



Соколов С. С., Новоселов Р. Ю., Митрофанова А. В.

МЕТОДЫ ОБЕСПЕЧЕНИЯ ДОСТУПНОСТИ ИНФОРМАЦИИ В ВЫСОКОНАГРУЖЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

В статье рассматривается проблема доступности информации в современном мире в высоконагруженных информационных системах. Рассмотрены основные понятия, такие как - распределенная вычислительная система, высоконагруженная информационная система, big data. Приведены критерии распределенной вычислительной системы. Также описаны основные методы управления нагрузкой и повышения отказоустойчивости высоконагруженных информационных систем, их преимущества и недостатки, в частности баз данных. Приводится пример реализации одного из методов.

Ключевые слова: высоконагруженные информационные системы, распределенные вычислительные системы, базы данных, доступность информации, методы оптимизации нагрузки.

Sokolov S. S., Novoselov R. Y., Mitrofanova A. V.

METHODS TO ENSURE THE AVAILABILITY OF INFORMATION IN HIGHLOAD INFORMATION SYSTEM

The article analyses problem of information availability in modern world in highload information systems. Described main concepts, such as distributed computer system, highload information system, big data. Gave criteria of distributed computer system. Also considered main methods of load managing and increasing resilience of highload systems, its advantages and disadvantages, particularly databases. Gave an example of realization one of methods.

Keywords: highload information system, distributed computer system, databases, information availability, methods of load managing.

Введение

В современном мире количество информации неуклонно растет. Существует проблема, которая называется “информационный взрыв” - постоянное увеличение скорости и объемов публикаций (объёма информации) в масштабах планеты. Как известно, главными свойствами данных является конфиденциальность, целостность и доступность [1]. В силу глобализации мира, развития интернета, повышения требований к скорости, появляется множество задач, связанных с обеспечением доступности информации. Современные информационные системы в качестве основной характеристики имеют распределенную архитектуру, но даже распределенная система не спасает от нагрузки, как следствие, от проблем доступа, а соответственно, не может обеспечить часть или все требования к информации. Существуют различные методы обеспечения доступности. В данной статье будет дано определение высоконагруженной информационной системе, а также будут рассмотрены способы оптимизации нагрузки.

Определение понятий и постановка задачи

Для обсуждения данной проблемы нужно дать определение высоконагруженной ИС, но сначала поговорим о распределенной вычислительной системе.

Эндрю Таненбаум, в своем фундаментальном труде «Распределенные системы. Принципы и парадигмы» [2] предложил следующее определение:

«Распределенная вычислительная система – это набор соединенных каналами связи независимых компьютеров, которые с точки зрения пользователя некоторого программного обеспечения выглядят единым целым».

Выделяют следующие важные характеристики РВС:

- возможность работы с различными типами устройств: с различными поставщиками устройств, с различными операционными системами, с различными аппаратными платформами;

- возможность простого расширения и масштабирования;

- перманентная (постоянная) доступность ресурсов (даже если некоторые элементы РВС некоторое время могут находиться вне доступа);

- сокрытие особенностей коммуникации от пользователей.

Вычислительные среды, состоящие из множества вычислительных систем на базе разных программно-аппаратных платформ, называются *гетерогенными*.

Для того чтобы РВС могла быть представлена пользователю как единая система, применяют следующие типы прозрачности в РВС:

- *прозрачный доступ к ресурсам* – от пользователей должна быть скрыта разница в представлении данных и в способах доступа к ресурсам РВС;

- *прозрачное местоположение ресурсов* – место физического расположения требуемого ресурса должно быть несущественно для пользователя;

- *репликация* – сокрытие от пользователя того, что в реальности существует более одной копии используемых ресурсов;

- *параллельный доступ* – возможность совместного (одновременного) использования одного и того же ресурса различными пользователями независимо друг от друга. При этом факт совместного использования ресурса должен оставаться скрытым от пользователя;

- *прозрачность отказов* – отказ (отключение) каких-либо ресурсов РВС не должен оказывать влияния на работу пользователя и его приложения.

Как мы видим, многие описанные выше тезисы характерны и для систем, которые мы называем высоконагруженными. Но чтобы выяснить, что из этого является классом и подклассом, мы попробуем дать определение высоконагруженной ИС.

Высоконагруженная ВС - это распределенная ВС, к которой при проектировании и разработке применяются соответствующие техники не только распределенных систем, а также другие, например: определенные шаблоны при проектировании базы данных, алгоритмика и др.

Также понятие высоконагруженной ВС может быть (но не обязательно) связано с понятием больших данных (*big data*). Не будем углубляться в большие данные, лишь дадим определение и примеры использования, чтобы читатель мог понять, как связана тема статьи и *big data*.

Большие данные (англ. *big data*) — серия подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объемов и значительного многообразия для получения воспринимаемых человеком результатов, эф-

фективных в условиях непрерывного роста, распределения по многочисленным узлам вычислительной сети, сформировавшихся в конце 2000-х годов, альтернативных традиционным системам управления базами данных и решениям класса Business Intelligence.

То есть, как ни странно **big data** - это не объем данных, а методы работы с ними.

Примеры задач, в которых используется big data:

- Логи поведения пользователей в интернете.
- GPS-сигналы от автомобилей для транспортной компании.
- Данные, снимаемые с датчиков в большом адронном коллайдере.
- Оцифрованные книги в Российской Государственной Библиотеке.
- Информация о транзакциях всех клиентов банка.
- Информация о всех покупках в крупной ритейл сети и т.д.

Такие данные скорее всего хранятся в базе данных, или в простых текстовых файлах, что реже (напр. логи поведения пользователей в интернете). В рамках данной статьи под высоконагруженной системой мы будем рассматривать web-приложение.

Таким образом, в рамках статьи мы будем решать задачу снижения нагрузки на информационную систему, и в первую очередь на базу данных.

Способы решения задачи

В предыдущей работе [3] было доказано, что скорость работы базы данных напрямую зависит от количества данных, содержащихся в ней. Поэтому скорее всего первое, что будет тормозить систему, это база данных. Необходимо выделить ее на отдельный сервер, подобные действия позволят увеличить ее производительность и снизить ее негативное влияние на остальные компоненты (вычислительные ресурсы, веб-сервер и др.). Далее, выносим веб-сервер на отдельный узел. Освободим больше ресурсов для вычисления. После можно разделить вычисления на несколько серверов, подключить кэширование, очереди, балансировщики. Но обычно самым тяжелым элементом системы является база данных, поэтому оптимизации данного узла необходимо уделить гораздо больше внимания. Масштабирование баз данных - это одна из самых сложных задач во время роста ИС. Существует очень много практик — денорма-

лизация, репликации, шардинг и многие другие. Рассмотрим основные:

Денормализация данных

Существует так называемая нормальная форма хранения данных, которая предполагает избегания дублирования данных. Ключевых правила два:

- **Атомарность** означает, что все сущности хранятся в неделимом виде. Например, если мы храним адрес, то он скорее всего будет поделен на название города, страны и улицу. Все они должны быть представлены отдельными таблицами. Название города будет атомарным, т.к. дальше делиться не будет.
- **Уникальность** требует, чтобы каждая сущность была определена только один раз. Например, название города с идентификатором 1 должно присутствовать только в таблице *cities*.

Денормализация — это постепенный процесс избавления от правил нормализации там, где это необходимо. Обычно это случаи, в которых есть частые повторные запросы к логически связанным данным [4].

Простым примером является хранение названия города вместе с данными пользователя. Если денормализовать эти данные, то можно будет получить название города одним SQL запросом, а не двумя. Далее в рамках статьи мы рассмотрим пример денормализации данных.

Репликация данных

Репликация — одна из техник масштабирования баз данных. Состоит эта техника в том, что данные с одного сервера базы данных постоянно копируются (реплицируются) на один или несколько других (называемые репликами). Таким образом появляется возможность распределить нагрузку с одного сервера на несколько.

Существует два основных подхода при работе с репликацией данных:

- Репликация Master-Slave;
- Репликация Master-Master [5].

Помимо повышения производительности, есть несколько причин для использования репликаций:

- Отказоустойчивость - если Slave сервер откажет, можно без проблем перевести все запросы на чтение на другие сервера. Если откажет Master, можно перевести запросы записи на реплики, и когда Master восстановит свою работу, он сможет перенять на себя роль Slave сервера;
- Резервирование данных - на какое-то

время в случае, например, оптимизации таблицы, можно остановить сервер и ничего страшного не случится;

– Отложенные вычисления - если в очереди есть сложный запрос, для его выполнения можно использовать Slave сервер, чтобы не замораживать работу всей системы.

Также, в силу того, что на Slave сервера передаются не сами данные, а запросы на их изменения, можно использовать разные схемы, типы таблиц или индексы.

Шардинг

Шардинг — это принцип масштабирования базы данных, когда данные разделяются по разным серверам. В нашем распоряжении есть два подхода:

Вертикальный шардинг - это простое распределение таблиц по серверам. Например, вы помещаете таблицу `users` на одном сервере, а таблицу `orders` на другом. В этом случае, группы таблиц, по которым выполняются JOIN, должны находиться на одном сервере.

Горизонтальный шардинг - разделение очень больших таблиц, которые перестают помещаться на одном сервере, на разные сервера. Это сильно усложняет логику приложения, однако на данный момент не существует лучших механизмов масштабирования [5].

Лучшим вариантом, по мнению автора статьи совмещение всех этих методов. Денормализация данных поможет избавиться от сложных JOIN запросов - соединение нескольких таблиц является очень ресурсоемкой операцией. Реплики, как было сказано выше, помогут распределить нагрузку с одного сервера на несколько. Ну и как самый эффективный метод, шардинг позволит получить действительно оптимизированную и быструю высоконагруженную систему.

Пример оптимизации базы данных

Для примера разберем некоторую социальную сеть. Мы заполнили таблицу тестовыми данными - 10000 пользователей и 5 городов. Примеры таблиц представлены ниже. Время выполнения запросов измерялось с помощью встроенного профайлера MySQL.

Предположим, что необходимо выбрать пользователя с ID 6, включая город пользователя. С учетом текущей схемы базы данных необходимо использовать JOIN. Запрос будет выглядеть следующим образом:

```
SELECT *, `name` AS `city_name` FROM `users` LEFT JOIN `cities` ON `users`.`city_id`=`cities`.`id` WHERE `users`.`id`=6
```

Часть таблицы `users`

ID	firstname	lastname	city_id
1	Jonh	Smith	2
2	Anthony	Park	3
3	Richard	Edwards	1
4	Reynard	McKinney	2
5	Charles	Turner	1

Таблица 2

Часть таблицы `cities`

ID	name
1	London
2	New-York
3	Paris

Данный запрос занимает 0.000502 секунды. Теперь попробуем денормализовать данные: добавим столбец `'city_name'` в таблицу `'users'`, куда запишем название города.

Добавляем поле:

```
ALTER TABLE `users` ADD COLUMN `city_name` VARCHAR(100);
```

Теперь для того, чтобы получить информацию о пользователе, включая его город, необходимо выполнить простой запрос:

```
SELECT * FROM `users` WHERE `id`=6;
```

Данный запрос занимает 0.000398 секунды. Соответственно, прирост производительности более чем на 20%. А в случае с большим объемом данных [3], результат будет гораздо лучше.

Заключение

Задача с доступностью данных с ростом ИТ становится все актуальнее, и как показано в статье. Современный мир отличается глобализацией, когда информация и люди, которым она требуется, находятся в разных уголках земного шара. С помощью методов, описанных в данной статье, возможно обеспечить информацию необходимым набором свойств.

Самым простым способом оптимизации является денормализация данных. Однако, такое решение конечно - невозможно денормализовать данные бесконечно. Репликация и шардинг являются более востребованными способами оптимизации, к тому же могут применяться вместе. Но стоит отметить, что ни один из представленных способов не является панацеей, и на этапе проектирования системы необходимо использовать все способы в комплексе, для достижения наилучшего результата.

Литература

1. ГОСТ Р 53114-2008. Обеспечение информационной безопасности в организации. Основные термины и определения // Защита информации. Москва: Стандартинформ, 2008.
2. Таненбаум Э. Распределенные системы. Принципы и парадигмы. - СПб: Питер, 2003.
3. Новоселов Р.Ю., Соколов С.С. Тестирование и анализ скорости двух популярных реляционных баз данных при различных нагрузках // Материалы работы науч.-исслед. конф. студентов и аспирантов ф-та информационных технологий. «IT: ВЧЕРА, СЕГОДНЯ, ЗАВТРА — 2016» 11 декабря 2015 г./ СПб.: Изд-во ГУМРФ им. Адмирала С. О. Макарова, 2016.
4. Денормализация данных [Электронный ресурс] // HighLoad++; URL: <https://ruhighload.com/post/Денормализация+данных> (дата обращения: 24.01.2018).
5. Репликация данных [Электронный ресурс] // HighLoad++; URL: <https://ruhighload.com/post/Репликация+данных> (дата обращения: 12.01.2018).
6. 5 стратегий работы с высокими нагрузками в MySQL [Электронный ресурс] // HighLoad++; URL: <https://ruhighload.com/post/4+стратегии+работы+с+высокими+нагрузками+в+MySQL> (дата обращения: 17.12.2017).

References

1. GOST R 53114-2008. Obespecheniye informatsionnoy bezopasnosti v organizatsii. Osnovnyye terminy i opredeleniya // Zashchita informatsii. Moskva: Standartinform, 2008.
2. Tanenbaum E. Raspredeleennyye sistemy. Printsipy i paradigmy. - SPb: Piter, 2003.
3. Novoselov R.YU., Sokolov S.S. Testirovaniye i analiz skorosti dvukh populyarnykh relyatsionnykh baz dannykh pri razlichnykh nagruzkakh // Materialy raboty nauch.-issled. konf. studentov i aspirantov f-ta informatsionnykh tekhnologiy. «IT: VCHERA, SEGODNYA, ZAVTRA — 2016» 11 dekabrya 2015 g./ Spb.: Izd-vo GUMRF im. Admirala S. O. Makarova, 2016.
4. Denormalizatsiya dannykh [Elektronnyy resurs] // HighLoad++; URL: <https://ruhighload.com/post/Denormalizatsiya+dannykh> (data obrashcheniya: 24.01.2018).
5. Replikatsiya dannykh [Elektronnyy resurs] // HighLoad++; URL: <https://ruhighload.com/post/Replikatsiya+dannykh> (data obrashcheniya: 12.01.2018).
6. 5 strategiy raboty s vysokimi nagruzkami v MySQL [Elektronnyy resurs] // HighLoad++; URL: <https://ruhighload.com/post/4+strategii+raboty+s+vysokimi+nagruzkami+v+MySQL> (data obrashcheniya: 17.12.2017).

СОКОЛОВ Сергей Сергеевич, заведующий кафедрой «Комплексное обеспечение информационной безопасности», Государственный университет Морского и речного флота имени адмирала С.О. Макарова, 198035, г. Санкт-Петербург, ул. Двинская, д.5/7. E-mail: sokolovss@gumrf.ru

НОВОСЕЛОВ Роман Юрьевич, аспирант, Государственный университет Морского и речного флота имени адмирала С.О. Макарова, 198035, г. Санкт-Петербург, ул. Двинская, д.5/7. E-mail: rnovoselov93@gmail.com.

МИТРОФАНОВА Анастасия Витальевна, аспирант, Государственный университет Морского и речного флота имени адмирала С.О. Макарова, 198035, г. Санкт-Петербург, ул. Двинская, д.5/7. E-mail: mitrofanovaav@gumrf.ru

SOKOLOV Sergey, chef of department “Complex providing information security”, Admiral Makarov State University of Maritime and Inland Shipping, 5/7, Dvinskaya str, Saint-Petersburg, 198035. E-mail: sokolovss@gumrf.ru

NOVOSELOV Roman, graduate student, Admiral Makarov State University of Maritime and Inland Shipping, 5/7, Dvinskaya str, Saint-Petersburg, 198035. E-mail: rnovoselov93@gmail.com

MITROFANOVA Anastasya, graduate student, Admiral Makarov State University of Maritime and Inland Shipping, 5/7, Dvinskaya str, Saint-Petersburg, 198035. E-mail: mitrofanovaav@gumrf.ru