RESEARCH ARTICLE OPEN ACCESS

Introducing Novel Graph Database Cloud Computing For Efficient Data Management

[1]Dr.Arunkumar B R, [2]Komala R
Prof. and Head, Dept. of MCA, BMSIT, Bangalore and Ph.D. Research supervisor, VTU RRC, Belagavi
Asst. Professor, Dept. of MCA, Sir MVIT, Bangalore and Ph.D. Research Scholar,VTU RRC, Belagavi

ABSTRACT
Graph theory stands as a natural mathematical model for cloud networks, axiomatic cloud theory further defines the cloud with formal mathematical model. keeping axiomatic theory as a basis, paper proposes bipartite cloud and proposes graph database model as a suitable database for data management .it is highlighted that perfect matching in bipartite cloud can enhance searching in bipartite cloud.
Keywords:Graph database cloud,cloudcomputing,bipartite graph.

I. Introduction

Cloud computing (CC) is technically developing business model which offers several advantages over long term traditional model of computing. The CC is set to revolutionize the IT industry offering essential cost effective and yet secured services. This internet revolution technology, in its very simple terms can be thought of delivery of pooled network resources such as CPU, RAM, Storage, and Software over the web. These services can be quickly providedand released on demand. Today, hosting companies are providing cloud servers, cloud storage, software hosted on the cloud environment. Cloud computing helps the business to get the required computing resources at very minimal cost.

A major component of many cloud services is query processing on data stored in the underlying cloud cluster. Therefore, data management in CC has attracted many researchers. Graph theoretical structures such as Graph Database, Bipartite Graphcan play significant key roles in modeling and data management.

This emerging technology which is innovative and cost effective is getting added new dimensions of services. Hence, it is difficult to explain the CC with a single comprehensive definition. Nevertheless, there are numerous informal definition of cloud computing are available in the literature.The U.S. National Institute of Standards and Technology definition, stress the spirit of cloud computing : ―Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

1.1 Cloud structure

Conversely, exact axiomatic, formal mathematical/ graph theoretic concepts are found rarely in the field of cloud computing.Even so, significant work doneby Joe Weinman need attention. His, "Axiomatic cloud theory", define the cloud as a structure, CS (S,T,G,Q,d, q0) satisfying five formal axioms: it must be 1) Common, 2) Location-independent, 3) Online, 4) Utility, and 5) on-Demand. S is space, Tis time; G = (V, E) is a directed graph; Qis a set of states, where each state combines assignments of resource capacity and demand, resource allocations, node location, and pricing; q0 is an initial state; and d is a transition function that determines state trajectories over time: mapping resources, allocations, locations, and pricing to a next state of resources, allocations, locations, and pricing. This captures the interrelationships in a real cloud: capacity relative to demand can drive pricing, pricing and resource location drive allocation, allocation patterns can drive new resource levels[3].
Further, "Axiomatic cloud theory" identifies some aspects of finite state automata, Turing machines, Petri nets, and other constructs in the model. Precisely, JoeWeinman mathematically model is as follows:
Definition: An allocation A is feasible at time t if and only if

$$\forall v \in V \ , \sum_{u \in lv} A_{uv}(t) + R_v(t) - \sum_{x \in O_v} A_{vx}(t) \geq 0$$

[3]

Definition of "Common:"A cloud structure (S, T, G, Q,δ, q_0) is common if and only if there at least two non-zero flows that are commonly sourced.[3]

Definition of "Location independence: "A completely feasible cloud structure (S, T, G,Q,δ, q_0)

is location independent if and only if $\forall t \in T, if q = \delta(t) = (R_q, A_q, L_q, P_q)$ and total price at time t is $\hat{p}_q$, then $\forall L' \in M^v, \exists A' \in R^{rE}$, such that if $q' = (R_q, A', L', P_q)$, A' is feasible and $\hat{p}_q$ (t)≤ $\hat{p}_q$ (t) [3]

Definition of "Online: "A cloud structure (S,T,G,Q,δ, q_0) is online if and only if G is weakly connected ,i.e for $\forall u, v \in V$, there is semi path from u to v.[3]

Definition of "Utility:"A cloud structure (S, T, G, Q, δ, q_0) is utility if and only if $\forall_e \in E, \forall t \in T$, q=$\delta(t) \rightarrow P_q(e)$is linear in A_q (e) and A_q(e) and $A_q(e) \neq 0 \rightarrow P_q(e) \neq 0$ [3]

Definition of "on-Demand": A cloud structure (S, T, G, Q, δ, q_0) is on –Demand if and only if it is completely feasible.[3]

This approach may be viewed as a first step towards formal modeling of clouds and further there is a scope to work on this model with different thoughts. This research work keeps "Axiomatic cloud theory" as a basis.This paper introduces and discuss the novel Graph Database Cloud (GDC), applies the Bipartite graph (BG).

1.2 CloudGraph database?

Traditional database such as relational database where schema is fixed is no longer suitable for computing application which manages massive amount of data generated [1][4]. The massive amounts of data generated using countless numbers of servers results in data spread across multiple machines; traditional relational **SQL JOIN** (used to combine rows from multiple tables) operations are just not possible.

There is a need for storage system that provides index-free adjacency so that every element in the database contains a direct link to its adjacent element, no index lookups are required; every element (or node) knows what node or nodes it is connected with, this connection is called an edge. This allows graph database systems to replace traditional relational database completely or partially and further, utilize graph theory to very rapidly examine the connections and interconnectedness of nodes.

The power of the edge allows a graph database to find results in associative data sets – data where information is the aggregation of links between nodes – faster than relational databases. Graph databases can scale more naturally to large data sets and to datasets with changing or on-the-fly schemas. On the other hand, relational databases are still better at performing the same operation on large numbers of identical data. When you want your bank balance, you don't want a rapid list of all your transactions – just your bottom line.

The use of graph databases is rapidly spreading to many applications through the use of mixed-database approaches, where a graph search is used to identify the extent of the data, and a subsequent relational search is used to provide the detailed analytics. While this approach presently involves developing (and supporting) two database structures, it yields rapid response and targeted data analysis. Some solutions present the graph results to users while the analytics are being pulled and crunched; other systems serve up old results while the new results are being calculated. How analytics and associations coexist is one of the considerations that must be made when architecting your solution.

It is needless to say that the graph database is the future especially for big data analytics and cloud computingwherever huge data get generated. If a third or more of your relational tables describe links between data elements, the database is heavily associative, and can be a graph database candidate. The final decision requires a complete analysis of how the data is being used, volume and growth patterns, and not just a review of table structures. If your data is used for statistical analysis, data mining and exploration, or operational research, the relational database approach is still at least part of the architectural solution. The next paragraph explains applying graph database to cloud computing.

The Graph Database CloudComputing (GDC) which is being introduced for cloud computing isbasedon the Graph Database model which is derived by the Glossary of Big Data Terminology by Author, Christopher Horne. Here,database usesgraph structures for semantic queries with nodes, edges, and properties to represent and store data. A graph database cloud (GDC)is any storage system that provides index-free adjacency. This means that every element contains a direct pointer to its adjacent elements and no index lookups are necessary. General graph databases that can store any graph are distinct from specialized graph databases such as triple stores and network databases.

In a connected cloud computing world, isolated/remote pieces of information are not expected, but rich, connected domains. Core aspect of GDC data model is able to store, process, and query connections efficiently. While other databases compute relationships expensively at query time, a GDCdatabase stores connections as first class citizens, readily available for any "join-like" navigation operation. Accessing those already persistent connections is an efficient, constant-time operation and allows cloud computing to quickly traverse millions of connections per second per core.

Independent of the total size of your dataset, Graph Cloud Databases do extremely well at managing highly connected data and complex queries. Equipped only with a pattern and a set of

starting points, Graph Cloud Databases explore the larger neighborhood around the initial starting points — collecting and aggregating information from millions of nodes and relationships — leaving the billions outside the search perimeter untouched.
Graph Cloud databases are based on graph theory concepts, employing nodes, properties and edges. Nodes represents data/item that CC want to follow, Properties are relevant information that relate to nodes. For instance, if "Wikipedia" were one of the nodes, one might have it tied to properties such as "website", "reference material", or "word that starts with the letter 'w'", depending on which aspects of "Wikipedia" are pertinent to the particular database.

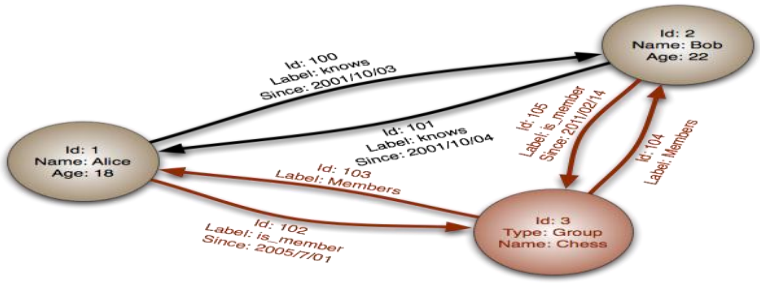

Figure1: Graph database [7]

Edges are the lines that connect nodes to nodes or nodes to properties and they represent the relationship between the two. Most of the important information is really stored in the edges. Meaningful patterns emerge when one examines the connections and interconnections of nodes, properties, and edges.

Nodes can be tagged with labels representing their different roles in your domain. In addition to contextualizing node and relationship properties, labels may also serve to attach metadata—index or constraint information—to certain nodes.

Edge Relationships provide directed, named semantically relevant connections between two node-entities. An edge relationship always has a direction, a type, a *start node*, and an *end node*. Like nodes, relationships can have any properties. In most cases, relationships have quantitative properties, such as weights, costs, distances, ratings, time intervals, or strengths. As relationships are stored efficiently, two nodes can share any number or type of relationships without sacrificing performance. Note that although they are directed, relationships can always be navigated regardless of direction.

Compared with relational databases, graph databases are often faster for associative data sets and map more directly to the structure of object-oriented applications. They can scale more naturally to large data sets as they do not typically require expensive join operations. As they depend less on a rigid schema, they are more suitable to manage ad hoc and changing data with evolving schemas. Conversely, relational databases are typically faster at performing the same operation on large numbers of data elements.

Graph databases are a powerful tool for graph-like queries, for example computing the shortest path between two nodes in the graph. Other graph-like queries can be performed over a graph database in a natural way.

II. Conclusion

This paper introduces the graph database concept to cloud computing as a complete replacement or hybrid approach which uses graph database as well as relational to some extent. Graph theory again proved its capability as natural model including cloud computing. The graph database concepts will revolutionize the processing of huge database such as used in cloud and big-data.

References

[1]. Diestel, Reinard (2005). "*Graph Theory, Grad. Texts in Math*."Springer.ISBN 978-3-642-14278-9.
[2]. Asratian, Armen S., Denley, Tristan M. J., Häggkvist, and Roland (1998), "*Bipartite Graphs and their Applications"*, Cambridge Tracts in Mathematics **131**, Cambridge University Press, ISBN 9780521593458.
[3] Axiomatic cloud Theory working paper,July 29,2011 Joe Weinman.
[4] http://www.seguetech.com/blog/2013/02/04/what-are-the-differences-between-relational-and-graph-databases
[5]. *www.graphdatabase.com*
[6]. *http://readwrite.com/2011/04/20/5-graph-databases-to-consider*
[7]. *https://id.wikipedia.org/wiki/Graph_database*