International Journal For Technological Research In Engineering
Volume 3, Issue 12, August-2016 *ISSN (Online): 2347 - 4718*

A NOVEL FRAMEWORK FOR MULTI DATA STORE IN CLOUD STORAGE

Balakrishna P[1], Sreenivasulu S[2]
[1]M.Tech Student, [2]Professor
Department of CSE, Prakasam Engineering College, Prakasam (D.t), Andhrapradesh

ABSTRACT: In period of time generation of enormous quantity data and also rise of cloud computing has introduced new aspects for data management. Many applications got to move with many heterogeneous data stores betting on the sort of data they need to manage: ancient data types, documents, and simple key value data, graph data, etc. Interacting with heterogeneous data models via different APIs, multiple data store applications imposes difficult task s to their developers. Indeed, programmers got to be conversant in completely different APIs. Additionally, developers got to master and handle the advanced processes of cloud discovery, and deployment of application and execution. Projected system represents a declarative approach using ODBAPI sanctioning to lighten the burden of the complex and non-standard task s of discovering relevant cloud surroundings and deploying applications on them whereas letting developers to easily concentrate on specifying their storage and computing needs.
Keyword: ODBAPI, data stores, rational data stores, NoSQL.

I. INTRODUCTION

Cloud computing, a comparatively recent term, has become today a hokum within the net applications world. Despite the importance of this paradigm, there's no agreement on the cloud computing definition. during this context, consultants gift a set of twenty one dentitions of cloud computing. supported these dentitions, we are able to dine cloud computing as an outsized scale distributed computing paradigm supported virtualized computing and storage resources, and trendy net technologies. Over the web network, cloud computing provides climbable and abstracted resources as services. of these services area unit on demand and oared on a pay-per-use model. Cloud computing is commonly given at 3 levels: Infrastructure as a Service (IaaS) wherever purchasers will deploy their own package, Platform-as-a-Service (PaaS) wherever purchasers will program their own applications mistreatment the services of the PaaS and package as a Service (SaaS) wherever purchasers use existing applications of the cloud. during this paper, we tend to specialize in the PaaS level. many PaaS commercial2 Rami Sellami and Bruno Defude solutions exist (e.g. Microsoft Azure [2], Salesforce Force.com [3], Google App Engine [4], etc.) and plenty of analysis comes area unit seeking to boost the safety, measurability, snap and ability within the PaaS. Some researches, massive table [5], PNUTS, Dynamo, etc., take into ac-count the information management within the cloud. withal, not sucient to deal with all the goals of information management within the PaaS that are [8]: (i) de-signing climbable management architectures, (ii) sanctionative snap and fewer complexness throughout databases migration, and (iii) coming up with intelligent and involuntary software. Indeed, associate application uses in most cases one knowledge Base Management System (DBMS or knowledge store) to manage its knowledge. This software is meant to support the entire wants of associate application. Thus, it appears unreal to nd one software that eminently supports numerous applications with deterrent necessities in terms of information management. We focus, during this paper, on existing solutions of the progressive supporting multiple knowledge stores within the cloud. a lot of exactly, our contributions area unit (i) American state scribe different situations regarding the manner applications use knowledge stores in an exceedingly PaaS (ii) de ne necessities of applications in an exceedingly PaaS (iii) analyze and classify existing works on cloud data management, that specialize in multiple knowledge stores necessities.

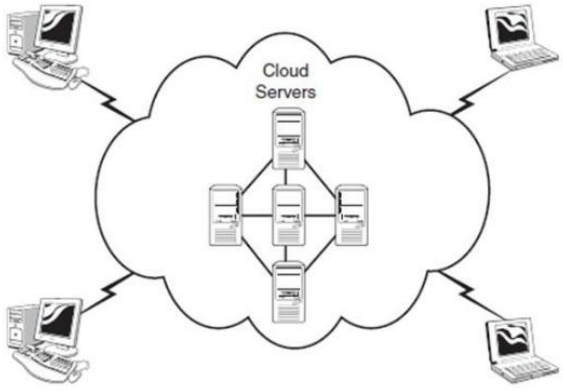


Fig1: Cloud Computing architecture

II. RELATED WORKS

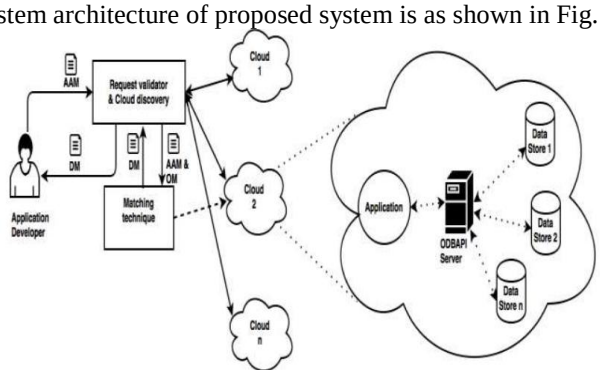
Cloud computing creates an oversized range of security problems and challenges. a listing of security threats to cloud computing is conferred in [5]. These problems vary from the desired trust within the cloud supplier and attacks on cloud interfaces to misusing the cloud services for attacks on alternative systems. the most drawback that the cloud computing paradigm implicitly contains is that of secure outsourcing of sensitive also as business-critical information and processes. once considering employing a cloud service, the user should remember of the actual fact that everyone information given to the cloud supplier leave the own

management and protection sphere. Even more, if deploying data- process applications to the cloud (via IaaS or PaaS), a cloud supplier gains full management on these processes. Hence, a robust trust relationship between the cloud supplier and therefore the cloud user is taken into account a general necessity in cloud computing. looking on the political context this trust could bit legal obligations. as an instance, Italian legislation needs that government information of Italian voters, if collected by official agencies, need to stay at intervals Italy. Thus, employing a cloud supplier from outside of Italy for realizing associate e-government service provided to Italian voters would instantly violate this obligation. Hence, the cloud users should trust the cloud supplier hosting their information at intervals the borders of the country associated ne'er repeating them to an off -country location nor providing access to the information to entities from abroad. associate assaulter that has access to the cloud storage element is in a position to require snapshots or alter data within the storage. This can be done once, multiple times, or unceasingly. associate assaulter that additionally has access to the process logic of the cloud also can modify the functions and their input and output information. albeit within the majority of cases it's going to be legitimate to assume a cloud supplier to be honest and handling the customers' affairs in an exceedingly respectful and accountable manner, there still remains a risk of malicious staff of the cloud supplier, booming attacks and compromisation by third parties, or of actions ordered by a subpoena. In [6], an outline of security flaws and attacks on cloud infrastructures is given. Some examples and more modern advances area unit shortly mentioned within the following. Ristenpart et al. [7], [8] conferred some attack techniques for the virtualization of the Amazon EC2 IaaS service. In their approach, the assaulter allocates new virtual machines till one runs on an equivalent physical machine because the victim's machine. Then, the assaulter will perform cross-VM facet channel attacks to be told or modify the victim's information. The authors gift ways to achieve the required victim machine with a high likelihood , and show a way to exploit this position for extracting confidential information, e.g., a crypto graphical key, from the victim's VM. Finally, they propose the usage of dazzling techniques to fend cross-VM side-channel attacks.

III. IMPLEMENTATION DETAILS

A. System Architecture

System architecture of proposed system is as shown in Fig.1



B. Application Developer:

Application developer sends Abstract Application Manifest (AAM) JSON to request validate and cloud discovery for getting cloud information. This request contains all cloud details, application requirement and DB requirement. Request Validate and Cloud Discovery: Upon receiving request from application developer this module first validates the request and after successful validation sends request to cloud matching technique module for selecting best match within list of available cloud. It sends offer manifest(OM) JSON to cloud matching technique with list of cloud requirement and application requirement.

C. Matching Technique:

This technique reads list of cloud requirement from offer manifest JSON and selects best suiting cloud from list of available cloud. After selecting the cloud this module deploy the application on that cloud and return address information of cloud to query data store technique. On receiving cloud information from cloud matching technique, request validate and cloud discovery modules returns success response to application user along with address of cloud through deployment manifest(DM). Abstract application manifest: This manifest contains 2 classes of requirements. First, needs in term of data stores. The developer provides 5 info regarding the specified data stores such as type, name, version, size and the query type to execute. It's worth noting that once the developer fills this manifest, he has the freedom to specify one or multiple information. For each info, he offers a constraint expressed by a constanvalue, a joker (denoting any values) or certain conditions (stated as inequalities). Hence, a lot of flexibility within the model is ensured. Fig.2. depicts structure of abstract application manifest.

D. Modules

- Unifying data model
- REST API/services
- Virtual data stores

Module description

Unifying data model:

We define a data model which abstracts from the underlying (explicit/implicit) integrated data store models, and provide a common and unified view so that developers can define their queries over heterogeneous data stores. During the development step, the developers dispose of a global data model expressed according to our unifying model and which integrates local data store models. Our unifying data model decouples query definitions from the data stores specific languages.

REST API/services:

Based on our unifying data model, we define a resource model upon which we develop a REST API, called ODBAPI, enabling to interact with involved data stores in a unique and uniform way. Each data store will be then wrapped behind a REST service implementing ODBAPI. Our API decouples the interactions with data stores from

their specific drivers. By using our unifying data model to express the queries and ODBAPI to interact with the data stores, developers do not have to deal with various languages and APIs and do not have to adapt their code.

Virtual data stores:
Wrapper REST services enable executing simple queries over the involved data stores. However, they are not meant to execute complex queries (such as join, union, etc.). In our approach, we consider virtual data store (VDS for short) a specific component responsible for executing queries submitted by a multiple data store application. A VDS holds the global data model integrating the different data stores and which is specified according to our unifying data model and a set of correspondence rules.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we have a tendency to planned a generic approach to facilitate the developer task and change the event of applications victimization multiple knowledge stores whereas remaining agnostic to those latter. we have a tendency to introduced 3 solutions: eight ODBAPI for CRUD operations: we have a tendency to outlined a generic resources model to represent the various components of heterogeneous knowledge stores in an exceedingly Cloud surroundings. supported this, we have a tendency to outline a novel REST API that permits the management of the delineate resources in an exceedingly uniform manner. This API is termed ODBAPI and permits the execution of CRUD operations on relative and NoSQL knowledge stores. The highlights of ODBAPI area unit twofold: (i) decoupling cloud applications from knowledge stores so as to facilitate their development and their migration, and (ii) easing the developers task by lightening the burden of managing completely different Apis. it's noteworthy that within the current version of ODBAPI server, we have a tendency to took into consideration four knowledge stores: MySQL, Riak, CouchDB, and MongoDB. eight Virtual knowledge stores for complicated queries execution: we have a tendency to planned virtual knowledge stores to execute complicated queries (including joins) across NoSQL and relative knowledge stores. For this purpose, we have a tendency to outlined a unifying knowledge model ready to describe the heterogeneous knowledge models of knowledge stores. it's utilized by the user to specific his complicated question and by the virtual knowledge store to method it. Once a virtual knowledge store receives a fancy question, it constructs AN best question execution arrange, composed by sub-queries at the extent of target knowledge sources, conversion and shipping operations and a final question recombining partial results._ Manifest for knowledge stores discovery and automatic application deployment: Once the developer has completed the event of his application, we have a tendency to provided him the likelihood to specific his application necessities in terms of knowledge stores within the abstract application manifest. Then, he sends it to the matching module that interacts with the cloud suppliers discovery module to elect the suitable cloud supplier to the appliance necessities. Indeed, the cloud suppliers discovery module discovers the capabilities of knowledge stores of every cloud supplier and returns these capabilities within the provide manifest. supported that, the matching module implements the matching algorithmic program so as to elect the adequate cloud supplier to the appliance necessities and generates the preparation manifest of the appliance. Once it's done, we have a tendency to deploy the appliance victimization the COAPS API that takes as input the preparation manifest.
Currently, we have a tendency to area unit acting on applying ODBAPI and the virtual knowledge store question improvement and execution approach to different qualitatively and quantitatively varied eventualities within the OpenPaaS project. this permits North American country to spot potential discrepancies and create our work additional reliable for public use. additionally, we have a tendency to aim to check AN implementation for Hive permitting access to Hadoop knowledge stores. Our second perspective consists in providing another matching algorithmic program supporting approximate matching. therefore we have a tendency to change additional flexibility in knowledge stores discovery and applications preparation. Our third perspective is AN extension to virtual knowledge stores, permitting to support a bigger category of complicated queries across NoSQL and relative knowledge stores (union, intersection, aggregates, cluster by like operations) and introducing additional elaborate question process improvement techniques, as well as asynchronous analysis.

REFERENCES

[1] Rami Sellami, Sami Bhiri, and Bruno Defude ”Supporting multi data stores applications in cloud environments”,IEEE Transactions on Services Computing 2015
[2] C. Baun, M. Kunze, J. Nimis, and S. Tai, ”Cloud Computing -Web-Based Dynamic IT Services”, Springer, 2011.
[3] T. Kraska, M. Hentschel, G. Alonso, and D. Kossmann, ”Consistency rationing in the cloud: Pay only when it matters,” PVLDB, vol. 2, no. 1, pp. 253-264, 2009.
[4] R. Sellami, S. Bhiri, and B. Defude, ”ODBAPI: a unified REST API for relational and NoSQL data stores,” in The IEEE 3rd International Congress on Big Data (BigData’14), Anchorage, Alaska, USA, June 27-July 2, 2014,2014.
[5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. zonwinski,G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the clouds: A berkeley view of cloud computing,” University of California, Berkeley, Tech. Rep. UCBEECS-2009-28,Feb 2009.
[6] Mohiuddin Ahmed, Abu Sina Md. RajuChowdhury, Mustaq Ahmed, “An Advanced Survey on Cloud Computing and State -of-the -art Research Issues,” In ternational Journal of Computer Science Issues, Vol.9, No.1 pp. 201-207, 2012.
[7] http://www.asigra.com/blog/cloud-types-private

-public-and-hybrid

[8] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," Proc. 41st Ann. ACM Symp. Theory Computing (STOC 09), ACM, pp. 169-178.2009.

[9] "Security Policy and Key Management: Centrally Manage Encryption Key". Slideshare.net. 2012 -08-13. Retrieved 2013-08-06.

[10] E. Bertino, F. Paci, and R. Ferrini, "Privacy - Preserving Digital Identity Management for Cloud Computing," IEEE Computer Society Data Engineering Bulletin ,Mar. 2009, pp. 1–4.

[11] M. Ko, G.-J. Ahn, and M. Shehab "Privacy - Enhanced User-Centric Identity Management," Proc. IEEE Int'l Conf. Communications , IEEE Press, 2009, pp. 998–1002

