

УДК 004.9

ПРОБЛЕМА БАЛАНСИРОВКИ НАГРУЗКИ В РАСПРЕДЕЛЕННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЕ

Нехаев А.П., Громова Г.М., студенты гр. ПИМ-191, I курс
Научный руководитель: Крюкова В.В., кандидат технических наук, доцент
Кузбасский государственный технический университет имени Т. Ф.
Горбачева г. Кемерово

Проблему планирования нагрузки предпочтительно решить на этапе развития проекта, на самой ранней его стадии. Выход из строя сервера, так называемое «падение», несет за собой большие проблемы, в частности из-за того, что происходит всегда в самый неподходящий момент и приводит к огромным материально-моральным убыткам. Изначально недостаток мощности сервера решается путем наращивания мощностей и оптимизацией алгоритмов, но эти способы действенны лишь до определенного момента, дальше эти способы оказываются недостаточными.

Балансировка нагрузки применяется для оптимизации вычислений, помогая распределять нагрузку на процессоры равномерно. При увеличении нагрузки, программное обеспечение, установленное на ЭВМ, помогает решать проблему распределения нагрузки (баланса), решая вопрос на каком конкретно узле будут выполняться вычисления, а также, перераспределение нагрузки с самых загруженных узлов на менее загруженные.

За долгую историю обеспечения баланса выявлено много проблем. В современном мире проблема баланса вышла на уровень информационных систем. Для своевременного проведения балансировки системы и правильной нагрузки на узлы системы необходимо понимать, когда и как проводить ее.

Поговорим о распределенной информационной системе, что это и какие особенности она в себе несет. Данная система представляет собой совокупность взаимосвязанных компонентов, связанных и взаимодействующих друг с другом. Данные компоненты не привязаны к конкретной роли и могут быть приложением, реализующимся в отдельном процессе, - так реализуется в распределенных системах, без зависимости происхождения взаимодействия, то есть *где, когда, как*. В такой системе координат информационные системы подобно «масонам», должны скрывать от конечного пользователя различия и связи между компьютерами. Одной из важнейших особенностей распределенных систем является способность обеспечить единообразной работой пользователя в системах. В продолжение исследования особенностей распределенной системы, хотелось бы отметить, что поломка части распределенной системы никак не должна влиять на работу всей распределенной системы, и тем более конечный пользователь не должен знать о том, что часть системы в данный момент вышла из строя.

Так как мы говорим про балансировку нагрузки в распределенных системах, хотелось бы дать определение понятию балансировки нагрузки. Под балансировкой нагрузки понимается такое разделение вычислительной нагрузки между процессорами многопроцессорной вычислительной системы, которое позволяет занять каждый процессор полезной нагрузкой по возможности большую часть времени [1].

Так же нужно понимать разницу между статистической и динамической балансировкой [2].

При распределении процессов статистическая балансировка опирается на операции и выполнение алгоритмов прошлого, используя данный опыт. Данный вид балансировки выполняется в самом начале, до начала работы всех приложений. Следует помнить, что предварительное размещение бесполезно, потому что среда, в которой происходят все процессы, очень изменчива и банально может перестать быть работоспособным один из узлов, что приведет в свою очередь к дисбалансу и увеличению нагрузки на конкретные узлы. Данная информация помогает осознать, что распределение процессов с целью выполнения обработки в то же время имеет малый процент эффективности или считается вовсе неэффективным.

В свою очередь, говоря про динамическую балансировку, надо отметить, что вычислительные нагрузки на узлы перераспределяются во время работы.

При реализации динамической балансировки выделяют [3]:

- частоту обменов сообщениями;
- загрузку узлов;
- пропускную способность;

Основываясь на данных, решается переносить или нет процессы с одного узла на другой. Так, выделяется несколько уровней для достижения баланса нагрузки процессоров [4]:

- балансировка операционной системы, разделением нагрузки со сменой процессов во время выполнения процесса;

- балансировка промежуточного программного обеспечения, балансировка нагрузки запроса или в контексте сессии:

- балансировка пользовательского приложения, алгоритмы, которые реализуют баланс в параллельно функционирующих приложениях.

Выделяют следующие подходы к реализации балансировки [5]:

- Клиентский.

Реализуется на компьютере пользователя, он выбирает сервер. С помощью интеллектуального выбора предоставляется возможность выбрать сервер, при просмотре страниц навигатор выбирает один из случайных серверов и отправляет ему запрос. Конечно, элемент случайности, как и везде, не гарантирует никаких точных положительных результатов, в данном случае тенденция сохраняется. Выбор сервера можно реализовать с помощью различных апплетов, которые в свою очередь запускаются на ПК пользователя и предназначены для возможности отслеживать состояния

серверов и задержек. Как и у всего существует недостаток, в виде задержки которая провоцируется определением состояний серверов.

- DNS и диспетчерский.

Подход основывается на решении проблемы балансировки на стороне системы. Благодаря программному обеспечению, сервер способен определять и выделять один из серверов для обработки запросов, также подобное программное обеспечение позволяет DNS определять загруженность тех или иных серверов, которые в свою очередь могут быть находиться в разных точках планеты. Сервер способен определить задержку, доступность и выбрать наиболее подходящий, использующий клиент-серверную технологию. Для контролирования запросов используется отображение с помощью IP-адреса. Диспетчеру отдается полный контроль над маршрутизацией и при необходимости перенаправлять пакеты, переписывая IP. Роль диспетчера выполняет TCP-маршрутизатор. В качестве недостатков данного подхода является расходы на изменение адреса.

- Серверный.

Данный подход использует двухуровневую диспетчеризацию, данный подход определяет сервер, к которому пойдет запрос, после чего сервер может переадресовать запрос к любому другому серверу, благодаря данной системе перераспределения нагрузки, все сервера могут перенаправлять запросы и тем самым балансировать нагрузку.

Цели для достижения балансировки:

- справедливость: необходимо обеспечить уверенность, что запрос обработается и на него будут выделены ресурсы, а также минимизировать ситуации, при которых получается очередь из запросов;

- эффективность: необходимо создать ситуацию занятости серверов, не допускать простоя серверов (учитывая, что мы говорим про идеальные условия загрузки серверов на 100% и отсутствия простоя, в реальности же достижение данных целей не всегда реально);

- сокращение времени выполнения;

- сокращение времени отклика;

Рассмотрим этапы решения проблемы балансировки.

Для того, чтобы начать балансировку, нужно определить момент, когда она необходима, но учитывать факт, что чрезмерное выполнение балансировки будет замедлять весь процесс. Также не стоит забывать, что проведение балансировки, - это финансовые затраты и чрезмерное выполнение может нивелировать выгоду от балансировки. Для того, чтобы начать балансировку, следует определить момент, когда возникает потребность в балансировке. Сопоставить целесообразность проведения и выгоды проведения балансировки нагрузки. Существует два вида определения дисбаланса: синхронный и асинхронный.

При синхронном происходит остановка всех процессоров и начинается процесс сравнения показателей каждого процессора со средним

показателем загрузки. Особенностью асинхронного вида определения баланса является то, что каждый процессор способен хранить историю загрузок, таким образом, данный процесс становится фоновым и остановка уже не требуется.

Рассмотрим принятие решений в процессе балансировки. Существует два класса динамической балансировки: централизованная стратегия и распределенная стратегия.

При централизованной стратегии весь процесс сбора информации о состоянии системы лежит на одном выделенном компьютере, который по итогу и принимает решение о перераспределении задач в случае необходимости.

При распределенной стратегии балансировки каждый процессор обладает собственным алгоритмом, с помощью которого, способен выполнить обмен данными без участия одного «особенного» компьютера.

Последним этапом решения проблемы балансировки является перемещение объектов. Следуя информации из предыдущего пункта, где на данном этапе уже известно о том, что объекты будут перемещены по разным процессорам. Главное в процессе перемещения – сохранение его целостности, также для сложных данных используются дополнительные функции приложения.

Причины возникновения проблемы вычислительной нагрузки. Неоднородность структуры, так как разные процессы могут требовать разные мощности, производительность и пропускную способность, а также сюда можно отнести и неоднородность кластера.

Адаптивность, производительность, надежность, простота в обслуживании - главные свойства распределенных систем. Как и всё в этом мире, распределенные системы имеют ряд недостатков и проблем.

Мы рассмотрим три проблемы и начнем с самой важной для сервера - это его «падение» [2, 5].

Событие, которое происходит редко, но метко, главная задача человека, следящим за соблюдением баланса системы, вычислить сервер, находящийся в критическом состоянии и убрать его из системы, в противном случае производство рискует потерять сервер.

Во-вторых, рассмотрим проблему хранения сессий. В современном мире время - главный ресурс и отнимать его у пользователя для того, чтобы он в сотый раз вводил свои учетные данные, было бы преступлением, поэтому сессионные данные нужно хранить. Один из вариантов хранения сессий в одной базе данных, к которой есть общий доступ среди прикладных серверов. Другим подходом является привязка пользователя к одному серверу на время одной сессии, данный метод является более гибким. В качестве недостатков можно отметить то, что нагрузка возрастает, и в связи с этим надежность может пострадать.

Третьей проблемой отметим проблему кэширования содержания. Подход такой же, как и у хранения сессий, кэш хранится в сетевой папке или базе данных, к которой есть общий доступ среди прикладных серверов.

В заключение хотелось бы подвести итог.

Использование балансировки напрямую влияет на расширение эффективности использования серверов, влияет на рост отказоустойчивости, потому что балансировка не прекращает своей работы после выхода из строя нескольких элементов системы, влияет также на возможность расширить кластер или же уменьшать его.

В век современных информационных технологий мы имеем дело с огромным множеством различных способов балансировки нагрузки. Отдавая предпочтение какому-то одному алгоритму или методу, следует учитывать фактор направления и тематики проекта и учитывать поставленные задачи и цели, в таком случае, выбор метода может быть успешным решением для экономии времени сил и ресурсов.

В процессе работы были исследованы методология и подходы к решению проблемы балансировки распределенной информационной системы.

Список литературы:

1. Балансировка нагрузки в распределенных системах 2012. URL: <http://masters.donntu.org/2012/fknt/volokhova/library/article3.htm> (дата обращения 20.07.2015).
2. Проблемы распределенных систем, автор научной работы — Цветков Виктор Яковлевич, Алпатов Алексей Николаевич 2014. URL: <https://cyberleninka.ru/article/n/problemy-raspredeleennyh-sistem> (дата обращения 23.07.2018).
3. Исследование стратегий балансировки нагрузки в системах распределенной обработки данных. 2009. URL: <https://cyberleninka.ru/article/n/issledovanie-strategy-balansirovki-nagruzki-v-sistemah-raspredelenoy-obrabotki-dannyh> (дата обращения 21.06.2017).
4. Алгоритмы динамической балансировки вычислительной нагрузки и их реализации. URL: https://dspace.kpfu.ru/xmlui/bitstream/handle/net/51926/___Main_48_20_63.pdf (дата обращения 22.07.2019).
5. Обзор методов балансировки нагрузки в гетерогенных распределенных файловых системах. 2017. URL: <https://fundamental-research.ru/ru/article/view?Id=41743> (дата обращения 23.07.2018).