

УДК 004.75

ОБЗОР МЕТОДОВ БАЛАНСИРОВКИ НАГРУЗКИ В ГЕТЕРОГЕННЫХ РАСПРЕДЕЛЕННЫХ ФАЙЛОВЫХ СИСТЕМАХ**Дворников В.С., Долгов В.В., Венцов Н.Н.***ФГБОУ ВО «Донской государственный технический университет», Ростов-на-Дону,
e-mail: vasily.v.dolgov@yandex.ru*

В данной статье проводится обзор существующих подходов балансировки нагрузки в неоднородных системах хранения данных. Описываются некоторые проблемы распределенных систем, в частности осуществляющих хранение и обработку файлов, важнейшей из которых является проблема перегрузки рабочих узлов. Обозначены цели балансировки нагрузки. Приведены основные типы стратегий, их преимущества и недостатки, а также применимость различных подходов к решению проблемы в гетерогенной среде. По устройству такие системы подразделяются на централизованные, распределенные и гибридные схемы. Для каждой схемы существуют свои сценарии применимости, некоторые из них можно найти в данной статье. Описаны такие стратегии балансировки нагрузки, как случайный выбор, циклически алгоритм, стратегия наименьшей нагрузки, зондирование и переговоры. Содержится упоминание направлений и областей науки, которые могут представить эффективные способы решения проблемы балансировки нагрузки в гетерогенных распределенных файловых системах.

Ключевые слова: балансировка нагрузки, распределенная система, файловая система, узел, стратегия, гетерогенность, эффективность, динамический подход

REVIEW OF LOAD BALANCING METHODS IN HETEROGENEOUS DISTRIBUTED FILE SYSTEMS**Dvornikov V.S., Dolgov V.V., Ventsov N.N.***Don State Technical University, Rostov-on-Don, e-mail: vasily.v.dolgov@yandex.ru*

This article reviews the various load balancing approaches in heterogeneous storage systems. Some problems of distributed systems are described, in particular, those that store and process files, the most important of which is the problem of overloading storage nodes. The main types of load balancing strategies are given. Their advantages and disadvantages are listed, as well as the applicability of different approaches to solving the problem in a heterogeneous environment. By internal structure such systems are divided into centralized, distributed and hybrid circuits. Each scheme has its own applicability scenarios, some of which can be found in this article. We describe such load balancing strategies as: random selection, cyclical algorithm, least-load strategy, probing and negotiation. There is a mention of directions and areas of science that can provide effective ways to solve the problem of load balancing in heterogeneous distributed file systems.

Keywords: load balancing, distributed system, file system, storage node, strategy, heterogeneity, efficiency, dynamic strategy

Распределенные файловые системы в отличие от классических систем хранения файлов характеризуются разделением хранения данных на множестве узлов. Такой подход позволяет добиться возможности широкого масштабирования и, как следствие, больших объемов хранения, а также повышения отказоустойчивости системы и эффективности многопользовательского доступа. Основной целью создания распределенных файловых систем является прозрачность деталей хранения для пользователя, таких как местоположение данных, репликации, миграции, параллелизмы, отказов. Гетерогенные распределенные системы представляют высокий интерес, исходя из того, что современные кластерные системы обычно имеют неоднородную структуру. Это объясняется тем, что после отказа существующего оборудования целесообразнее заменить его на более новое, что также обычно означает и изменение характеристик [1]. Также распределенные си-

стемы могут состоять из различных типов оборудования, что обеспечивает большую гибкость таких систем.

Построение систем с распределенной обработкой данных обычно представляет нетривиальную задачу. В процессе проектирования архитектуры подобных систем возникает немало проблем, которые необходимо решить, прежде чем перейти к реализации. Основными характеристиками распределенной файловой системы должны быть отказоустойчивость и масштабируемость. Данные характеристики достигаются благодаря множеству различных приемов, важнейшим из которых является балансировка нагрузки на узлах системы [2]. Так как основным предметом функционирования файловой системы является хранение и обработка файлов, в рамках рассматриваемой проблемы, под нагрузкой нужно понимать клиентские запросы к системе, данные хранимые узлами, а также задания, выполняемые вычислительными узлами [3].

Как правило, распределенные системы предоставляют многопользовательский доступ. С ростом количества пользователей, а также их обращений к системе растет и нагрузка на сервера выполняющие обработку таких запросов [4]. В ситуации постоянно возрастающей нагрузки необходимы инструменты, позволяющие распределить ее на узлах системы таким образом, чтобы система эффективно справлялась со своими задачами, имела приемлемое время отклика и при этом сводились к минимуму ситуации частичного или полного отказа системы из-за нагрузки на отдельные узлы [5].

Целью балансировки нагрузки в распределенной системе может являться как повышение качества обслуживания клиентов, так и повышение эффективности работы самой системы. Данные цели не исключают друг друга и могут достигаться одновременно. В отношении повышения качества обслуживания клиентов это может быть среднее время отклика, среднее время выполнения задания. Если говорить о достижении большей эффективности системы, то такими критериями могут стать рациональное использование ресурсов системы, а также производительность системы.

В рамках решаемой проблемы балансировки нагрузки эффективный алгоритм должен решать следующие задачи: выбор задания на передачу, осуществление информационного обмена между узлами, выбор целевого узла исполнителя [6].

Типы балансировки нагрузки

Для построения гетерогенной распределенной файловой системы необходимо подобрать наиболее эффективный способ решения проблемы балансировки нагрузки. Существует множество подходов для решения этой задачи. На рис. 1 приведены основные типы систем балансировки нагрузки.

Прежде всего следует рассмотреть типы выполнения балансировки, дифференцированные по моменту выполнения и, как следствие, оперирующие разным объемом информации о системе. В данном случае различают статическую и динамическую балансировку [7].

Статическая балансировка производится перед началом работы распределенной системы. Такой подход зачастую основан на знаниях о предшествующих отработках целевой системы или подобных систем и позволяет распределить ответственность за обработку запросов между узлами на основе этих знаний. Статическая балансировка может продемонстрировать преимущества в системах с предсказуемой нагрузкой. В случае систем, отличающихся стохастической неопределенностью выполнения заданий, данный подход не позволяет эффективно распределить обработку по нескольким причинам: динамическое изменение среды выполнения, нагрузка на узлах в подобных системах не может быть предсказана однозначно в начале работы [7, 8].

В отличие от статической балансировки, выполняемой перед началом работы системы, динамический подход позволяет ориентироваться на состояние системы в текущий момент и учитывать данные метрики при осуществлении распределения нагрузки. Так как в данном случае нас интересует гетерогенная распределенная система, нагрузка в отдельный момент времени которой подвержена стохастической неопределенности, динамический подход позволяет эффективно производить балансировку на узлах такой системы [7–9]. На этапе выполнения существует возможность снятия показателей системы, позволяющих произвести решение, таких как текущая нагрузка на узлах, доступность узлов, пропускная способность каналов передачи данных и т.д.

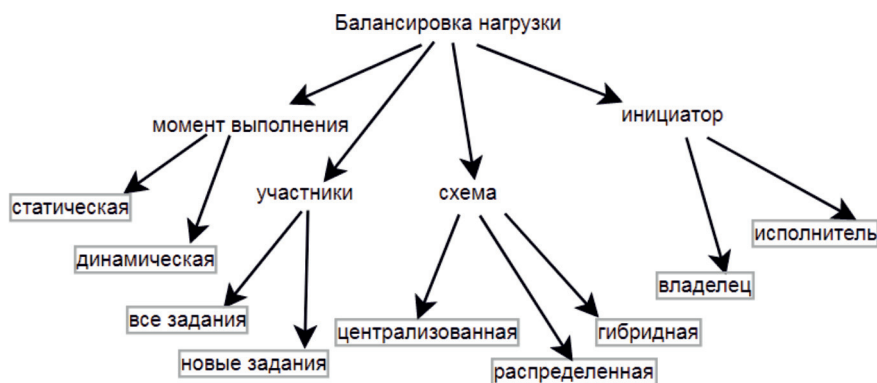


Рис. 1. Основные типы систем балансировки нагрузки

Некоторые исследования динамической балансировки нагрузки показывают, что алгоритмы, принимающие решение о такой балансировке с использованием большого количества подробно собранных данных о системе, не дают какого-то существенного прироста производительности системы, относительно алгоритмов, полагающихся при вынесении решения на минимальный набор информации [10].

Алгоритмы могут отличаться по типу инициаторов. Существуют подходы, при использовании которых инициатива о передаче задания поступает от владельца задания. Данный подход хорош тем, что возможна минимизация информационного обмена между узлами системы, в случаях, когда узел-владелец решает исполнять задание локально, а также балансировка иницируется только тогда, когда имеется задание на распределение. В качестве альтернативы инициатором балансировки могут выступать узлы-исполнители, создавая заявки, обозначающие готовность узла принять дополнительную нагрузку [9, 11]. При использовании данной стратегии балансировка нагрузки находится в состоянии частичного исполнения если в системе не существует текущих заданий, так как заявки иницируются по мере освобождения узлов, в отличие от предыдущего подхода. Существующие исследования показывают большую эффективность стратегии инициатора-исполнителя в системах испытывающих сильную нагрузку, в отличие от стратегии инициатора-владельца, которая достаточно хорошо работает в умеренно нагруженных системах [12].

Балансировка нагрузки может осуществляться только на прибывших заданиях, либо в балансировке может участвовать все множество активных заданий, что усложняет сам процесс балансировки, но согласно некоторым исследованиям такой подход дает лучшее распределение [13].

По типу устройства системы балансировки нагрузки могут быть как централизованными, так и распределенными. Централизованная схема предполагает расположение управляющего компонента на единственном узле. Иначе схема, в которой каждый узел участвует в процессе балансировки нагрузки, собирает данные о текущей нагрузке, принимает решение о обработке запроса, а также выполняет миграцию данных, имеет распределенную структуру [6].

Централизованная схема балансировки нагрузки обычно влечет минимальные издержки на коммуникацию узлов, так

как вся информация о состоянии системы концентрируется в едином месте. Минусом данного подхода может являться низкая отказоустойчивость, так как при выходе из строя центрального узла, ответственного за балансировку нагрузки, такая схема перестает работать. Также в данном случае централизация распределения заданий или подключений сама по себе может вызвать перегрузку центрального звена [9].

Распределенная схема балансировки лишена недостатков централизованной и в свою очередь может выполняться с использованием кооперативного и не кооперативного подходов [14]. Если говорить о кооперативном подходе, при котором решение о передаче заданий выполняется на основе коллективного решения, в данном случае может возникнуть высокая нагрузка системы связи, связанная с издержками на коммуникацию между узлами [15]. Решением данной проблемы видится минимизация избыточного взаимодействия узлов в рамках распределенного алгоритма балансировки нагрузки. В рамках некооперативных распределенных стратегий осуществляются индивидуальные решения узлов о распределении нагрузки, на основе информации об общем состоянии системы [16]. Некооперативный подход позволяет снизить издержки на коммуникацию, а также воспользоваться основными преимуществами распределенного подхода, такими как отказоустойчивость и рациональное использование ресурсов [9].

В качестве альтернативы может использоваться смешанная, частично распределенная схема балансировки нагрузки, где множество всех узлов разделено на подмножества (кластеры), среди которых в свою очередь властвует централизованная политика [9]. Данная модификация позволяет воспользоваться преимуществами как распределенной, так и нераспределенной схемы.

Существуют исследования, в рамках которых показано преимущество схемы централизованной балансировки нагрузки по сравнению с другими схемами, в системах, полагающихся на небольшой набор рабочих узлов [17].

Способы динамической балансировки нагрузки

Случайный выбор. Простейшей стратегией балансировки нагрузки является случайный выбор. Используя данный подход в самом простом случае нет нужды собирать данные о нагрузке системы и со-

стояниях компонентов. Определение узла, который получит задание на обработку запроса, производится на основе случайного выбора [9]. Преимуществом данного алгоритма является простота его реализации и низкая нагрузка на систему балансировки, но ценность данных преимуществ сомнительна с учетом недостатка в виде распределения нагрузки без учета информации о загруженности узлов, что может являться критичным и привести к серьезным последствиям, в виде частичного или полного отказа системы. Случайная стратегия дает неплохие результаты относительно систем, не использующих балансировку, и часто используется в качестве отправной точки для сравнения других алгоритмов балансировки нагрузки [18]. На рис. 2 изображена работа стратегии случайного выбора с использованием централизованной схемы.

ма в гетерогенной распределенной системе является неоправданным, так как учет нагрузки системы и сложности задач не осуществляется, вследствие чего резко возрастает риск отказа системы [19]. Взвешенный циклический алгоритм является усовершенствованной версией циклической стратегии и ориентирован на работу в гетерогенной среде [20]. Суть алгоритма в том, что каждый узел маркируется определенным весом, который обозначает производительность обработчика. В зависимости от величины данного веса узел получает большее количество заданий [21, 22]. На рис. 3 изображена работа взвешенного циклического алгоритма с использованием централизованной схемы.

Стратегия наименьшей нагрузки. При поступлении нового запроса балансировщик нагрузки использующий данную

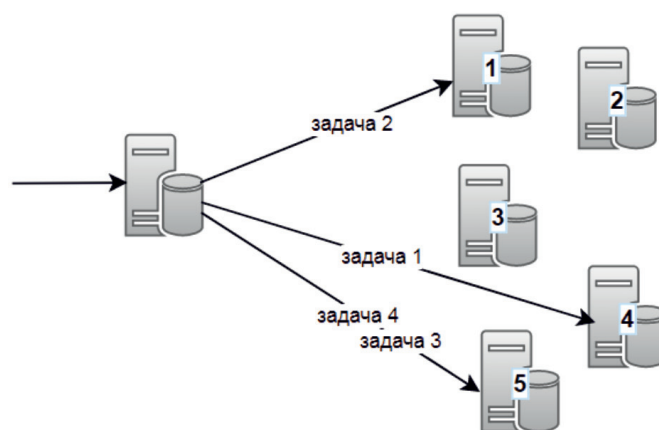


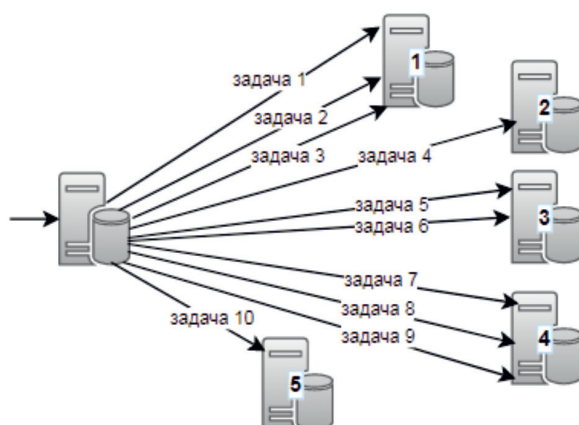
Рис. 2. Централизованная схема стратегии случайного выбора

Циклический алгоритм. Данный алгоритм подразумевает распределение задач по обработчикам с использованием циклического или кругового подхода [7]. Множество задач X определяются на обработчики Y итеративно, так что каждому $X[i]$ обработчику достается $Y[j]$ задача, где $j = [0, \text{size}(Y))$, а $i = j \bmod \text{size}(X)$. Циклическая стратегия обычно применяется, когда задачи по своей сложности и структуре являются типичными. Такой алгоритм достаточно прост в исполнении и предполагает наивную балансировку нагрузки, рассчитывая на предположение о том, что, когда очередь на обработку повторно обратится к узлу, он успеет освободить достаточно ресурсов для обработки следующего запроса. В данном случае использование кругового алгорит-

стратегию определяет количество исполняемых запросов на узлах и осуществляет соединение с наименее нагруженным из них [7]. Данный алгоритм удачно подходит для систем, в которых нагрузка в большей степени определяется количеством выполняемых заданий. Как и в случае с циклической стратегией балансировки нагрузки, у данного алгоритма существует расширенная версия, использующая информацию о производительности сервера. В данном случае усовершенствованная версия оперирует не только текущей нагрузкой, но также учитывает и веса узлов при осуществлении балансировки [20]. На рис. 4 изображена работа взвешенной стратегии наименьшей нагрузки с использованием централизованной схемы.

Зондирование. Существует 3 стратегии, которые работают в рамках данного подхода: стратегия предельного значения, жадная стратегия и кратчайшая стратегия. В рамках стратегии предельного значения выбирается случайный узел, оценивается уровень нагрузки при передаче и исполнении задания, на этом узле, в случае, когда такая нагрузка оказывается выше порогового значения выбирается следующий узел. Если за определенное количество шагов целевой узел не был найден, то задание выполняется локально. Использование данной стратегии позволяет достичь значительного прироста производительности по сравнению со случайной стратегией [8]. Жадная стратегия является разновидностью пороговой стратегии, в свою очередь использует циклический

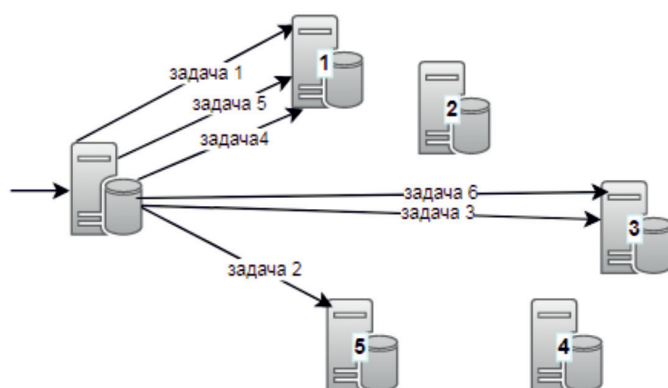
подход, при зондировании узлов. При использовании кратчайшей стратегии выбирается случайное подмножество узлов, в данном подмножестве определяется наименее загруженный узел. Если длина очереди обработки на таком узле меньше, чем определенный порог, то работа передается, иначе она выполняется локально. Значительного прироста в отношении стратегии предельного значения такая стратегия не дает [9]. На рис. 4 изображена работа взвешенной стратегии наименьшей нагрузки с использованием централизованной схемы. На рис. 5 продемонстрирована распределенная схема жадной стратегии зондирования, в которой максимальным числом оценок установлено 3, а в качестве порога используется уровень нагрузки владельца.



Начальное состояние системы

Узел	Мощность
1	100
2	20
3	70
4	100
5	20

Рис. 3. Централизованная схема взвешенного циклического алгоритма



Начальное состояние системы

Узел	Мощность	Нагрузка
1	100	0
2	20	90
3	70	40
4	100	100
5	20	40

Рис. 4. Централизованная схема взвешенной стратегии наименьшей нагрузки

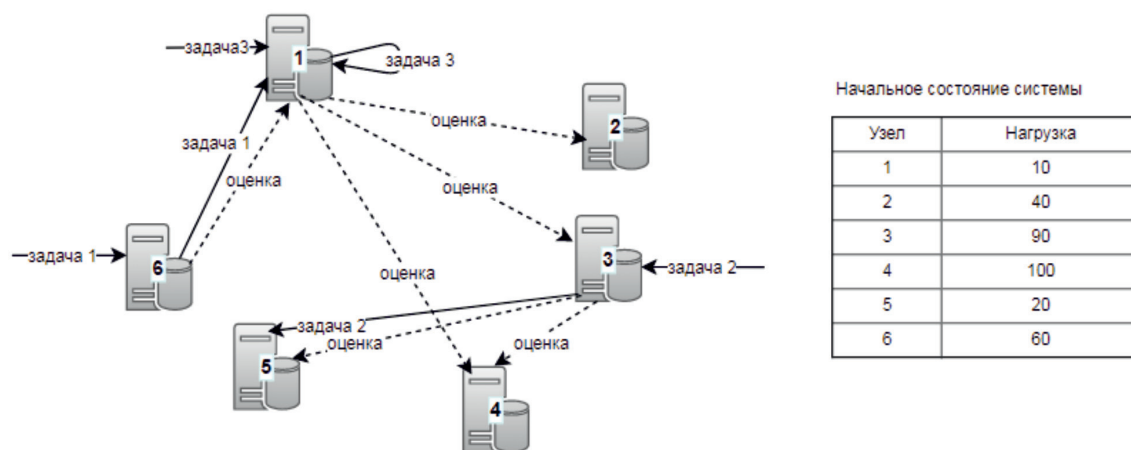


Рис. 5. Распределенная схема жадной стратегии зондирования

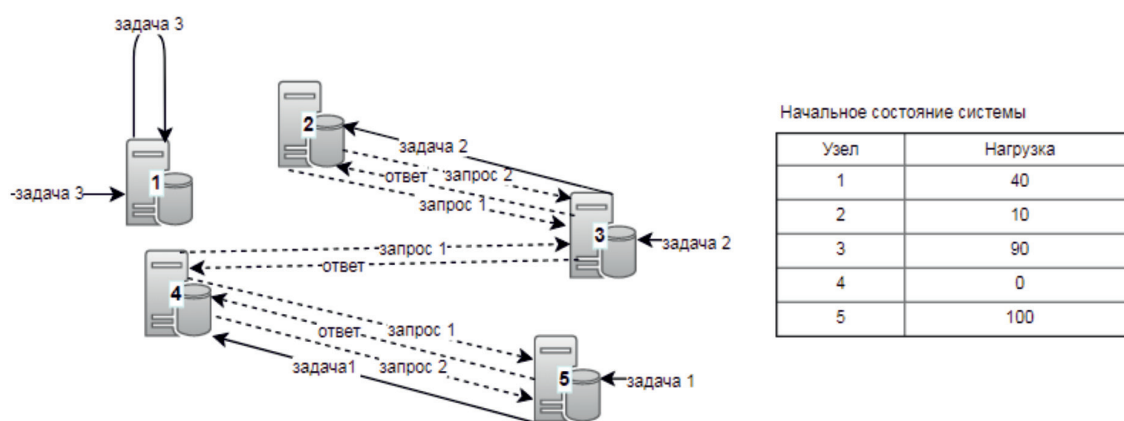


Рис. 6. Распределенная схема стратегии переговоров

Существуют алгоритмы балансировки, основанные на концепции переговоров. В стратегии переговоров нагруженный узел инициирует запрос-заявку, в которой описывает свою нагрузку, а также задания, которые он хотел бы передать. Удаленный узел проверяет данный запрос и сравнивает нагрузку узла инициатора со своей, в случае если она меньше, этот удаленный узел отправляет заявку на исполнение задания инициатору. Иначе узел игнорирует сообщение. В итоге узел-инициатор получает заявки на принятие заданий и выбирает из них самые незагруженные узлы. Проблемой данного подхода может быть ситуация, в которой узлы с меньшей загрузкой могут стать перегруженными в результате победы во многих одновременных переговорах. Избавиться от этой проблемы можно, используя ограничение на количество заявок [9]. Существует другая версия стратегии переговоров. Узлы разделены на 3 группы: слабо нагруженные,

умеренно нагруженные и сильно нагруженные. Узлы наблюдают за своей нагрузкой и периодически меняют свои состояния, оповещая об этом остальные узлы. Таблица состояния узлов хранится у каждого узла. Слабо нагруженные узлы отправляют заявки на принятие заданий всем сильно загруженным узлам. Узлы, испытывающие сильную загруженность, отправляют ответ, в котором содержится информация о задачах, которыми данный узел мог бы поделиться. Когда узел-инициатор собирает все ответы, он принимает решение, какие задачи он может принять, отправляя новый запрос нагруженным узлам. Нагруженные узлы в свою очередь отправляют задания, уменьшая собственную нагрузку [9]. Стратегия переговоров может эффективно использоваться в распределенной схеме балансировки. На рис. 6 продемонстрирована распределенная схема подхода переговоров, инициатором в данном случае выступает исполнитель.

Достоинства и недостатки рассмотренных методов

Метод	Достоинства	Недостатки
Случайный выбор	Простота реализации Низкая нагрузка на системы связи и балансировки	Низкая эффективность Нет учета текущей нагрузки Нет учета производительности узлов
Циклический алгоритм	Простота реализации Низкая нагрузка на системы связи и балансировки Направленная смена узлов Учет производительности узлов (взвешенная версия)	Низкая эффективность Нет учета текущей нагрузки
Стратегия наименьшей нагрузки	Учет текущей нагрузки Высокая эффективность (относительно стратегии случайного выбора и циклического алгоритма) Учет производительности узлов (взвешенная версия)	Издержки на коммуникацию (сбор динамической информации)
Стратегия зондирования	Учет текущей нагрузки Высокая эффективность Возможность учета производительности узлов Ориентированность на распределенную схему	Издержки на коммуникацию (сбор динамической информации)
Стратегия переговоров	Учет текущей нагрузки Высокая эффективность Варианты инициаторов Возможность учета производительности узлов Ориентированность на распределенную схему	Издержки на коммуникацию (сбор динамической информации) Сложность реализации

Кластеризация задач. Существует стратегия балансировки нагрузки, основанная на кластеризации узлов сети, имеющих схожие характеристики, такие как пропускная способность сети, количество и частота процессоров, дисковое пространство и т.д. Такое разделение может использоваться для отображения групп задач на отдельные кластеры узлов [23, 24]. Данная стратегия хорошо сочетается с частично распределенной схемой балансировки.

Балансировка нагрузки также может осуществляться с использованием известных алгоритмов, вдохновленных природой, теории игр и генетических алгоритмов [6, 15, 16, 25].

В таблице представлены некоторые достоинства и недостатки рассмотренных методов динамической балансировки нагрузки.

Заключение

Сегодня существует множество подходов к решению проблемы балансировки нагрузки в распределенных системах, в частности распределенные файловые системы не являются исключением. Данные подходы могут использоваться для улучшения качества обслуживания клиентов, повышения производительности системы, а также более рационального использования ресурсов. В качестве преимуществ эффективной балансировки нагрузки стоит отметить поддержку отказоустойчивости и масштабируемости системы.

Работа выполнена при поддержке РФФИ, проект № 16-01-00390.

Список литературы

1. Heath T. et al. Energy Conservation in Heterogeneous Server Clusters // Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming. – 2005. – P. 186–195.
2. Болодурина И.П., Парфенов Д.И. Алгоритмы комплексной оптимизации потребления вычислительных ресурсов в облачной системе дистанционного обучения // Вестник Оренбургского государственного университета. – 2013. – Vol. 9, № 158. – P. 177–184.
3. Sundaram K.S. Load balancing for distributed file systems // Int. J. Adv. Comput. Eng. Netw. – 2013. – Vol. 1, № 1. – P. 2320–2106.
4. Deshmukh S.C. A Survey: Load Balancing for Distributed File System Sudarshan S Deshmukh // Int. J. Comput. Appl. – 2015. – Vol. 111, № 5. – P. 975–8887.
5. Натальченко И.А. Анализ механизмов передачи крупных массивов данных через сеть интернет с помощью технологии веб-сервиса // Инженерный вестник Дона. – 2008. – № 4. URL: <http://xn--80aqiwl.idon.ru/ru/magazine/archive/n4y2008/98>.
6. Ramu Y., Kavitha D., Swathi R. A Framework to Identify Node-Load by Decision Tree in Dynamic Load Balancing Mechanism // Int. J. Adv. Res. Comput. Sci. Softw. Eng. – 2016. – Vol. 6, № 3. – P. 19–23.
7. Ali M.F., Khan R.Z. The Study On Load Balancing Strategies In Distributed Computing System // Int. J. Comput. Sci. Eng. Surv. – 2012. – Vol. 3, № 2. – P. 19–30.
8. Casavant T.L., Kuhl J.G. A taxonomy of scheduling in general-purpose distributed computing systems // IEEE Trans. Softw. Eng. – 1988. – Vol. 14, № 2. – P. 141–154.
9. Alakeel A.M. A Guide to Dynamic Load Balancing in Distributed Computer Systems // IJCSNS Int. J. Comput. Sci. Netw. Secur. – 2010. – Vol. 10, № 6. – P. 153–160.
10. Eager D.L., Lazowska E.D., Zahorjan J. Adaptive load sharing in homogeneous distributed systems // IEEE Trans. Softw. Eng. – 1986. – Vol. SE-12, № 5. – P. 662–675.
11. Barmon C., Faruqi M., Battacharjee G. Dynamic load balancing algorithm in a distributed system // Microprocess. Microprogramming. – 1991. – Vol. 29, № 5. – P. 273–285.
12. Eager D.L., Lazowska E.D., Zahorjan J. A comparison of receiver-initiated and sender-initiated adaptive load sharing (extended abstract) // IEEE Trans. Softw. Eng. – 1986. – Vol. 6. – P. 53–68.

13. Goswami K.K., Devarakonda M., Iyer R.K. Prediction-based dynamic load-sharing heuristics // IEEE Trans. Parallel Distrib. Syst. – 1993. – Vol. 4, № 6. – P. 638–648.
14. Xiao Z. et al. Learning non-cooperative game for load balancing under self-interested distributed environment // Appl. Soft Comput. J. – 2017. – Vol. 52. – P. 376–386.
15. Grosu D., Chronopoulos A.T., Ming-Ying Leung. Load balancing in distributed systems: an approach using cooperative games // Proceedings 16th International Parallel and Distributed Processing Symposium. IEEE, 2002. – P. 501–510.
16. Grosu D., Chronopoulos A.T. Noncooperative load balancing in distributed systems // J. Parallel Distrib. Comput. – 2005. – Vol. 65, № 9. – P. 1022–1034.
17. Zhou S. A trace-driven simulation study of dynamic load balancing // IEEE Trans. Softw. Eng. – 1988. – Vol. 14, № 9. – P. 1327–1341.
18. Varsha J., Mandre B.R. Balancing of Load for Distributed File Systems in Clouds Using Load Rebalancing // Int. J. Comput. Sci. Mob. Comput. – 2016. – Vol. 5, № 1. – P. 170–177.
19. Kaur S., Kumar K., Singh J. Round-robin based load balancing in Software Defined Networking // Comput. Sustain. Glob. Dev. – 2015. – P. 2136–2139.
20. Wang W., Casale G. Evaluating Weighted Round Robin Load Balancing for Cloud Web Services // 2014 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. IEEE, 2014. – P. 393–400.
21. Привалов А.Н., Клепиков А.К. Автоматическая балансировка нагрузки в гибридных вычислительных сетях // Известия Тульского государственного университета. Технические науки. – 2013. – Vol. 1, № 9. – P. 188–194.
22. Singh J. Weighted Round-Robin Load Balancing Using Software Defined Networking // Int. J. Adv. Res. Comput. Sci. Softw. Eng. – 2016. – Vol. 6, № 6. – P. 621–625.
23. Cirou B., Jeannot E. Triplet: A clustering scheduling algorithm for heterogeneous systems // Proceedings International Conference on Parallel Processing Workshops. IEEE Comput. Soc, 2001. – P. 231–236.
24. Gandhi A., Zhang X., Mittal N. HALO: Heterogeneity-Aware Load Balancing // 2015 IEEE 23rd International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. IEEE, 2015. – P. 242–251.
25. Rajeshkannan R., Aramudhan M. Comparative Study of Load Balancing Algorithms in Cloud Computing Environment // Indian Journal of Science and Technology. – 2016. – Vol. 9, № 20. – P. 1–7.