АНАЛИЗ NOSQL БАЗ ДАННЫХ НА ОСНОВЕ ПРОИЗВОДИТЕЛЬНОСТИ ПРИ ВЫСОКИХ РАБОЧИХ НАГРУЗКАХ В СИСТЕМАХ ДИСТАНЦИОННОГО ОБУЧЕНИИЯ

А.А. Гришкевич, А.В. Михайловская

Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь, andrew.grichkevitch@gmail.com, alexandra.mikh@gmail.com

Abstract. Relational databases are a technology used universally that enables storage, management and retrieval of varied data schemas. However, execution of requests can become a lengthy and inefficient process for some large databases. The purpose of this paper is to compare different NoSQL databases, to evaluate their performance according to the typical use for storing and retrieving data. We tested NoSQL databases to better understand how performance is affected by each database type and their internal mechanisms.

В 2010 году, когда мир был впечатлен возможностями облачных систем и новых баз данных, разработанных с целью их обслуживания, группа исследователей из Yahoo решила изучить NoSQL. Они разработали YCSB фреймворк, чтобы иметь возможность оценить производительность новых инструментов и найти лучшие варианты их использования.

В нашей работе мы провели независимый и объективный сравнительный анализ следующих NoSQL баз данных:

- ***Cassandra***: хранилище на основе семейства столбцов;
- ***HBase***: хранилище на основе семейства столбцов;
- ***MongoDB***: документо-ориентированная база данных;
- ***Riak***: хранилище типа «ключ-значение».

Мы также протестировали ***MySQL Cluster*** и ***Sharded MySQL***, принимая их в качестве контрольных показателей.

В качестве основы для анализа используется Yahoo Cloud Serving Benchmark (YCSB), включающий в себя фреймворк для генерации нагрузок и набор сценариев нагрузки БД.

Нагрузочное тестирование включало в себя следующий набор операций:

- *Read*: чтение записи;
- *Insert*: создание новой записи;
- *Update*: редактирование существующей записи путем изменения значения одного из полей.

Каждая база данных, участвующая в анализе, включает в себя 100.000.000 записей, каждая из которых занимает 1000 байт и содержит 10 полей со строкой типа “user234123” в качестве уникального ключа.

Производительность базы данных была определена как скорость, с которой база данных производит базовые операции. В данном контексте, базовая операция -- действие, выполняемое исполнителем нагрузки (workload executor), который одновременно управляет несколькими клиентскими потоками. Каждый поток выполняет последовательный ряд операций, которые представляют собой вызов методов интерфейса базы данных для загрузки данных в базу (фаза загрузки) и исполнение поставленных задач (фаза транзакций). Потоки ограничивают скорость генерации запросов, что позволяет непосредственно контролировать нагрузку на базу.

В первую очередь была проанализирована производительность загрузки данных. В данном случае лидером стала HBase, достигшая скорости 40 тыс. оп/сек. Cassandra также показала хорошую производительность около 15 тыс оп/сек (см. рис. 1).

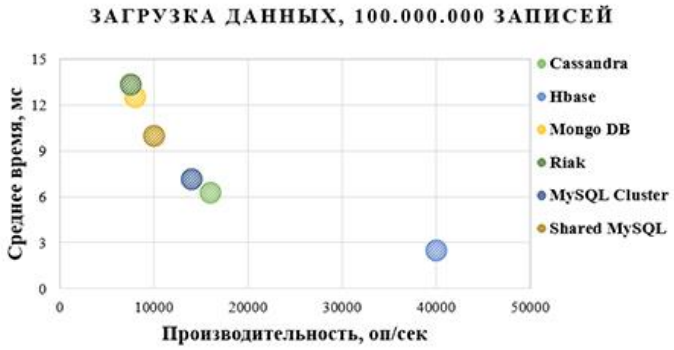

Рисунок 1 – Производительность при загрузке данных

Сценарий 1: частое обновление - Во время обновлений HBase и Cassandra ушли далеко вперед относительно остальных со средним временем ответа не превышающим 2 миллисекунды.

Сценарий 2: частое чтение - В данном случае Sharded MySQL показал лучший результат. Результат MongoDB был близок благодаря документо-ориентированному подходу.

Сценарий 3: только чтение - Благодаря индексам на основе бинарного дерева, Shared MySQL выходит победителем со скоростью более чем 3 тыс. оп/сек. MongoDB, Cassandra и HBase демострируют примерно одинаковые результаты на уровне 2 тыс. оп/сек.

Сценарий 4: частая запись - В данном случае HBase показала лучший результат под нагрузкой, включающей в себя большое количество вставок. Cassandra оказалась на втором месте, после нее MySQL Clsuter и Riak примерно на одном уровне.

NoSQL базы данных обещают хорошую производительность и масштабируемость для простых операций над большими объемами данных. В данной работе мы проанализировали производительность некоторых из наиболее перспективных примеров, проверяя их характеристики при чтении, записи и смешанной нагрузке.

Каждая база данных имеет свои преимущества и недостатки, которые становятся более или менее значимымы в зависимости от конкретных предпочтений и в особенности от сценариев использования.

Хранилище данных может иметь превосходную производительность, но при увеличении количества записей до какого-то определенного уровня, скорость выполнения операций сильно падает. Это может означать, что такая база данных подойдет для задач с приемущественно операциями редактирования или высокоскоростных выборок, а не с чтением/записью.

Литература

1. Benchmarking Cloud Serving Systems with YCSB [Электронный ресурс] / Yahoo! Research. – Santa Clara, CA, USA, 2010. – Режим доступа: https://www.cs.duke.edu/courses/fall13/compsci590.4/838-CloudPapers/ycsb.pdf. – Дата доступа: 01.09.2015.

2. The Apache Cassandra Project [Электронный ресурс] / The Apache Software Foundation. – Режим доступа: http://cassandra.apache.org/. – Дата доступа: 01.09.2015.

3. Apache HBase [Электронный ресурс] / The Apache Software Foundation. – Режим доступа: http://hbase.apache.org/. – Дата доступа: 01.09.2015.

4. MongoDB [Электронный ресурс] / MongoDB, Inc. – Режим доступа: https://www.mongodb.org/. – Дата доступа: 01.09.2015.

5. Riak for Big Data Application Products [Электронный ресурс] / Basho Technologies, Inc. – Режим доступа: http://basho.com/products/. – Дата доступа: 01.09.2015.