Хранение данных в системах автоматизированного мониторинга

Урбан А.В.
Кафедра систем управления
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
avurban@gmail.com

***Аннотация*—Этот доклад содержит анализ применения документо-ориентированных баз данных в качестве хранилища данных в системах автоматизированного мониторинга. Рассмотрены преимущества и недостатки использования документо-ориентированных баз данных по сравнению с реляционными СУБД в системах с большим количеством наблюдателей и интенсивным изменением состояния объектов. Приведены способы и конфигурации использования документо-ориентированных баз данных в системах управления.**

Ключевые слова: хранение информации, базы данных, документо-ориентированная база данных, производительность

I. Введение

Во многих информационных системах, предоставляющих возможность наблюдения и изменения данных, важным параметром качества работы является производительность. Это понятие включает в себя время отклика системы на запросы изменения и получения данных, затраты аппаратных ресурсов на обслуживание запросов. Существуют различные способы оптимизации и повышения производительности систем. Одним из таких способов является применение документо-ориентированных баз данных вместо или совместно с реляционными СУБД. В данной статье рассмотрены основные концепции документо-ориентированных баз данных, приведены основные существующие решения СУБД в данной области, произведен анализ применения данных хранилищ информации.

II. Документо-ориентированные базы данных

Документо-ориентированных базы данных служат для получения, изменения и управления информацией имеющей гибкую структуру. Данные представлены в виде документов имеющих иерархическую структуру. Таким образом, хранимая информация избыточна и не является нормализованной.

Данный тип баз данных входит в категорию так называемых NoSQL СУБД. К основным концепциями NoSQL можно отнести нереляционную модель данных, открытый исходный код, хорошую горизонтальную масштабируемость. Синтаксис запросов и формат хранения данных в СУБД зависит от реализации ядра.

В данный момент к наиболее популярным СУБД рассматриваемого типа можно отнести следующие: Cassandra, Mongodb, CouchDB, Redis, Riak, Membase, Neo4j и HBase. К основным параметрам сравнения различных реализаций СУБД можно отнести следующие: наличие транзакций, возможность и реализация масштабирования, ограничения по размеру хранилища данных, количество используемых аппаратных ресурсов и реализация способа хранения данных. Представленные СУБД имеют свою область применения исходя из специфики работы ядра, достоинств и недостатков. Выбор той или иной СУБД зависит от особенностей разрабатываемой системы и предъявляемых требований к хранению данных.

III. Применение документо-ориентированных баз данных в системах автоматизированного мониторинга

Данный тип СУБД не является универсальным решением, подходящим для всех типов систем, однако при правильном проектировании может значительно повысить производительность. Также, не всегда оправдано использование NoSQL СУБД вместо реляционных баз данных.

При использовании документо-ориентированных баз данных существует ряд особенностей, которые необходимо учитывать в стадии выбора используемых технологий, разработки архитектуры системы и программной реализации проекта.

В качестве примера документо-ориентированной СУБД рассмотрим MongoDB. В качестве системы – комплекс, предоставляющий информацию большому числу потребителей и интенсивно обновляющий наблюдаемые данные. Данные являются сложными объектами, отображающими совокупность состояний различных компонентов системы. Рассмотрим процессы хранения, чтения, записи и обработки информации в данной СУБД.

Хранимые данные являются избыточными и представлены в виде иерархического документа. В случае реляционных СУБД для получения аналогичного объекта необходимо производить объединение таблиц. Таким образом, использование документов позволяет значительно сократить время поиска информации в базе для данной сложной структуры. Документо-ориентированный подход полезен, когда данные об объектах наблюдения имеют динамическую или гибкую структуру. Однако наличие избыточности приводит к необходимости следить за целостностью данных в базе, что необходимо учитывать при проектировании системы. Процесс слежения за целостностью данных может требовать выполнения большого количества операций по поиску и обновлению документов. Запуск данного процесса целесообразно производить в фоновом режиме для сохранения приемлемого времени отклика системы на действие, что может привести к задержкам между запросом изменения данных и их фактическим

обновлением в базе для всех соответствующих документов.

Сравнение времени чтения данных для 600 сложных объектов представлено на «Рис. 1».

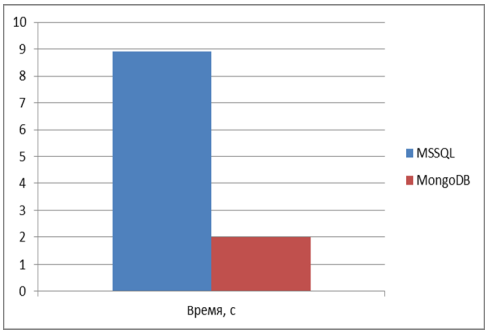

Рис. 1. Сравнение скорости поика данных

MongoDB предоставляет возможность безопасной или неконтролируемой вставки данных. В случае отсутствия контроля над безопасностью операции добавления данных, будет получено значительное ускорение данного действия, однако нет гарантии, что все операции пройдут успешно. Выбор того или иного режима зависит от конкретной системы и ее приоритетов. Особенностью данной СУБД является то, что операция обновления существующих документов и массивов происходит значительно быстрее, чем создание новых. Это обусловлено реализаций механизма работы с памятью диска. Таким образом, для получения повышения производительности при записи в базу данных необходимо проектировать систему с минимальной частой добавление новых документов и полей, производя запись с помощью обновления уже созданных документов и массивов. Также, MongoDB предоставляет атомарную команду, которая обновит существующий документ, соответствующий заданным условиям, а в случае отсутствия подходящего документа будет создан новый. Использование данной команды исключает необходимость отдельного запроса поиска документа и является потокобезопасной.

Сравнение времени исполнения вставки данных для различных режимов MongoDB, а также для MSSQL представлены на «Рис. 2».

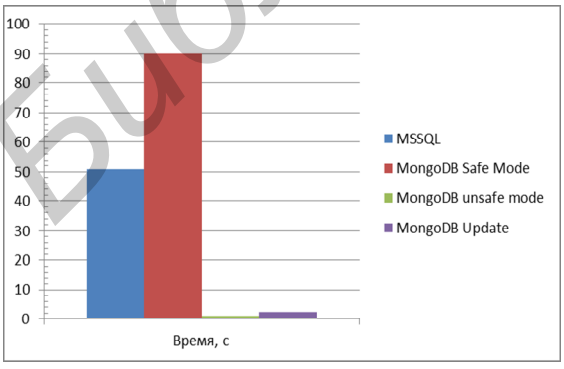

Рис. 2. Сравнение скорости операций записи

MongoDB не имеет встроенной системы проведения транзакций, что также накладывает ограничения на сферу ее применения. Однако MongoDB поддерживает механизм compare-and-set. Это механизм, который гарантирует отказ в изменении объекта, если с момента последнего чтения объект был изменен другим клиентом [4].

Также в данной СУБД отсутствуют встроенные агрегирующие функции. Для осуществления операций по агрегации данных предусмотрены специальные map/reduce функции на языке JavaScript. Выполнение данных функций не является высокопроизводительным, однако СУБД предоставляет возможность сохранять рассчитанные значения в специальные коллекции. Использование данных коллекций позволяет при соответствующей организации системы

Таким образом, помимо прироста производительности, использование документо-ориентированных СУБД накладывает условия на архитектуру системы. Использование рассмотренной СУБД в качестве основного хранилища данных целесообразно в системах, в которых наблюдение происходит за ограниченным числом сложных объектов, изменение состояния этих объектов происходит интенсивно и требования к системе позволяют реализовать архитектуру, удовлетворяющую особенностям работы с названной базой данных. Также некоторые процессы обработки данных системы могут производиться в фоновом режиме. В этом случае, возможно получить существенный прирост производительности, хорошую масштабируемость и гибкую систему хранения данных.

В случае, если требования к системе не позволяют использовать NoSQL СУБД, возможен гибридный подход. В качестве основного хранилища данных выступает реляционная СУБД, обеспечивающая проведение транзакций и целостность данных. В качестве промежуточного хранилища выступает NoSQL СУБД, содержащая состояние сложных объектов. Данных подход может являться альтернативой, либо частью подсистемы кеширования состояния объектов.

Таким образом, NoSQL СУБД не являются полноценной заменой реляционных баз данных. Они имеют ряд существенных особенностей, которые необходимо учитывать при выборе той либо иной технологии данных. Однако, при правильном использовании, данные СУБД способным значительно улучшить показатели производительности системы, упростить ее архитектуру за счет иной концепции в методах хранения и обработки данных.

[1] Kristina Chodorow, Michael Dirolf.MongoDB: The Definitive Guide/ O'Reilly Media, Inc., 2010 – pp.195.

[2] Kristina Chodorow. 50 Tips and Tricks for MongoDB Developers/ O'Reilly Media, Inc., 2011 – pp.53.

[3] Kristóf Kovács: Cassandra vs MongoDB vs CouchDB vs Redis vs Riak vs HBase vs Membase vs Neo4j comparison / Kristóf Kovács // kkovacs [Электронный ресурс]. - 2011. – Режим доступа : http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis. – Дата доступа 01.10.2012

[4] Транзакции в MongoDB // habrahabr [Электронный ресурс]. - 2012. – Режим доступа : http://habrahabr.ru/post/153321. – Дата доступа 06.10.2012

[5] NoSQL // wikipedia [Электронный ресурс]. - 2012. – Режим доступа : http://en.wikipedia.org/wiki/NoSQL. – Дата доступа 02.10.2012