ИУ7-54Б, 16_KOZ, Турчанский

ОПРЕДЕЛЕНИЯ

В настоящей расчетно-пояснительной записке применяют следующие термины с соответствующими определениями.

Балансировщик нагрузки — это устройство, которое находится между пользователем и группой серверов и действует как невидимый посредник, обеспечивая одинаковое использование всех серверов ресурсов.[1]

Время ответа — это общее время, затрачиваемое сервером на обработку входящих запросов и отправку ответа.[1]

Вес — вероятность, с которой балансировщик нагрузки в следующий раз выберет этот узел [2].

Вычислительный узел (узел) — устройство, выполняющее основную логику обработки запроса [3] Распределенная система - это вычислительная среда, в которой различные компоненты распределены между несколькими компьютерами (или другими вычислительными устройствами)

Запрос — это единица нагрузки

ВВЕДЕНИЕ

На заре развития компьютеры (или ЭВМ, электронно-вычислительные машины) были очень дорогим и штучным инструментом, позволить который могли себе только наиболее крупные институты и предприятия. Вычислительные ресурсы приходилось экономить всеми возможными способами. Первые разработчики писали код в режиме «офлайн» и передавали их оператору ЭВМ, который последовательно вводил программы в машину и производил расчеты. В начале 1960-х годов зародилась концепция разделения времени – распределение вычислительных ресурсов между несколькими пользователями: пока один вводит данные, машина занимается расчетами других С увеличением масштабов компьютерных систем — когда они начали состоять из сотен единиц — и ростом мощности, механизмы разделения времени перестали быть актуальными. Понадобились средства, которые бы управляли нагрузкой на множестве компьютеров сразу [4].

Цель работы — исследование методов балансировки высоконагруженных систем.

Для достижения поставленной цели необходимо выполнить следующие задачи (дополнить):

— провести анализ предметной области балансировки нагрузки высоконагруженных систем;

— провести обзор существующих методов балансировки нагрузки высоконагруженных систем;

— сформулировать критерии сравнения методов балансировки нагрузки высоконагруженных систем;

— провести сравнение методов балансировки нагрузки высоконагруженных систем;

— сформулировать выводы.

1 Анализ предметной области

С ростом числа запросов к системе, встает вопрос о ее масштабировании. Масштабирование — это процесс роста системы со временем, для эффективной обработки все большего и большего количества запросов в единицу времени [5]. Выделяют два вида масштабирования: горизонтальное и вертикальное [3; 6; 7]. Вертикальное масштабирование происходит за счет увеличения мощности вычислительного узла. Однако, использования только такого подхода часто не хватает, поскольку постоянно увеличивая мощность, однажды будет достигнут «потолок» производительности и дальнейшие аппаратные улучшения будут недоступны. В таком случае для дальнейшего роста производительности применяют горизонтальное масштабирование, которое заключается в добавлении новых вычислительных узлов, выполняющих одинаковые функции. Для расширения возможностей горизонтального масштабирования используются балансировщики нагрузки [6; 7].

Балансировка нагрузки — это механизм, который позволяет перемещать задания с одного компьютерана другой в рамках распределенной системы (это процесс приблизительного выравнивания рабочей нагрузки между всеми узлами распределенной системы). Это ускоряет обслуживание заданий, например, сводит к минимуму время отклика на задание и повышает эффективность использования ресурсов. Некоторые из основных целей алгоритма балансировки нагрузки, как указано в, заключаются в следующем: добиться большего общего улучшения производительности системы при разумных затратах, например, сократить время отклика задачи при сохранении приемлемых задержек; одинаково относиться ко всем заданиям в системе независимо от их происхождения; обладать отказоустойчивостью: выносливостью производительности при частичном сбое в системе; иметь возможность модифицировать себя в соответствии с любыми изменениями или расширяться в конфигурации распределенной системы; и поддерживать стабильность системы: способность учитывать чрезвычайные ситуации, такие как внезапный всплеск поступлений, чтобы производительность системы не ухудшалась сверх определенного порога, одновременно предотвращая, чтобы узлы распределенной системы тратили слишком много времени на передачу заданий между собой вместо выполнения эти рабочие места. [8]

Следует еще четко понимать, о чем идет речь: о распределении или

балансировке нагрузки. Несмотря на свою схожесть, эти понятия нельзя назвать взаимозаменяемыми. Так, распределение нагрузки предполагает ее равномерное разделение между серверами. А вот балансировка – это уже ее частный случай, учитывающий ряд факторов, подверженных изменению. [8]

1.1 Алгоритмы решения задачи балансировки

Балансировщик нагрузки работает по одному из алгоритмов, решающих задачу балансировки. На вход этому алгоритму подается некоторое число запросов, приходящих в систему и набор вычислительных узлов, которыми располагает система. Задача алгоритма сводится к минимизации времени обработки запросов, за счет распределения запросов по вычислительным узлам.

Для анализа алгоритмов балансировки могут быть выделены следующие параметры [3]:

— точность прогнозирования — степень соответствия расчетных результатов работы алгоритма их фактическому значению;

— отказоустойчивость — показывает устойчивость алгоритма к возникновению разнообразных ошибок;

— время обработки нового запроса — время от поступления нового запроса до его перенаправления к цели.

Алгоритмы балансировки разделяют на статические и динамические [3].

1.2 Статическая балансировка

Статическая балансировка – это метод распределения нагрузки на узлы, основанный на заранее определенных параметрах. Основная цель статической балансировки состоит в том, чтобы равномерно распределить трафик между несколькими узлами, чтобы каждый узел получал приблизительно одинаковое количество нагрузки.

При использовании статической балансировки узлы могут быть настроены по разному:

— По равномерному распределению нагрузки: каждый узел получает аналогичное количество запросов.

— По приоритету: выделенные узлы имеют больший приоритет и получают больше запросов.

— По типу запроса: различные типы запросов могут быть отправлены на разные узлы для оптимизации обработки.

Процесс работы статической балансировки выглядит следующим образом:

— Балансировщик нагрузки получает запрос от клиента.

— Балансировщик принимает решение о том, на какой сервер направить запрос, основываясь на предварительных настройках.

— Балансировщик перенаправляет запрос на выбранный сервер.

— Сервер обрабатывает запрос и отправляет ответ клиенту через балансировщик.

Основные принципы статической балансировки:

— Предварительная настройка: перед запуском системы настраивается конфигурация балансировщика нагрузки. В этой конфигурации определяются узлы, которые будут использоваться, а также правила распределения нагрузки.

— Равномерное распределение: статический балансировщик нагрузки следит за текущим состоянием узлов и равномерно распределяет нагрузку между ними в соответствии с заданными правилами.

— Отказоустойчивость: статический балансировщик нагрузки способен обнаруживать отказы узлов и исключать их из распределения нагрузки. Это позволяет обеспечить непрерывную работу системы даже при отказе одного или нескольких узлов.

Преимущества статической балансировки:

— Простота настройки: статическая балансировка не требует сложной конфигурации и может быть быстро настроена.

— Эффективность: статическая балансировка обеспечивает равномерное распределение нагрузки между всеми узлами, что позволяет повысить производительность и отзывчивость системы.

— Надежность: отказоустойчивость статического балансировщика нагрузки позволяет обеспечить непрерывную работу системы в случае отказа одного или нескольких узлов.

Однако статическая балансировка также имеет свои недостатки:

— Отсутствие гибкости: при статической балансировке узлы настраиваются заранее и не могут мгновенно реагировать на изменение нагрузки или состояния системы.

— Одиночная точка отказа: если балансировщик отказывает, вся система может быть недоступна.

В целом, статическая балансировка является эффективным методом распределения нагрузки, но ее ограничения могут стать проблемой в высоконагруженных или динамических средах. В таких случаях могут быть предпочтительнее динамические методы балансировки, которые позволяют автоматически адаптироваться к изменениям нагрузки. Статическая балансировка является эффективным и надежным методом равномерного распределения нагрузки на сервера. Она позволяет повысить производительность, обеспечить отказоустойчивость системы, экономить ресурсы и легко масштабировать систему по мере роста нагрузки.

Статическая балансировка является одним из методов равномерного распределения нагрузки на сервера и имеет некоторые преимущества по сравнению с динамической балансировкой. [9]

1.3 Динамическая балансировка

Динамическая балансировка является одним из основных принципов равномерного распределения нагрузки на узлы. При использовании динамической балансировки, узлы могут эффективно управлять нагрузкой, обеспечивая стабильную работу и предотвращая перегрузку отдельных узлов.

Динамические алгоритмы осуществляют мониторинг состояния каждого из узлов и выбирают «наилучший», в рассматриваемый момент времени, из

них [webmanage]. К динамическим алгоритмам относятся:

- Least Connections
- Weighted Least Connections
- Least Time

1.3.1 Least Connections

Алгоритм Least Connections распределяет нагрузку между узлами в зависимости от количества активных соединений, обслуживаемых каждым узлом. Узел с наименьшим числом соединений будет обрабатывать следующий запрос, и узлы с большим числом соединений будет перераспределять свою нагрузку на узлы с меньшей загрузкой [10].

1.3.2 Weighted Least Connections

Данный алгоритм комбинирует принципы алгоритмов Least Connections и Weighted Round Robin [2]. Он учитывает как веса узлов, так и количество активных соединений. Новое сетевое подключение предоставляется узлу, который имеет минимальное отношение количества текущих активных подключений к его весу [11].

1.3.3 Least Time

Алгоритм Least Time сочетает время отклика узла и активные соединения для определения лучшего узла [1].

Основные принципы метода наименьшего времени ответа включают:

- Измерение времени ответа: Для эффективной работы этого метода необходимо непрерывно измерять время ответа от узла. Это может быть выполнено с помощью мониторинга, сбора статистики или других средств измерения производительности.
- Выбор узла с наименьшим временем ответа: Когда клиент отправляет запрос, система выбирает сервер с наименьшим текущим временем ответа. Это позволяет направлять запросы к узлу, который, по всей видимости, находится в наилучшем состоянии для обработки данного запроса.

- Динамическая адаптация: Время ответа от узла может изменяться со временем в зависимости от нагрузки и состояния узлов. Метод наименьшего времени ответа учитывает эти изменения и позволяет системе адаптироваться к текущей ситуации.
- Предотвращение перегрузки: Этот метод также может включать в себя механизмы для предотвращения перегрузки узлов, например, не отправляя новые запросы на узел, который уже перегружен.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое балансировка нагрузки? [Электронный ресурс]. — Режим доступа: https://aws.amazon.com/ru/what-is/load-balancing/ (дата обращения: 05.10.2023).
2. Алгоритм распределения нагрузки в программной системе, построенной на основе протокола HDP [Электронный ресурс]. — Режим доступа: https://cyberleninka.ru/article/n/algoritm-raspredeleniya-nagruzki-v-programmnoy-sisteme-postroennoy-na-osnove-protokola-hdp (дата обращения: 06.10.2023).
3. СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ БАЛАНСИРОВКИ НАГРУЗКИ В СРЕДЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ [Электронный ресурс]. — Режим доступа: https://cyberleninka.ru/article/n/sravnitelnyy-analiz-algoritmov-balansirovki-nagruzki-v-srede-oblachnyh-vychisleniy (дата обращения: 07.10.2023).
4. Сетевая эквилибристика: как развивались системы балансировки трафика. — Режим доступа: https://mcs.mail.ru/blog/load-balancer-istoriya-razvitiya.
5. А. М. Д. МАСШТАБИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЙ //. — Теория и практика современной науки, 2021.
6. Алгоритм распределения нагрузки в программной системе, построенной на основе протокола HDP. — https://cyberleninka.ru/article/n/algoritm-raspredeleniya-nagruzki-v-programmnoy-sisteme-postroennoy-na-osnove-protokola-hdp.
7. В. Г. А. О. К. В. Сравнение алгоитмов балансировки нагрузки //. — Инновационное развитие техники и технологий промышленности, 2020.
8. A Guide to Dynamic Load Balancing in Distributed Computer Systems. — https://www.researchgate.net/publication/268200851_A_Guide_to_Dynamic_Load_Balancing_in_Distributed_Computer_Systems.
9. Статическая и динамическая балансировка: объяснение и применение. — https://helpdoma.ru/faq/staticeskaya-i-dinamiceskaya-balansirovka-obyasnenie-i-primenenie?ysclid=lolze4rsx757530210.

10. Analisis Algoritma Round Robin, Least Connection, Dan Ratio Pada Load Balancing Menggunakan Opnet Modeler [Электронный ресурс]. — Режим доступа: https://www.neliti.com/publications/67705/analisis-algoritma-round-robin-least-connection-dan-ratio-pada-load-balancing-me (дата обращения: 06.10.2023).

11. An Improved Weighted Least Connection Scheduling Algorithm for Load Balancing in Web Cluster Systems [Электронный ресурс]. — Режим доступа: https://www.irjet.net/archives/V5/i3/IRJET-V5I3455.pdf/ (дата обращения: 06.10.2023).