

**ОБЕСПЕЧЕНИЕ ОТКАЗОУСТОЙЧИВОГО ДОСТУПА ВЕБ-ПРИЛОЖЕНИЙ
К ВЫСОКОНАГРУЖЕННЫМ СЕРВЕРАМ**

Е.А. Олисевич

jenia1702@gmail.com

SPIN-код: 2369-3990

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Описаны основные методы и решения при проектировании высоконагруженных программных приложений для освоивания теоретических и практических навыков работы с ними в связи с возникшей потребностью их применения во многих информационных сферах современного времени. Подробно рассмотрены основные технологии обеспечения доступа к высоконагруженным системам, различные варианты их реализации, методы оптимизации для обеспечения стабильной работы. На примере показаны и приведены наиболее эффективные решения для одной из самых используемых систем. Указаны основные способы применения технологий для основных составляющих частей программного приложения.

Ключевые слова

Отказоустойчивый доступ, кластеризация, оптимизация, высокая нагрузка, сервера, веб-приложение, репликация, шардинг, кластер, кеширование, база данных

Поступила в редакцию 08.04.2019

© МГТУ им. Н.Э. Баумана, 2019

Введение. В наши дни существует множество различных программных приложений, особенно в сети Интернет, каждое из которых постоянно нагружено в той или иной степени. В связи с этим появилась необходимость в поиске решений возникшей проблемы, что, в свою очередь, породило ряд технологий и методик, так как современные объемы получаемой информации любой организаций растут вместе с её ростом в её сфере, соответственно, этот рост влечет за собой увеличение количества цифровых данных, которые необходимо обрабатывать, что приводит к увеличению вычислительных мощностей. Но стоит отметить, что добавление вычислительных единиц далеко не самый бюджетный вариант решения проблемы, поэтому возникает необходимость оптимизации и распределения нагрузки на имеющиеся мощности.

Рассмотрим популярные предлагаемые на Российском рынке программные продукты, в которых осуществлены решения подобных задач.

Российская компания «1С» (<http://www.1c.ru/>) предлагает единую технологическую среду на платформе «1С», предназначенную для ускоренного создания различных программ. Стоит отметить, что компанией был разработан инструментарий для обеспечения отказоустойчивости в сфере ее деятельности: при необходимости пользователь может масштабировать ее до необходимых границ.

Российская транснациональная компания «Яндекс», владеющая системой поиска в сети и различными интернет порталами, предлагает сервис «Яндекс.Облако», содержащий необходимый инструментарий для создания веб-сайта с DNS-балансировкой нагрузки, а также оптимизации и других различных нагрузок.

На основе анализа предлагаемых продуктов на рынке принимается решение о необходимости освоения собственных методов обеспечения отказоустойчивости. Данное решение обеспечит понимание основных методов отказоустойчивости для создания необходимой структуры веб-приложения при минимальных аппаратных затратах и финансовой независимости от сторонних компаний.

Так как процесс обработки и хранения данных приложения включает в себя много трудоемких задач, то для освоения теоретической части таких методик оптимизации были приведены различные примеры их практических решений.

Применение серверных кластеров для распределения нагрузки высоконагруженных серверов. Говоря об обеспечении безотказного доступа к высоконагруженным серверам прибегают к различным методам оптимизации обмена данными, одним из этих методов являются технологии кластеров [1].

Кластер — обычно это несколько компьютеров, выполняющая одну общую функцию. Каждый такой сервер получил название «нода».

Выделяют следующие кластеры [2]:

- отказоустойчивые (при таком подходе группа компьютеров дублирует друг друга);
- балансировщики (распределяют запросы на сервере в случайном порядке);
- высокопроизводительные (кластер распределяет поступающие задачи на несколько серверов для увеличения скорости обработки данных, такой кластер распределяет поступающие задачи на несколько серверов для увеличения скорости обработки данных).

Для примера рассмотрим традиционное веб-приложение, поскольку на нем проще всего провести анализ. Оно работает на одном сервере, на этом сервере сразу выполняется несколько операций: вычисления веб-сервера, кэширование, работа с базой данных. Достаточно часто этих ресурсов не хватает, и приходится устанавливать более мощное «железо» — данный шаг называется «вертикальное масштабирование». Но настает момент, когда в «апгрейде железа» достигается лимит и масштабировать дальше нельзя.

Здесь без кластеров не обойтись. Задача сводится к тому, чтобы научиться представлять и разбивать узлы веб-приложения на кластеры, которые будут взаимозаменяемы и в случае недоступности одного из серверов приложение будет продолжать стабильно работать. Кластер решает сразу две проблемы: производительность и отказоустойчивость.

Решение данной задачи заключается в масштабировании баз данных, кэша, веб-серверов, при этом каждая составляющая помещается в определенную

группу серверов. Соответственно, сервера будут взаимозаменяемыми. Для достижения таких целей существуют соответствующие решения, а именно:

1) кластеризация веб-сервера (ее суть заключается в том, чтобы все сервера обрабатывали запросы синхронно, не теряя данных при взаимодействии друг с другом, также необходимо соблюдать баланс нагрузки на этих серверах, чтобы нагрузка распределялась равномерно [3]);

2) распределенный кэш данных (является формой кеширования, позволяющей кэшу работать с несколькими серверами [4]);

3) обеспечение непрерывных сессий (требуется для исключения вероятности потери данных, например, хранить сессии в базе данных);

4) репликация базы данных с помощью языка программирования PHP (постоянное дублирование базы данных на несколько серверов [5], т. е. появляется возможность получить доступ к базе данных с нескольких серверов) — рис. 1;



Рис. 1. Схема репликации [5]

5) шардинг базы данных (иная схема масштабирования, которая заключается в распределении базы данных на части, каждую из которых помещают на отдельный сервер. Существуют два типа шардинга: вертикальный и горизонтальный. В первом случае выделяют таблицы базы данных (БД) или группы таблиц БД, распределенных на сервер. В случае горизонтального шардинга одна таблица БД распределяется на разные сервера [4].

Анализ структуры базы данных. Вернемся к рассматриваемому примеру. С учетом вышесказанного следует разделить наше приложение на части, которые будут работать на разных серверах: веб-сервер, база данных, кэш — три сервера. Такая схема дает преимущество в производительности, т. е. веб-приложение сможет выдержать более высокую нагрузку, но это не обеспечит хорошую отказоустойчивость. В случае выхода из строя одного из серверов, оно перестанет работать, так как они не зарезервированы.

Одним из более простых подходов является репликация, которую можно реализовать практически на любой базе данных. Обычно в репликации фигурируют два типа серверов: Master и Slave. Master-сервер отвечает за изменение данных, а Slave — за чтение.

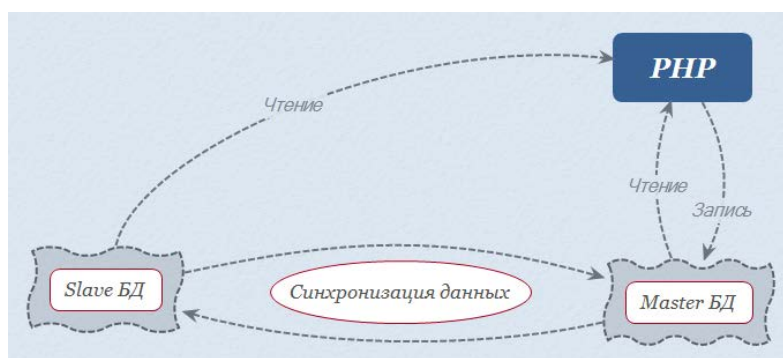


Рис. 2. Краткая схема Master — Slave [5]

Таким образом, все запросы на чтение/получение данных будет отправляться на Slave сервера, а на запись — на сервера Master. После организации подобной репликации нужно будет затронуть ядро веб-приложения для того, чтобы наладить организацию и управление этих процессов. Это даст следующие преимущества [6]:

- 1) возможность почти неограниченного масштабирования;
- 2) можно выделить отдельный сервер под резервное копирование, чтобы этот процесс не мешал основной работе приложения;
- 3) увеличение отказоустойчивости.

Таким образом, можно разбить базу данных на сервера и частично решить поставленную задачу.

Кластеризация веб-сервера. Добавляем несколько серверов (нод), изменяя логику веб-приложения, между серверами и запросами помещаем балансировщик, который может быть программным или в виде «железного» варианта, к примеру, от фирмы Cisco. Его задачей будет распределение запросов по серверам — нодам. Тем самым решается задача баланса нагрузки на сервера, но этого мало, так как нужно правильно распределить контент приложения между добавленными серверами: при добавлении любых данных пользователями, к примеру, картинок, необходимо, чтобы они были доступны немедленно и на других серверах.

Не стоит забывать и про пользовательские сессии: если пользователи авторизуются на одном сервере, то и другие сервера должны быть оповещены об этом, т. е. при подключении авторизованного пользователя на другой веб-сервер он должен считаться авторизованным, другими словами, необходимо организовать синхронизацию данных между серверами. Одним из подходящих решений является вынесение всех этих данных в отдельное централизованное хранилище на отдельном сервере с удобным доступом, разрешенным только для веб-приложения. Например, можно использовать FTP-сервер, а сеансы сессий хранить в базе данных. Резервировать файлы в таком случае можно с помощью подручных средств операционной системы или других служб.

При таком способе реализации получают ряд преимуществ. Повышается отказоустойчивость веб-приложения, также можно перемещать контент при-

ложения прямо во время работы, не прерывая его работу. То есть контент и веб-сервер разделены, а значит, возникает возможность отвязаться от поставщика хостинга.

Распределение кэша данных. Кэширование является неотъемлемой частью любого веб-приложения [5]. Так как во многих системах важно время отклика, а в цепи работы веб-приложения существуют запросы, которые выполняются отнюдь не быстро. Такие проблемы касаются даже высокопроизводительных систем, потому что сами ресурсы работают медленно. Решением такой задачи является кэширование (рис. 3).



Рис. 3. Примерная схема кэширования веб-приложения [7]

Суть кэширования заключается в том, что результаты вычислений помещаются в отдельное хранилище, которое сможет быстро обработать запросы и организовать доступ к этим данным. Обращение происходит не к серверам, а к кэшу. Соответственно, теперь нужно обеспечить отказоустойчивый доступ к кэшу.

Принцип работы с кэшем похож на принцип работы с базой данных. Сервера подключаются с учетом их масштабирования с учетом предъявляемых к ним требований [8].

Заключение. Существует много методик для решения поставленной задачи, в данной статье приведены лишь одни из основных и наиболее простых для понимания методов реализации, которые используются регулярно. В данной работе даны рекомендации для наилучшего и эффективного обеспечения отказоустойчивости программного приложения. Рассмотрены три основные части веб-приложения, а именно — веб-сервер, база данных, кэш и их разделение на дополнительные зарезервированные сервера, которые используются наиболее часто.

Проанализировав принципы работы высоконагруженных систем, можно сделать вывод, что далеко не каждая организация нуждается в проектировании таких систем, но со временем это становится необходимым мероприятием для обработки большего количества информации при минимальном расходе бюджета.

В связи с тем, что появился спрос по внедрению отказоустойчивых систем к высоконагруженным серверам, были проанализированы и рекомендованы варианты внедрения данных методик.

Литература

- [1] Муравьев К.А., Терехов В.В. Методы управления сетевым трафиком гетерогенных распределенных телекоммуникационных систем. *Проектирование и технология электронных средств*, 2017, № 2, с. 15–21.
- [2] Моск Д. Что такое кластер. *dmosk.ru: веб-сайт*. URL: <https://www.dmosk.ru/terminus.php?object=cluster> (дата обращения: 15.04.2019).
- [3] Хан И. Распределенное кэширование на пути к масштабированию. *msdn.microsoft.com: веб-сайт*. URL: <https://msdn.microsoft.com/ru-ru/magazine/dd942840.aspx> (дата обращения: 15.04.2019).
- [4] Карпунин А.А., Ганев Ю.М., Чернов М.М. Методы обеспечения качества при проектировании сложных программных систем. *Надежность и качество сложных систем*, 2015, № 2(10), с. 78–84.
- [5] Репликация данных. *ruhighload.com: веб-сайт*. URL: <https://ruhighload.com/%D0%A0%D0%B5%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F+%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85> (дата обращения: 15.04.2019).
- [6] Масштабирование баз данных — партиционирование, репликация и шардинг. *web-creator.ru: веб-сайт*. URL: https://web-creator.ru/articles/partitioning_replication_sharding (дата обращения: 15.04.2019).
- [7] Кэширование и memcached. *habr.com: веб-сайт*. URL: <https://habr.com/ru/post/42607/> (дата обращения: 15.04.2019).
- [8] Муравьев К.А., Терехов В.В. Анализ показателей качества систем передачи данных. *Тр. межд. симп. Надежность и качество*, 2018. т. 1, с. 128–133.
- [9] Аверьянихин А.Е., Котельницкий А.В., Муравьев К.А. Методика расчета оптимального числа узлов кластера виртуализации частного облака виртуальных рабочих столов по критерию эффективности. *Международный научно-исследовательский журнал*, 2016, № 5-3(47), с. 6–13.
- [10] Гуренко В.В., Бородин А.Ф., Назарков Д.А. Сравнительный анализ фреймворков для веб-разработки. *Технологии инженерных и информационных систем*, 2017, № 2, с. 3–14.
- [11] Грошев С.В., Пивоварова Н.В. Особенности проектирования СУБД для web-ориентированной системы оценки качества Парето-аппроксимации в задаче многокритериальной оптимизации. *Технологии инженерных и информационных систем*, 2015, № 2, с. 3–11.

Олисевиц Евгений Александрович — студент кафедры «Конструирование и технология электронной аппаратуры», МГТУ им Н.Э. Баумана, Москва, Российская Федерация.

Кондаков Никита Андреевич — инженер кафедры «Конструирование и технология электронной аппаратуры», МГТУ им Н.Э. Баумана, Москва, Российская Федерация.

ENSURING THE FAILURE-RESISTANT ACCESS OF WEB APPLICATIONS TO HIGHLY-LOAD SERVERS

E.A. Olisevich

jenia1702@gmail.com
SPIN-code: 2369-3990

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The paper is concerned with the main methods and solutions for designing high-loaded software applications for mastering theoretical and practical skills of working with them due to the need to use them in many informational areas of present time. The author considered in detail the main technologies for providing access to high-loaded system, various options for their implementation, optimization methods to ensure stable operation. The example shows and presents the most effective solutions for one of the most used systems. In this paper, the author indicated the main ways of applying technologies for the main components of a software application.

Keywords

Fault-tolerant access, clustering, optimization, high load, servers, web application, replication, sharding, cluster, caching, database

Received 08.04.2019

© Bauman Moscow State Technical University, 2019

References

- [1] Murav'yev K.A., Terekhov V.V. Methods of management of the network traffic of the heterogeneous distributed telecommunication systems. *Proektirovanie i tekhnologiya elektronnykh sredstv* [Design and technology of electronic means], 2017, no. 2, pp. 15–21 (in Russ.).
- [2] Mosk D. Chto takoe klaster [What is cluster]. *dmosk.ru: website*. URL: <https://www.dmosk.ru/terminus.php?object=cluster> (accessed: 15.04.2019) (in Russ.).
- [3] Khan I. Raspredelenoe keshirovanie na puti k masshtabirovaniyu [Distributed caching on the way to scaling]. *msdn.microsoft.com: website* (in Russ.). URL: <https://msdn.microsoft.com/ru-ru/magazine/dd942840.aspx> (accessed: 15.04.2019).
- [4] Karpunin A.A., Ganey Yu.M., Chernov M.M. Quality assurance methods in complex program systems design. *Nadezhnost' i kachestvo slozhnykh sistem* [Reliability & Quality of Complex Systems], 2015, no. 2(10), pp. 78–84 (in Russ.).
- [5] Replikatsiya dannykh [Data replicating]. *ruhigload.com: website* (in Russ.). URL: <https://ruhigload.com/%D0%A0%D0%B5%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F+%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85> (accessed: 15.04.2019).
- [6] Masshtabirovanie baz dannykh — partitsionirovanie, replikatsiya i sharding [Data bases scaling – partitioning, replicating and sharding]. *web-creator.ru: website* (in Russ.). URL: https://web-creator.ru/articles/partitioning_replication_sharding (accessed: 15.04.2019).
- [7] Keshirovanie i memcached [Caching and memcached]. *habr.com: website* (in Russ.). URL: <https://habr.com/ru/post/42607/> (accessed: 15.04.2019).

- [8] Murav'yev K.A., Terekhov V.V. [Analysis of quality parameters of data transmission systems]. *Tr. mezhd. simp. Nadezhnost' i kachestvo* [Proc. Int. Symp. Quality and Reliability], 2018. vol. 1, pp. 128–133 (in Russ.).
- [9] Aver'yanikhin A.E., Kotel'nitskiy A.V., Murav'yev K.A. Method of calculation of optimum number of knots of the cluster of virtualization of the private cloud of virtual desktops by criterion of efficiency. *Mezhdunarodnyy nauchno-issledovatel'skiy zhurnal* [International Research Journal], 2016, no. 5-3(47), pp. 6–13 (in Russ.).
- [10] Gurenko V.V., Borodin A.F., Nazarkov D.A. Comparative analysis of frameworks for web development. *Tekhnologii inzhenernykh i informatsionnykh sistem* [Technologies of Engineering and Information Systems], 2017, no. 2, pp. 3–14 (in Russ.).
- [11] Groshev S.V., Pivovarova N.V. Special aspects of DBMS engineering for web-oriented system for quality assessment of Pareto approximation in multi-criteria optimization problem. *Tekhnologii inzhenernykh i informatsionnykh sistem* [Technologies of Engineering and Information Systems], 2015, no. 2, pp. 3–11 (in Russ.).

Olisevich E.A. — Student, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technology University, Moscow, Russian Federation.

Scientific advisor — Kondakov N.A., Engineer, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technology University, Moscow, Russian Federation.