빅데이터환경에서의 문서데이터베이스 활용방안

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

[*]한국정보화진흥원 경영기획부

[**]원광대학교 정보전자상거래학부(융복합창의연구소)

[***]브라운슈바이크공과대학 경영정보학과

[****]한성대학교 컴퓨터공학과

Application Plan of Document 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

많은 기업들은 경영활동 중에 매우 많은 데이터를 생성하고 관리하고 있다. 그러나, 관계형 데이터베이스, 계층형 데이터베이스, 네트워크 데이터베이스 같은 일반적인 데이터베이스로, 이러한 다량의 문서 데이터를 효과적이고 효율적으로 관리하고 분석하는 것은 불가능에 가깝다. 그래서, 본 논문에서는, 이러한 문제를 해결하기 위해, 우선 문서 데이터베이스를 정의하고, 일관성이나 트랜잭션 같은 특성들을 살펴본 후, 문서 데이터베이스를 적용해야 할 분야와 적용하지 말아야 할 분야에 대해 언급하고자 한다.

ABSTRACT

For Many enterprises are creating and handling huge amount of data in their business administration. However, it would be impossible for general databases such as Relational Databases, hierarchical databases, and network databases to manage and analyze this large amount of document data efficiently and effectively. So, in this paper, we define document databases and check out their characteristics such as consistency and transaction. And we propose appropriate or inappropriate subjects for application of document databases.

키워드

Big Data, Document Databases, Application Plan, NoSQL

I. Introduction

Although many enterprises are creating and handling huge amount of data in their business administration, it would be impossible for general databases such as Relational Databases, hierarchical databases, and network databases to manage and analyze this large amount of document data efficiently and effectively. Our research proposes appropriate or inappropriate subjects for application of document databases.

II. Document Databases

Many people consider Document Databases as the next logical step from simple key-/value-stores to slightly more complex and meaningful data structures as they at least encapsulate key-/value-pairs in documents. On the other hand, there is no strict schema documents have to conform to which eliminates the need schema migration efforts.

Documents in the form of XML, JSON, BSON, and so on are major concept of Document Databases. Even though Relational Databases should define all columns, Document Databases do not demand it. In inserting a new attribute, there is no need to define the attribute and change the existing document.
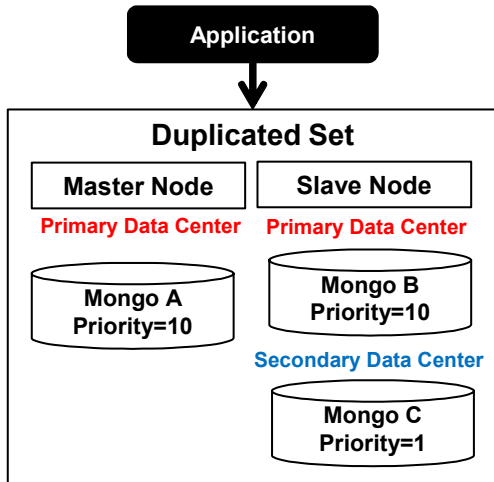
Application

Duplicated Set

| Master Node | Slave Node |

Primary Data Center | Primary Data Center

Mongo A
Priority=10

Mongo B
Priority=10

Secondary Data Center

Mongo C
Priority=1

Figure 1. Assignment of a Priority

Application

Duplicated Set

| Master Node | Slave Node |

Primary Data Center | Primary Data Center

Mongo A
Priority=10

Mongo B
Priority=10

Secondary Data Center

Mongo C
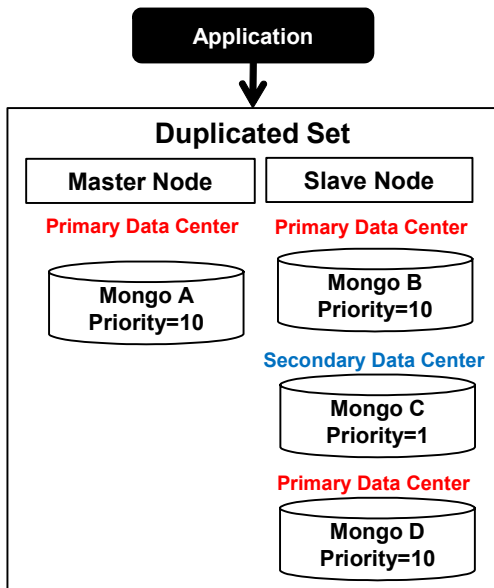Priority=1

Primary Data Center

Mongo D
Priority=10

Figure 2. Insertion of a Node

The most popular Document Databases are MongoDB, CouchDB, Terrastore, OrientDB, RavenDB, Notes Storage Facility, and so on. We use MongoDB in this paper. MongoDB is a schema-free document database written in C++ and developed in an open-source project. The project is driven mainly by the company 10Gen Inc. that also offers professional services around MongoDB. According to its developers the main goal of MongoDB is to close the gap between the fast and highly scalable key-/value-stores and feature-rich traditional RDBMSs relational database management systems. MongoDBs name is derived from the adjective humongous. MongoDB as a Document Databases guarantees consistency, atonic transaction, availability, various queries, and extensibility of document data. Figure 1 shows an example of assigning priority to each node. Figure 2 shows an example of inserting a new node mongo D into the existing duplicated set. Figure 3 shows an example of setting a shard for duplication of each Shard.

III. Application Plan of Document Databases for Big Data

The appropriate places of using Document Databases are event logging, contents management systems, blogging platform, Web analysis, real-time analysis, and applications for electronic commerce. But, Document Databases are not panacea for Big Data with complex transactions or queries for changing set structures.
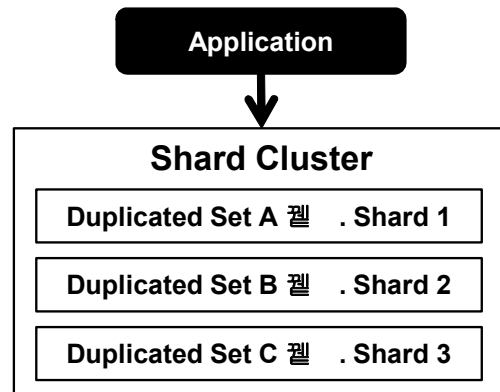
Application

Shard Cluster

Duplicated Set A 궬   . Shard 1

Duplicated Set B 궬   . Shard 2

Duplicated Set C 궬   . Shard 3

Figure 3. Setting of a Shard

IV. Conclusions

Document Database's document data model makes it easy to build on, since it supports unstructured data natively and doesn't require costly and time-consuming migrations when application requirements change. In addition, Document Databases supports rich queries and full indexes, distinguishing it from other document databases that make complex queries difficult or require a separate server layer to enable them. Its other features include automatic Sharding, replication, and more.

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.