An Application to Support Multi Database in Cloud Environments

[1]A. SivaNarayana Reddy [2]T. Anuja

[1]M.Tech Student, Department of CSE, Nalanda Institute of Engineering And Technology, Village Kantepudi, Mandal Sattenapalli, Dist Guntur, A.P, India.

[2]Assistant Professor, Department of CSE, Nalanda Institute of Engineering And Technology, Village Kantepudi, Mandal Sattenapalli, Dist Guntur, A.P, India.

ABSTRACT─ ***Advances in internet technology and also the proliferation of mobile devices and sensors connected to the web have resulted in large process and storage needs. Cloud computing has emerged as a paradigm that guarantees to fulfill these needs. This work focuses on the storage side of cloud computing, specifically on data management in cloud computing. Traditional relational databases were designed in an exceedingly totally different hardware and software era and are facing challenges in meeting the performance and scale needs of large knowledge. During this paper we have a tendency to propose an integrated set of models, algorithms as well as tools aiming at alleviating developers task for developing, deploying and migrating multiple knowledge stores applications in cloud computing. Our approach focuses primarily on 3 points. First, we offer a unifying knowledge model utilized by applications developers to act with heterogeneous relative and NoSQL knowledge stores. Supported that, they categorical queries use OPEN-PaaS-DataBase API (ODBAPI), distinctive REST API permitting programmers to write their applications code severally of the target knowledge stores. Second, we have a tendency to propose virtual knowledge stores that act as an intercessor and interact with integrated knowledge stores wrapped by ODBAPI. In the present***

version of ODBAPI server, we took into account four data stores: MySQL, Riak, CouchDB, as well as MongoDB.

1. INTRODUCTION

Cloud computing is a growing term that describes the development of several existing technologies and strategies to computing into something exclusive. The NIST defined cloud computing as "Cloud computing is a mannequin for enabling effortless, on-demand network access to computing assets (e.g., networks, servers, storage, functions, and offerings) that can be quickly provisioned as well as released with minimal administration effort of service provider interaction". In the average computing device computing, we run copies of programs in our procedure. The files that we created are stored in the same process i.e. PC centric. With cloud computing, the program packages aren't run from our laptop instead saved on servers and accessed through the internet i.e., PC centric. The advantage is that if our approach crashes, the application remains to be to be had. The equal thing will occur in the case of the information that saved within the cloud. Anyone with permission can access and alter the cloud information. "Cloud" includes gigantic quantity of computers and servers, linked and available via web. In latest years, advances in internet science and the proliferation of sensors and mobile gadgets related to the internet have resulted in the new release of big data units that ought to be processed as well as

stored. Just on Facebook, 2.4 billion content material objects are shared among buddies daily. In these days, firms generate big quantity of knowledge which has grown too massive to be managed and analyzed via usual data processing requirements. Indeed, traditional relational database management systems (RDBMS) have been designed in an era when the on hand hardware, as well because the storage and processing requisites, had been very extraordinary than they're in these days. For this reason, these solutions had been encountering many challenges in assembly the efficiency as well as scaling tools of this “Big Data" certainty. Big data is a time period used to refer to large and complicated datasets made from a form of data structures, including structured, semi-structured, and unstructured information. According to the Gartner staff, massive knowledge can also be defined by 3ways: quantity, rapidity, and multiplicity. at present, businesses are aware that this large volume of knowledge will be accustomed generate new opportunities and method enhancements through their process and analysis. At regarding identical time, cloud computing has conjointly emerged as a process paradigm for on-demand network access to a shared pool of computing resources (e.g., network, servers, storage, applications, as well as services) which will be quickly provisioned with smallest management effort. Cloud computing is related to service provisioning, within which service providers provide computer-based services to customers over the network. Generally these services are supported a pay-per-use model wherever the consumer pays just for the resources used. Overall, a cloud computing model aims to produce advantages in terms of lesser up-front investment, lower in operation costs, higher measurability, and physical property, quick access through the online and reduced business risks and maintenance expenses. Because of such characteristics of cloud computing, several applications are created in or migrated to cloud environments over the previous couple of years. In fact, it's fascinating to note the extent of activity between the process needs of huge information applications, and also the accessibility and measurability of computational resources offered by cloud services. Yet, the effective leverage of cloud infrastructure needs careful style and implementation of applications and information management systems. Information stores as cloud information management systems, the large variety of existing resolutions and also the discrepancies among them create it troublesome to formulate a perspective on the domain and even more difficult to pick the acceptable solution for a retardant at hand. This survey reviews NoSQL and NewSQL information stores with the intent of filling this gap. Additional specifically, this survey has the subsequent objectives:

1. To compare the characteristics of the leading solutions so as to produce steering to practitioners and researchers to decide on the acceptable information store for specific applications
2. To identify analysis challenges and opportunities within the field of large-scale distributed information management.

2. RELATED WORK

Dr Elaine Shi represented many enabling technologies towards this vision. Specifically, she told concerning 1) how to safeguard users' information against probably compromised applications; 2) how to safeguard users' knowledge

against a doubtless compromised computation supplier; and 3) how to safeguard users' knowledge against a doubtless compromised storage provider. She told concerning our current effort at group action these technologies to supply a cloud infrastructure that provides information security at the platform level. During this manner, users will enjoy the wealthy cloud applications without concern about the privacy of their data; and application developers will specialize in developing practicality whereas offloading the burden of providing security as well as privacy to the cloud platform. Performance consistent with a recent survey, forty ninth of users abandons a web site or switch to a competition when experiencing performance problems. And also the want for speed is barely increasing: in 2000, a typical user was willing to attend 8 sec for a webpage to load earlier than navigating away; by 2009, that number dropped to 3 sec.

Previous analysis work handling N1NF recursive models includes Abiteboul and Bidoit's model that projected a Non-First traditional type information model known as the Verso model that permits knowledge restructuring. Arbitrary projections are often achieved however they typically need a restructuring of the initial relation. 2 versions of the selection operation are outlined, a straightforward version of the selection operation, the Verso-selection as well as an extension of the selection, known as the "super-selection" which may be expressed by the Verso-selection, projection, and be a part of operations. The restriction operation is itself restricted in this it is often applied only to the "root" of the format. The Cartesian product operation needs the primary quantity to be an instance over a flat relation and this can be once more a major weakness. what is more, the key feature of their model, the restructuring operation, cannot reconstruct entirely the structures of the relations while not loss of knowledge, even once employing a combination of all 3 transformations, root as well as branch permutations, compactions as well as extensions. As a result, the potentiality of the operation is restricted to a restricted specter of cases.

In Roth, Korth and Silberschatz's model the divided normal form (PNF) property is outlined for nested relations. A relation R is in PNF if all the atomic attributes of R form a key for the relation and recursively, every relation-valued attribute of the relation is additionally in PNF. The simplicity and clarity of relations in PNF is obvious, additionally because the fact those relations in PNF have some smart properties compared to alternative relations. However, in general, relations in PNF impose 2 vital restrictions, that there's a minimum of one atomic attribute at each nesting level of the relation and conjointly that relation valued attributes cannot be a part of the key. 2 new operators, nest and unnest, are added to the fundamental set of operators. This approach includes a variety of limitations as bestowed. What is more, the algebraic operators are outlined in such some way that works among the category of PNF relations and so, they're closed only under PNF relations. Additionally, projection, selection, is a part of and Cartesian product operations can't be applied to sub-relations of nested relations.

3. FRAMEWORK

A. System Architecture

i. ODBAPI

In our proposed system we are using the REST API. Based on our unifying data model, we tend to outline

a resource model upon that we tend to develop a REST API, known as ODBAPI, enabling to act with concerned data stores in an exceedingly unique and uniform manner. Every data store are then wrapped behind a REST service implementing ODBAPI. Our API decouples the interactions with data stores from their specific drivers. By victimization our unifying data model to precise the queries and ODBAPI to act with the data stores, developers don't need to influence varied languages and APIs and don't need to adapt their code once migrating their applications.

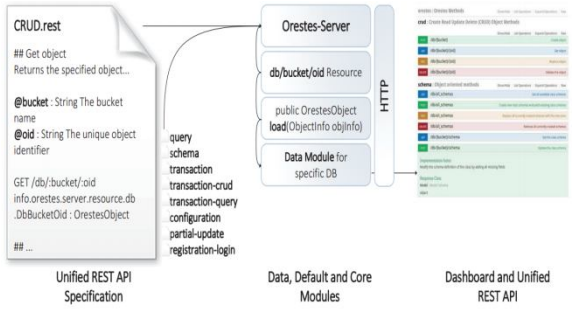


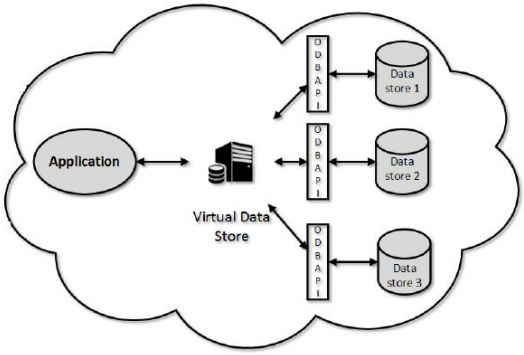
Figure1: REST API Modules

Request	Response	Explanation
POST /db/:bucket JSON-Object	Created object including assigned object id (oid) and version number	Creates a new object.
PUT /db/:bucket/:oid JSON-Object	Created/replaced object	Replaces or creates an object using an object id. The request can be conditioned on a version.
GET /db/:bucket/:oid	Database object	Fetches or revalidates an object. Request can be answered by web caches.
GET /db/:bucket?query &start=0&count=-1	List of matching ids	Executes a DB-specific ad-hoc query.
GET /db/all_schemas	All schemas for all classes	Retrieves all schemas.
POST /transaction	ID and URL of the transaction	Starts a new transaction.
POST /db/:bucket/:oid/:field	Success or validation failure	Performs a partial update (e.g. counter increase).

Figure2: Example requests from different REST API modules

ii. Virtual Data Stores

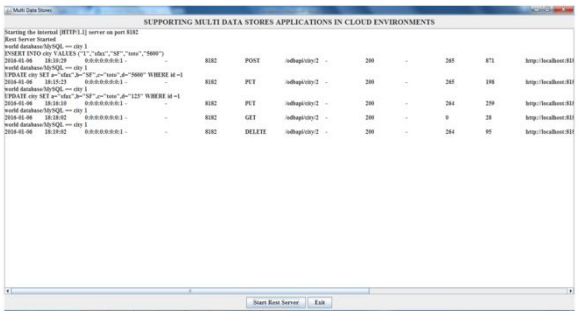
Wrapper REST services alter execution easy queries over the concerned data stores. However, they're not meant to execute advanced queries (such as join, union, etc.). In our approach, we have a tendency to take into account virtual data store (VDS for short) a selected element accountable for execution queries submitted by a multiple data store application.



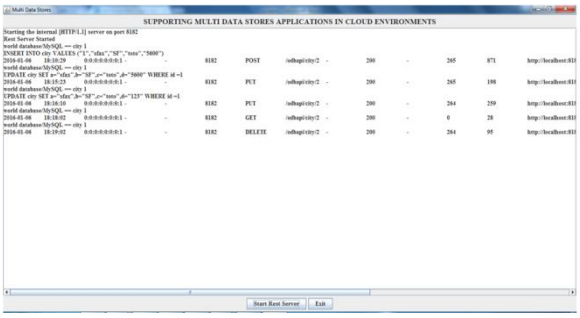
A multiple data store application submits CRUD and complex queries to the VDS that is accountable of their execution by interacting with acceptable data stores via their REST services. VDSs alter developers to specific their part of queries over multiple data stores during a declarative manner and take in charge the burden of their executions.

4. EXPERIMENTAL RESULTS

In our experiments, we need to start the server and then go to the MySQL database. And in the MySQL we will create a database. After completion of database creation, we need open the REST Client API. Then we can perform the CRUD operations. Then data will be stored into the MySQL database. Lastly, we can view the all operations in the server which are performed on REST Client API.



To connect Mongo DB, we have to create the folder in C Drive. Then open the command prompt and path change to mongo DB as well as executes the command (that is already in the code). Then, we need to open another command prompt as well as convert the paths to mongo DB and in that enter the "mongo".



Finally go to the REST client API and perform the CRUD operations. Lastly, we can view the all performed operations in the server.

5. CONCLUSION

In this paper we conclude that, we develop OPEN-PaaS-DataBase API (ODBAPI) to connect the multi data stores. The ODBAPI have two advantages: i) Decoupling cloud applications from data stores so as to facilitate their development and their migration, ii) Easing the developer's task through lightening the load of managing completely different APIs. In this project we took two relational databases those are MySQL and Mongo DB. And we can perform the CRUD operations of these data stores on REST Client API in cloud environment.

REFERENCES

[1] C. Baun, M. Kunze, J. Nimis, and S. Tai, Cloud Computing - Web- Based Dynamic IT Services. Springer, 2011.

[2] A. McAfee and E. Brynjolfsson, "Big data: The management revolution. (cover story)." Harvard Business Review, vol. 90, no. 10, pp. 60–68, 2012.

[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] S. Abiteboul and N. Bidoit, "Non first normal form relations: An algebra allowing data restructuring," J. Comput. Syst. Sci., vol. 33, no. 3, pp. 361–393, 1986.

[6] D. Kossmann, "The state of the art in distributed query processing," ACM Comput. Surv., vol. 32, no. 4, pp. 422–469, Dec. 2000.

[7] M. Sellami, S. Yangui, M. Mohamed, and S. Tata, "Paasindependent provisioning and management of applications in the cloud," in 2013 IEEE Sixth International Conference on Cloud Computing, Santa Clara, CA, USA, June 28 - July 3, 2013, 2013, pp. 693–700.

[8] R. Sellami and B. Defude, "Using multiple data stores in the cloud: Challenges and solutions," in Data Management in Cloud, Grid and P2P Systems - 6th International Conference, Globe 2013, Prague, Czech Republic, August 28-29, 2013. Proceedings, 2013, pp. 87–98.

[9] M. Pollack, O. Gierke, T. Risberg, J. Brisbin, and M. Hunger, Eds., Spring Data. O'Reilly Media, October 2012.

[10] P. Atzeni, F. Bugiotti, and L. Rossi, "Uniform access to nonrelational database systems: The sos platform," in Advanced Information Systems Engineering - 24th International Conference, CAiSE 2012, Gdansk, Poland, June 25-29, 2012. Proceedings, 2012, pp. 160– 174.