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.[**balance**]

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

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

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

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 [**2**]. 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 [**1**].

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 [8], заключаются в следующем: (1) добиться большего общего улучшения производительности системы при разумных затратах, например, сократить время отклика задачи при сохранении приемлемых задержек; (2) одинаково относиться ко всем заданиям в системе независимо от их происхождения; (3) обладать отказоустойчивостью: выносливостью производительности при частичном сбое в системе; (4) иметь возможность модифицировать себя в соответствии с любыми изменениями или расширяться в конфигурации распределенной системы; и (5) 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. [**4**]

Это лучше для аналита: Различные исследования, например, [2]- [18],

показали, что балансировка нагрузки между узлами распределенной системы значительно повышает производительность системы и увеличивает использование ресурсов. Согласно [21], балансировка нагрузки позволяет еще больше снизить среднее и стандартное отклонение времени отклика задачи больше, чем при распределении нагрузки.[**4**].

1 Аналитический раздел

Алгоритмы балансировки можно разделить на статические и динамические [**uzel**].

1.1 Динамическая балансировка

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

1.1.1 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 [**leastconnection**].

1.1.2 Weighted Least Connections

Данный алгоритм комбинирует принципы алгоритмов Least Connections и Weighted Round Robin [**weightedroundrobin**]. 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 [**mainsource**].

1.1.3 Least Time

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

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.