A New Application for the Management of the MongoDB Servers

Tudorica Bogdan George, Department of Modelling, Economic Analysis and Statistics, Petroleum-Gas University of Ploiesti, Ploiesti Romania & The Bucharest University of Economic Studies, Bucharest, Romania

ABSTRACT

The application presented in the following subsections intends to cover one of the noticeable gaps of the NoSQL domain, namely the relative lack of working tools and systems administration for new large data storage systems. Following the comparative analysis of the NoSQL solutions on the market, the MongoDB system was chosen as target application for this step of development, for reasons mainly related to proven performance, flexibility, market presence already in place and ease of use.

Keywords: Binary JavaScript Object Notation (BSON), Data Storage Systems, Database Management Tools, Mongo Database (MongoDB), Not Only Structured Query Language (NoSQL) Databases

APPLICATION DESIGN: TARGETED FEATURES

MongoDB is a NoSQL database system which has the following characteristics (data presented below are taken from a recent paper of the author):

- **MongoDB is a fast and scalable database:** The data is stored as documents, which allows the representation of the complex relationships in a single article. Documents can be composed of individual fields with fundamental data types, included documents or document vectors. This flexibility allows a developer to model a wide range of problems in a flexible manner without resorting to breaking up data in several tables. For cases where data cannot be modeled as a single document, MongoDB introduces the concept of DBRef, which is a pointer from a field in a document to another document;

- **Extracting and querying data in MongoDB is flexible:** The documents can be queried dynamically based on the main document, on any field in the document, on any included document, or any document included in a document vector.
Written in C++, MongoDB features the following:

- Storage-oriented documents (using the power and flexibility of the JSON type data schemes);
- Included objects, embedded vectors, geospatial information;
- Dynamic Queries;
- Support for indexes, including secondary indexes;
- Fast updates of data;
- Efficient storage of large binary objects (photos, videos);
- Support for replication and system failure;
- Auto-sharding (auto-partitioning) for scalability;
- MapReduce for complex aggregation;
- Commercial support, consulting and training.

MongoDB is licensed under a GNU license; the source code can be compiled on any operating system. It can be installed as an executable binary Linux, MacOS X, Windows and Solaris. The system runs as a daemon process mongod - core database server that can be accessed by drivers.

Partitioning and routing are performed via the mongos service. Drivers for server access are available for a variety of programming languages currently supported: C, C++, C#, &. NET, ColdFusion, Erlang, Factor, Java, Javascript, PHP, Python, Ruby, Perl.

MongoDB can be run in two modes depending on the application needs. The first way is “single master” where there is only one master server for all write operations. Read operations can be executed on the master server or client on any number of servers (slave).

For applications where the amount of data is large and the frequency of writing operations is high an auto-partition (auto-sharding) mode is used. In this mode, write operations are distributed automatically between partitions (shard - a database server or a database server group), each executing operations to read/write a piece of data.

Accessibility is achieved by replicating data across multiple MongoDB nodes, any of them being able to take the role of master at a time.

MongoDB also provides features that go beyond basic operations for databases. For example, it provides a good solution to store small and large files together in the database. Files are automatically divided into pieces. If MongoDB is running in a self-sharded environment, pieces of files are also replicated on multiple servers.

Storing files is a very difficult problem to solve effectively, especially when managing a large number of files. Saving files to the local system is seldom a good solution. MongoDB solves this problem by creating two internal collections: a collection of images that retains information about files metadata and a collection of partitions that store information about file partitions.

MongoDB allows for Map-Reduce operations. Map-Reduce are operations for handling large sets of information. The “map” operation applies a function to all documents and gets a new set of key-value pairs. The “reduce” operation takes the results of the map operation and applies other functions returning a single result per key. At the end of the “map” process, the result is sorted and grouped by key values. For each resulting key the “reduce” function is called with two arguments: a key and a vector with all values.

The programming language used on this project is C Sharp, a relatively new language, flexible, offered as a component of the Microsoft Visual Studio application development environment. The application was developed in the 2008 version of the C Sharp environment.

The actual connection to the database server was made using an open source driver developed by a third party.

The effective design of the application consists of the design of the interface and that of the used data processing.

The interface is multi-window one, relatively simple in terms of graphics and it’s intended to provide maximum intuitively and ease of use. The target group of users consid-
Related Content

[www.igi-global.com/article/measuring-democracy-quality-democracy-world/77344?camid=4v1a](www.igi-global.com/article/measuring-democracy-quality-democracy-world/77344?camid=4v1a)

Proposing a Supply Chain Model for the Production-Distribution of Fuelwood in Greece using Multiobjective Programming
[www.igi-global.com/chapter/proposing-a-supply-chain-model-for-the-production-distribution-of-fuelwood-in-greece-using-multiobjective-programming/82857?camid=4v1a](www.igi-global.com/chapter/proposing-a-supply-chain-model-for-the-production-distribution-of-fuelwood-in-greece-using-multiobjective-programming/82857?camid=4v1a)
Bank Credit Applications and Advancement Recommendations in Regards to Sustainable Buildings in Turkey
[www.igi-global.com/chapter/bank-credit-applications-and-advancement-recommendations-in-regards-to-sustainable-buildings-in-turkey/157897?camid=4v1a](www.igi-global.com/chapter/bank-credit-applications-and-advancement-recommendations-in-regards-to-sustainable-buildings-in-turkey/157897?camid=4v1a)

India's Looming Power Crisis and the Way Forward: An Ode to Sustainability
[www.igi-global.com/article/indias-looming-power-crisis-and-the-way-forward/181023?camid=4v1a](www.igi-global.com/article/indias-looming-power-crisis-and-the-way-forward/181023?camid=4v1a)