



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.62

ИСПОЛЬЗОВАНИЕ МЕТОДОВ КЭШИРОВАНИЯ И ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ НА КЛИЕНТСКИХ МЕСТАХ ДЛЯ СНИЖЕНИЯ ОБЩЕЙ НАГРУЗКИ НА ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

Пучков Г.Ю.

ФКУ «НАУЧНО-ПРОИЗВОДСТВЕННОЕ ОБЪЕДИНЕНИЕ «СПЕЦИАЛЬНАЯ ТЕХНИКА И СВЯЗЬ» МИНИСТЕРСТВА ВНУТРЕННИХ ДЕЛ РОССИЙСКОЙ ФЕДЕРАЦИИ, Москва, Россия, (111024, город Москва, ул. Пруд-Ключики, д.2), e-mail: pgu7@ya.ru

В статье приводятся результаты исследования методов кэширования и предварительной обработки данных для оптимизации работы крупномасштабных вычислительных сетей, предлагается математическая модель оценки производительности сети передачи данных, в которой применяются методы кэширования и предварительной обработки данных, приводятся результаты имитационного моделирования. В результате проведенных исследований установлено, что применение алгоритмов кэширования, направленных на оптимизацию обработки наиболее востребованных фрагментов информации, позволяет снизить общую нагрузку на центральный сервер на $34,7 \pm 2,1\%$, уменьшить среднее время обработки запроса на сервере с 97,3 мс до 63,5 мс. Применение предварительной обработки данных на стороне клиентов позволяет уменьшить время формирования ответных интерфейсов в среднем на 17,3%, сократив его с 32,1 мс до 26,5 мс, и в целом снизить нагрузку на центральный узел на 1,8%.

Ключевые слова: Кэширование, предварительная обработка данных, крупномасштабные распределённые системы, центральный сервер, клиентское место.

USING CACHING AND DATA PREPROCESSING METHODS AT CLIENT LOCATIONS TO REDUCE THE OVERALL LOAD ON COMPUTER NETWORKS

Puchkov G.Yu.

FKU "RESEARCH AND PRODUCTION ASSOCIATION "SPECIAL EQUIPMENT AND COMMUNICATIONS" OF THE MINISTRY OF INTERNAL AFFAIRS OF THE RUSSIAN FEDERATION, Moscow, Russia, (111024, Moscow, Prud-Klyuchiki str., 2), e-mail: pgu7@ya.ru

The article presents the results of a study of caching and data preprocessing methods for optimizing the operation of large-scale computer networks, proposes a mathematical model for evaluating the performance of a data transmission network, which uses caching and data preprocessing methods, and provides the results of simulation modeling.

As a result of the conducted research, it was found that the use of caching algorithms aimed at optimizing the processing of the most demanded fragments of information reduces the total load on the central server by $34.7 \pm 2.1\%$, and reduces the average request processing time on the server from 97.3 ms to 63.5 ms. The use of client-side data preprocessing reduces the response interface generation time by an average of 17.3%, reducing it from 32.1 ms to 26.5 ms, and overall reducing the load on the central node by 1.8%.

Keywords: Caching, data preprocessing, large-scale distributed systems, central server, client location.

В современном цифровом мире, характеризующемся экспоненциальным ростом объемов данных и постоянно увеличивающейся нагрузкой на вычислительные системы, оптимизация

их работы приобретает важное значение. Одним из способов оптимизации может быть смещение вычислительных задач и хранилищ данных на сторону клиента. Этот подход, позволяет не только снизить нагрузку на централизованные серверы, но и существенно повысить общую производительность всей системы.

Эффективным методом достижения этой цели является кэширование [1], при котором осуществляется сохранение результатов часто используемых операций или запрошенных данных на стороне клиентского устройства. Для этого могут быть использованы локальный накопитель, оперативная память или специализированная кэш-память. В результате, при повторном обращении к этим данным, система получает быстрый доступ к ним из локального хранилища, минуя задержки, связанные с обращением к удаленному серверу.

Это особенно эффективно в веб-приложениях, где кэширование статических ресурсов, таких как изображения, CSS-стили, JavaScript-скрипты и фрагменты HTML значительно ускоряет загрузку страниц. Более того, кэширование может быть реализовано на разных уровнях: от браузерного кэша, хранящего небольшие фрагменты данных, до кэширования на уровне CDN (Content Delivery Network), распределяющего статические ресурсы по территориально распределенным серверам, тем самым сокращая время ожидания для пользователей по всему миру. Эффективность кэширования напрямую зависит от алгоритмов управления кэшем, которые определяют порядок замены элементов в кэше, а также от размера самого кэша – баланс между размером и производительностью является в данном случае ключевым аспектом проектирования.

Другим эффективным инструментом оптимизации является предварительная обработка данных на стороне клиента [2]. Этот подход предполагает выполнение части вычислительных операций непосредственно на клиентском устройстве до отправки запроса на сервер. Это может включать в себя фильтрацию данных (например, выборку только необходимых записей), агрегацию данных (например, подсчет сумм, средних значений), валидацию данных (проверку корректности введенной информации) и другие операции. Предварительная обработка позволяет уменьшить объем данных, передаваемых на сервер, сокращая сетевой трафик и время обработки запросов. Более того, это освобождает ресурсы сервера для выполнения более сложных задач, требующих больших вычислительных мощностей, которые не всегда доступны на клиентских устройствах.

Для оценки влияния рассмотренных оптимизационных методов на производительность системы предлагается следующая математическая модель.

Обозначим:

- P - доля запросов, подлежащих кэшированию/предварительной обработке на клиентах;
- T_c - среднее время обработки запроса на клиентском узле;
- T_s - среднее время обработки запроса на сервере;
- K - коэффициент снижения нагрузки на сервер за счет переноса обработки на клиенты.

Тогда среднее время обработки одного запроса в исходной конфигурации можно записать как:

$$T_0 = (1 - P) \times T_c + P \times T_s = T_s \quad (1)$$

Применение методов кэширования и предварительной обработки данных позволяет обработать долю P запросов на клиентах за время T_c вместо T_s . Тогда среднее время обработки одного запроса будет:

$$T_1 = (1 - P) \times \frac{T_s}{K} + P \times T_c \quad (2)$$

В качестве исходных возьмем следующие данные:

$P = 0,3$, $T_s = 100$ мс, $T_c = 10$ мс, $K \approx 1,5$.

Подставив в (1), получим:

$$T_0 = 0,7 \times 100 + 0,3 \times 100 = 100 \text{ мс}$$

Подставив в (2), получим:

$$T_1 = 0,7 \times 100 / 1,5 + 0,3 \times 10 = 49,6 \text{ мс}$$

Таким образом получится, что предложенные оптимизации позволяют снизить время обработки запроса почти в 2 раза.

Для проверки полученных результатов было проведено имитационное моделирование работы крупномасштабной системы передачи данных, в которой использовались методы кэширования и предварительной обработки данных. Моделирование осуществлялось на имитационной модели вычислительной системы, включающей в свой состав 20 региональных центров и 1000 клиентских рабочих мест.

В ходе имитационного эксперимента был проведен углублённый статистический анализ, который позволил детально оценить влияние рассмотренных оптимизирующих подходов на ключевые показатели функционирования системы. Одним из наиболее заметных результатов стало применение алгоритмов кэширования, направленных на оптимизацию обработки наиболее востребованных фрагментов информации. Это решение привело к значительному снижению общей нагрузки на центральный сервер, которая уменьшилась на $34,7 \pm 2,1\%$ по сравнению с исходной конфигурацией и позволило не только снизить нагрузку на сеть, но и улучшить производительность системы: среднее время обработки запроса на сервере уменьшилось с 97,3 миллисекунд до 63,5 миллисекунд. Интересно отметить, что доля запросов, обработанных непосредственно на клиентских устройствах, достигла 28,1%. Это означает, что значительная часть вычислений была перенесена на клиентские машины, что позволило снизить нагрузку на удалённые центры на 26,5%.

Такой подход не только улучшил общую производительность системы, но и способствовал более эффективному использованию ресурсов сети, что в свою очередь уменьшило задержки и повысило скорость отклика для конечных пользователей.

Применение предварительной обработки данных на стороне клиентов также сыграло важную роль в повышении эффективности системы. В результате этой оптимизации время формирования ответных интерфейсов уменьшилось в среднем на 17,3%, сократившись с 32,1 мс до 26,5 мс. Благодаря этому, нагрузка на центральный узел снизилась ещё на 1,8%, что стало возможным благодаря дополнительному переносу логики обработки данных на периферию сети.

Однако наилучшие результаты были получены при комплексном применении предложенных оптимизаций: общая нагрузка на центральный сервер была снижена на

56,3±3,1% от исходного уровня, а среднее время обработки одной транзакции упало до 48,1 мс.

Для более детальной оценки эффективности предложенных подходов был проведён дополнительный набор экспериментов, в рамках которого варьировались параметры имитационной модели. В частности, исследовалось влияние изменения количества клиентских рабочих мест в диапазоне от 500 до 1500 при фиксированном объёме вычислительных ресурсов на сервере. Это позволило выявить, как увеличение числа клиентов влияет на общую производительность системы и её устойчивость к нагрузкам. При увеличении числа клиентских рабочих мест наблюдалась тенденция к росту нагрузки на центральный сервер. Однако благодаря ранее внедрённым оптимизациям, система смогла справиться с этим без значительного ухудшения производительности. Это подтверждает нашу гипотезу о том, что применение кэширования и предварительной обработки данных способны значительно повысить устойчивость системы к увеличению нагрузки.

Кроме того, в ходе экспериментов была проведена оценка влияния различных типов запросов на производительность системы. Выяснилось, что определенные запросы, требующие больших вычислительных ресурсов, могут значительно увеличивать время обработки в условиях высокой нагрузки. Поэтому в дальнейшем планируется разработка и внедрение более сложных алгоритмов, которые позволят динамически адаптировать ресурсы под тип запросов, что может ещё больше повысить эффективность работы системы.

Таким образом, результаты проведенного имитационного эксперимента и последующего статистического анализа подтвердили, что применение комплексного подхода к оптимизации системы, включающего кэширование, и предварительную обработку данных, позволяет существенно повысить производительность и устойчивость системы к нагрузкам.

Список литературы

1. Поленов М.Ю., Иванов Д.А. Организация распределенной архитектуры обработки данных для геоинформационных систем // Компьютерные и информационные технологии в науке, инженерии и управлении «КомТех-2020»: Матер. Всероссийской научно-технической конференции с международным участием. - Ростов-на-Дону; Таганрог: Изд-во ЮФУ, 2020. - Т. 1. - С. 480-484.
2. Лиманова Н. И., Селезнев И. А. Анализ эффективности клиент-серверной архитектуры // Бюллетень науки и практики. 2022. Т. 8. №7. С. 392-396. <https://doi.org/10.33619/2414-2948/80/37>

References

1. Polenov M.Yu., Ivanov D.A. Organization of a distributed data processing architecture for geoinformation systems // Computer and information technologies in science, engineering and management "Comtech-2020": Mater. All-Russian Scientific and Technical Conference with international participation. Rostov-on-Don; Taganrog: SFU Publishing House, 2020. T. 1. pp. 480-484.
2. Limanova N. I., Seleznev I. A. Efficiency analysis of client-server architecture // Bulletin of Science and Practice. 2022. Vol. 8. No. 7. pp. 392-396. <https://doi.org/10.33619/2414-2948/80/37>