

# АНАЛИЗ NOSQL БАЗ ДАННЫХ НА ОСНОВЕ ПРОИЗВОДИТЕЛЬНОСТИ ПРИ РАЗЛИЧНЫХ РАБОЧИХ НАГРУЗКАХ СИСТЕМЫ

А. А. Гришкевич, А. В. Михайловская

Кафедра информатики, Факультет компьютерных систем и сетей, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: andrey.grichkevitch@informatics.by, alexandra.mikh@gmail.com

На сегодняшний день, повсеместно используемой технологией для хранения и управления данными являются реляционные базы данных. Тем не менее, выполнение запросов может быть длительным и неэффективным процессом, когда базы достигают больших размеров. С другой стороны, технологии нереляционных баз данных, также известные как NoSQL, были разработаны для более полного удовлетворения потребностей хранения больших объемов данных по принципу ключ-значение. На сегодняшний день существует большое количество нереляционных баз данных, сравнительный анализ которых еще не производился. Целью данной работы является анализ различных NoSQL баз данных, чтобы оценить их производительность в соответствии с наиболее типичными сценариями использования при хранении и извлечении данных. В ходе анализа были использованы смешанные сценарии нагрузочного тестирования, чтобы с наибольшей точностью определить, как различные операции влияют на производительность.

## ВВЕДЕНИЕ

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.

Уже на 2014 год число нереляционных баз данных превысило 120, и их количество продолжает расти. Это разнообразие делает выбор наилучшего инструмента для каждого конкретного случая необычайно сложным.

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

- **Cassandra**: хранилище на основе семейств столбцов;
- **HBase**: хранилище на основе семейств 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.

## I. ОКРУЖЕНИЕ

В качестве основы для анализа используется Yahoo Cloud Serving Benchmark (YCSB), включающий в себя фреймворк для генерации нагрузок и набор сценариев нагрузки БД (см. рис. 1). 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 определяется скоростью, с которой база выполняет базовые операции. Каждая из баз была развернута на 4-ядерном кластере Amazon Extra Large Instances.

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.

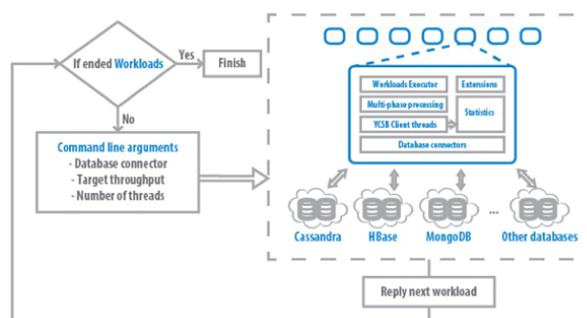


Рис. 1 – Диаграмма компонентов YCSB

## II. РЕЗУЛЬТАТЫ

В первую очередь была проанализирована производительность загрузки данных, которая представляла из себя запись в базу 100 миллионов записей, с 10 полями, состоящими из 100 случайных ASCII символов. В данном случае лидером стала HBase, достигшая скорости 40 тыс. оп/сек. Cassandra также показала хорошую производительность около 15 тыс оп/сек (см. рис. 2).



Рис. 2 – Производительность при загрузке данных

**Сценарий 1: частое обновление.** Данный сценарий моделирует работу базы данных, в ходе которой записываются типичные действия пользователя во время работы с приложениями электронной коммерции. При работе данного сценария используется соотношение чтения/обновления – 50/50.

Во время обновлений HBase и Cassandra ушли далеко вперед относительно остальных со средним временем ответа не превышающим 2 миллисекунды. HBase оказалась даже быстрее этого. У HBase-клиента был выключен механизм AutoFlush, так что изменения, накопившиеся в буфере клиента и ждущие записи, писались асинхронно, по мере того, как буфер заполнялся. Cassandra в свою очередь записывает изменения в журнал транзакций, после чего таблицы в памяти обновляются. Это работает медленнее, но куда более безопасно, нежели механизм HBase.

**Сценарий 2: частое чтение.** Данный сценарий на 95 процентов состоит из операций чтения, и только на 5 из операций записи. В качестве примера данного сценария можно привести тегирование данных: добавление нового тега это операция обновления, но большинство операций подразумевают чтения тегов.

В данном случае Sharded MySQL показал лучший результат. Результат MongoDB был близок благодаря документо-ориентированному подходу.

**Сценарий 3: только чтение.** Данный сценарий моделирует систему кэширования данных. В данном случае информация хранится вне системы, а приложение имеет возможность только читать её, но не вносить изменения. Соответственно используется соотношение чтения/обновления – 100/0.

Благодаря индексам на основе бинарного дерева, Sharded MySQL выходит победителем со

скоростью более чем 3 тыс. оп/сек. MongoDB, Cassandra и HBase демонстрируют примерно одинаковые результаты на уровне 2 тыс. оп/сек.

**Сценарий 4: частая запись.** Данный сценарий направлен на анализ производительности хранилищ данных при нагрузке в основном состоящей из операций вставки, который включает в себя соотношение вставки/чтения – 90/10.

В данном случае HBase показала лучший результат под нагрузкой, включающей в себя большое количество вставок. Cassandra оказалась на втором месте, после нее MySQL Cluster и Riak примерно на одном уровне.

## ЗАКЛЮЧЕНИЕ

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 значимыми 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 преимущественно операциями редактирования или высокоскоростных выборов, но не с чтением/записью.

1. Benchmarking Cloud Serving Systems with YCSB [Electronic resource] / Yahoo! Research. – Santa Clara, CA, USA, 2010. – Mode of access: <https://www.cs.duke.edu/courses/fall13/compsci590.4/838-CloudPapers/ycsb.pdf>. – Date of access: 01.08.2015.
2. NoSQL Benchmark Compares Aerospike, Cassandra, Couchbase and MongoDB [Electronic resource] / A. Avram – InfoQ, 2013. – Mode of access: <http://www.infoq.com/news/2013/04/NoSQL-Benchmark>. – Date of access: 10.08.2015.
3. Abramova, V. Experimental Evaluation of NoSQL Databases / V. Abramova, J. Bernardino, P. Furtado // International Journal of Database Management Systems. – 2014. – Vol. 6, № .3
4. The Apache Cassandra Project [Electronic resource] // The Apache Software Foundation. – Mode of access: <http://cassandra.apache.org/>. – Date of access: 29.08.2015.
5. Apache HBase [Electronic resource] // The Apache Software Foundation. – Mode of access: <http://hbase.apache.org/>. – Date of access: 30.08.2015.
6. MongoDB [Electronic resource] // MongoDB, Inc. – Mode of access: <https://www.mongodb.org/>. – Date of access: 31.08.2015.
7. Riak for Big Data Application Products [Electronic resource] // Basho Technologies, Inc. – Mode of access: <http://basho.com/products/>. – Date of access: 31.08.2015.