar Buyya, Rodrigo N. Calheiros, and Xiaorong Li. ―Autonomic cloud computing: Open challenges and architectural elements‖, *In* Proceeding of the Third International Conference on Emerging Applications of Information Technology (EAIT). (2012), 3-10, IEEE.

[20] Kaloxylos, Alexandros, Robert Eigenmann, Frederick Teye, Zoi Politopoulou, Sjaak Wolfert, Claudia Shrank, Markus Dillinger, ―Farm management systems and the Future Internet era‖, *Computers and Electronics in Agriculture*, 89 (2012): 130-144.

[21] Ranya Elsheikh, Abdul Rashid B. Mohamed Shariff, Fazel Amiri, Noordin B. Ahmad, Siva Kumar Balasundram, and Mohd Amin Mohd Soom, ―Agriculture Land Suitability Evaluator (ALSE): A decision and planning support tool for tropical and subtropical crops‖, *Computers and Electronics in Agriculture*, 93 (2013): 98-110.

[22] Raimo Nikkilä, Ilkka Seilonen, and Kari Koskinen, ―Software architecture for farm management information systems in precision agriculture‖, *Computers and Electronics in Agriculture*, 70(2) (2010): 328-336.

[23] C. G.Sorensen, S. Fountas, E. Nash, Liisa Pesonen, Dionysis Bochtis, Søren Marcus Pedersen, B. Basso, and S. B. Blackmore, ―Conceptual model of a future farm management information system‖, *Computers and Electronics in Agriculture*, 72(1) (2010): 37-47.

[24] Zhao Ruixue, ―Study on Web-based Agricultural Information System Development Method‖, *In Proceedings of the Third Asian Conference for Information Technology in Agriculture, China*, 601-605. 2002.

[25] C. G. Sørensen, Liisa Pesonen, D. D. Bochtis, S. G. Vougioukas, and Pasi Suomi, ―*Functional requirements for a future farm management information system*‖, Computers and Electronics in Agriculture, 76(2), (2011): 266-276.

[26] Yuegao Hu, Zhi Quan, and Yiyu Yao, ―Web-based Agricultural Support Systems‖, *In Proceeding of the Workshop on Web-based Support Systems*, 75-80. 2004.

[27] The Open Compute project—energy efficiency. (Available from: http://opencompute.org/about/energy-efficiency/) [Accessed on 21 November 2012].

[28] Hermenier F, Lorca X, Menaud J, Muller G, Lawall J. Entropy: a consolidation manager for clusters. Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, Washington, DC, USA, 2009; 41–50.

[29] Feller E, Rilling L, Morin C. Snooze: a scalable and autonomic virtual machine management framework for private clouds. Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), Ottawa, Canada, 2012; 482–489.

[30] Verma A, Dasgupta G, Nayak TK, De P, Kothari R. Server workload analysis for power minimization using consolidation. Proceedings of the 2009 USENIX Annual Technical Conference, San Diego, CA, USA, 2009; 28–41.

[31] Speitkamp B, Bichler M. A mathematical programming approach for server consolidation problems in virtualized data centers. IEEE Transactions on Services Computing (TSC) 2010; 3(4):266–278.

[32] Cardosa M, Korupolu M, Singh A. Shares and utilities based power consolidation in virtualized server environments.

Proceedings of the 11th IFIP/IEEE Integrated Network Management (IM), Long Island, NY, USA, 2009; 327–334.

[33] Kumar S, Talwar V, Kumar V, Ranganathan P, Schwan K. vManage: loosely coupled platform and virtualization management in data centers. Proceedings of the 6th International Conference on Autonomic Computing (ICAC), Barcelona, Spain, 2009; 127–136.

[34] Cloud Computing (2010), Wikepedia;en.wikipedia.org/wiki/

[35] Dr. Rich Wolski, (2010) Enterprise Cloud Control.

[36] Ezhil Arasan Babaraj, (2009), Driving Technology Direction on Cloud Computing Platform, Blog post; Hybridfox: Cross of Elasticfox and Imagination, ezhil.syscon.com/.

[37] GoGrid.com. http://gogrid.com/.

[38] IBM. ILOG CPLEX, http://www.ilog.com/products/cplex/