빅데이터환경에서의 그래프데이터베이스 활용방안

박승범[*] · 이상원[**] · 안현섭[***] · 정인환[****]

[*]한국정보화진흥원 경영기획부

[**]원광대학교 정보전자상거래학부(융복합창의연구소)

[***]브라운슈바이크공과대학 경영정보학과

[****]한성대학교 컴퓨터공학과

Application Plan of Graph Databases in the Big Data Environment

Sungbum Park[*] · Sangwon Lee[**] · Hyunsup Ahn[***] · In-Hwan Jung[****]

[*]Department of Management Planning, National Information Society Agency

[**]D. of Information & Electronic Commerce (I. of Convergence & Creativity), Wonkwang University

[***]Department of Wirtschaftsinformatik, Technische Universität Braunschweig

[****]Department of Computer Engineering, Hansung University

E-mail: parksb@nia.or.kr, sangwonlee@wku.ac.kr, hs.ahn@tu-bs.de, ihjung@hansung.ac.kr

요 약

관계형 데이터베이스가 많은 기업에서 널리 사용되고 있지만, 개체간의 관계를 효과적이고 효율적으로 관리하지는 못하고 있다. 빅데이터를 분석하기 위해서는 다양한 개체간의 관계를 그래프로 표현할 필요가 절실하다. 본 논문에서는 그래프 데이터베이스와 그의 구조를 정의하고, 트랜잭션, 일관성, 가용성, 검색 기능 및 확장 등의 그 특성에 대해 살펴본다. 또한, 그래프 데이터베이스를 적용해야할 분야와 적용하지 말아야 할 분야에 대해 살펴본다.

ABSTRACT

Even though Relational Databases have been widely used in many enterprises, the relations among entities are not managed effectively and efficiently. In order to analyze Big Data, it is absolutely needed to express various relations among entities in a graphical form. In this paper, we define Graph Databases and its structure. And then, we check out their characteristics such as transaction, consistency, availability, retrieval function, and expandability. Also, we appropriate or inappropriate subjects for application of Graph Databases.

키워드

Big Data, Graph Databases, Application Plan, NoSQL

I. Introduction

Relational Databases have been widely used in many enterprises. But, the relations among entities are not managed effectively and efficiently. Many enterprises absolutely need to express various relations among entities in a graphical form when analyzing Big Data. In this paper, we define graph databases and its structure. And then, we check out their characteristics such as transaction, consistency, availability, retrieval function, and expandability. The queries are explained by some examples of query languages such Gremlin and Cyper. Also, we appropriate or inappropriate subjects for application of column-family stores.

II. Graph Databases for Big Data

By definition, a graph database is a kind of storage system that provides index-free adjacency. Graph Databases are based on graph theory (Figure 1). Graph Databases employ nodes, properties, and edges. Nodes are very similar in nature to the objects.

Object-oriented programmers will be familiar with objects. So, a graph database is a database that uses graph structures with nodes, edges, and properties to represent and store data. Every element contains a direct pointer to its adjacent element. In addition, no index lookups are necessary. Rather than Relational Databases, Graph Databases are faster for associative data sets and map more directly to the structure of object-oriented applications. As they do not typically require expensive join operations, they can scale more naturally to large data sets. In the meantime, Relational Databases are typically faster at performing the same operation on large numbers of data elements, But, Graph Databases are more suitable to manage ad hoc and changing data with evolving schema. They depend less on a rigid schema. Like Column-Family Stores and Document Databases, Graph Databases guarantee consistency, transaction, availability, various queries, and extensibility of column-family data. Examples of Figure 2 and 3 are relations with attributes and node partition each.

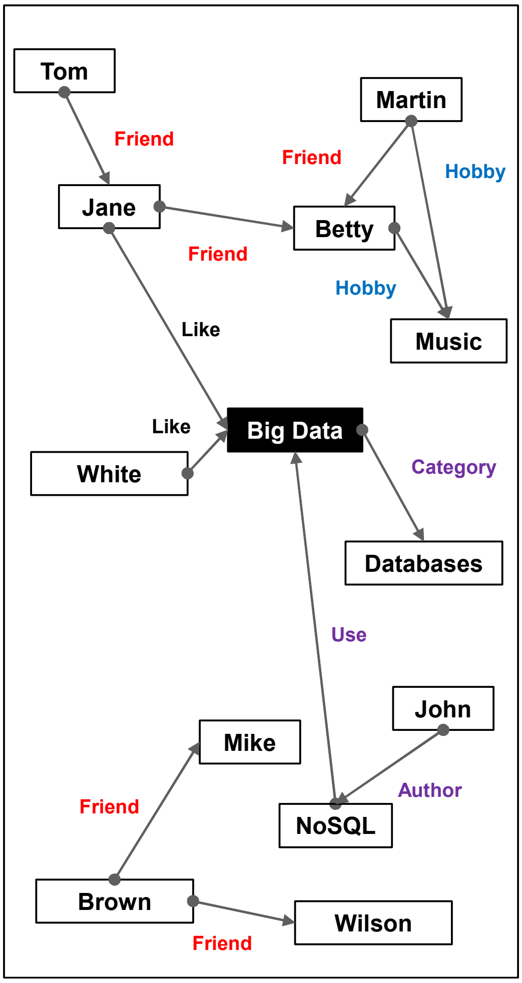

Figure 1. An Example of Graph Structure

III. Application Plan of Graph Databases for Big Data

The appropriate places of using Graph Databases are linked data, routing, dispatch, location-based service and recommendation service. But, Graph Databases do not support an optimal solution for entity update, especially in modifying entities or attributes.

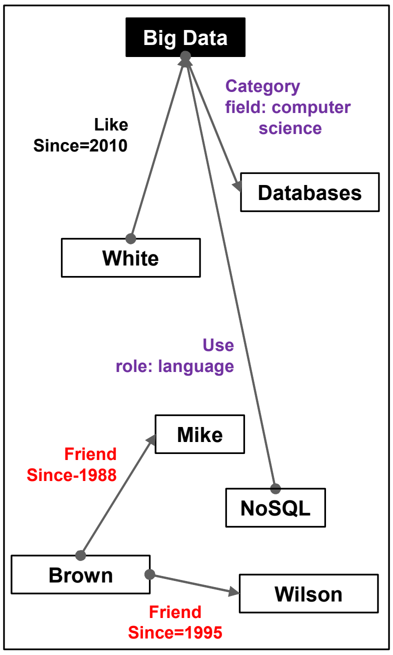

Figure 2. An Example of Relations with Attributes

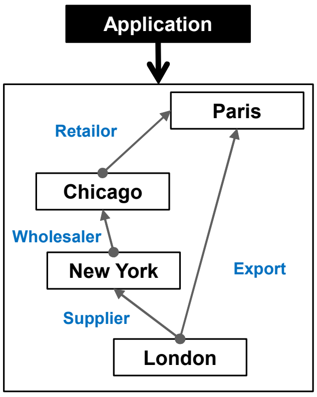

Figure 3. An Example of Node Partition

IV. Conclusions

Consequently, 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 such as graph' s diameter computations or community detection.

References

[1] A B M Moniruzzaman and Syed Akhter Hossain, “NoSQL Database: New Era of Databases for Big data Analytics - Classification and Characteristics and

Comparison," International Journal of Database Theory and Application, Vol. 6, No. 4, 2013.

[2] Burton H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," Communications of the ACM, Vol. 13, No. 1, pp. 422-426, 1970.

[3] Christof Strauch, NoSQL Databases, Stuttgart Media University Press, 2012.

[4] Edgar F. Codd, "A Relational Model of Data for Large Shared Data Banks," Communications of the ACM, Vol. 13, No. 6, pp. 377-387, 1970.

[5] John Ousterhout, Parag Agrawal, David Erickson, Christos Kozyrakis , Jacob Leverich, David Mazières, Subhasish Mitra, Aravind Narayanan, Guru Parulkar, Mendel Rosenblum, Stephen M. Rumble, Eric Stratmann and Ryan Stutsman, "The Case for RAMClouds: Scalable High-Performance Storage Entirely in DRAM," SIGOPS Operating Systems Review, Vol. 43, No. 1, pp. 92-105, 2010.

[6] Joshua Bloch, Effective Java - Programming Language Guide, Addison-Wesley Longman, 2001.

[7] Marcos K. Aguilera, Wojciech Golab and Mehul A. Shah "A Practical Scalable Distributed B-Tree," Proceedings of the VLDB Endowment, Vol. 2008, No. 1, pp. 598-609, 2008.

[8] Patrick O' Neil, Edward Cheng, Dieter Gawlick and Elizabeth O' Neil, "The Logstructured Merge-Tree," Acta Informatica, Vol. 33, No. 4, pp. 351-385, 1996.

[9] Pramod Sadalage and Martin Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley, 2012.

[10] Stavros Harizopoulos, Daniel J. Abadi, Samuel Madden and Michael Stonebraker, "OLTP through the Looking Glass, and What We Found Fhere," ACM SIGMOD International Conference on Management of Data, Vol. 2008, No.1, pp. 981-992, 2008.