

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ БАЛАНСИРОВКИ НАГРУЗКИ ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА

Крюков Д.А., Суглобов Д.С.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский Технологический университет» (МТУ), 119454, Россия, г. Москва, проспект Вернадского, д. 78, e-mail: justQuantum@ya.ru, dm.bk@bk.ru

Эффективность функционирования кластеров зависит от модуля балансировки нагрузки, а именно, от применяемого в нем алгоритма балансировки нагрузки. В статье проведен анализ базовых алгоритмов балансировки с позиции их влияния на скорость обработки запросов вычислительным кластером. Для анализа использовано программное решение имитационного моделирования. В результате моделирования получены сравнительные значения производительности рассмотренных алгоритмов, а также выявлены зависимости, оказавшие влияние на полученный результат.

Ключевые слова: балансировка нагрузки, Round-robin, Least Connections, Weighted Round-robin.

COMPARATIVE ANALYSIS OF COMPUTING CLUSTER LOAD BALANCING ALGORITHMS

D. Kryukov, D. Suglobov

Federal State Educational Institution of Higher Education "Moscow Technological University", 119454, Russia, Moscow, Vernadscogo avenue, 78, e-mail: justQuantum@ya.ru, dm.bk@bk.ru

The efficiency of clusters depends on the load balancing module, namely, on the load balancing algorithm used in it. The paper gives a review of the main balancing algorithms in terms of their impact on the speed of query processing computing cluster. The software solution for simulation modeling was used for the analysis. As a result of the simulation, were obtained the comparative values of the performance of the considered algorithms and revealed the factors that influencing the obtained values.

Key words: load balancing, Round-robin, Least Connections, Weighted Round-robin.

Необходимость решения комплексных вычислительных задач, выполнение которых с использованием бытовых компьютеров потребует выделения существенного временного ресурса, обуславливает востребованность различных вариантов систем балансировки нагрузки, различающихся по архитектуре построения, сложности и стоимости. Одним из таких решений является принцип построения высокопроизводительных систем, под названием «кластер».

Кластер является системой, внешне функционирующей как единое целое, но внутри представленной множеством связанных компьютеров, служащих для решения общих задач. Таким образом, их вычислительные ресурсы объединены с целью повышения эффективности решения общей задачи. Как правило, включенные в кластер компьютеры, по своей производительности не сильно превосходят обычные стационарные персональные компьютеры.

Одним из основных составляющих любого кластера является модуль балансировки нагрузки. Он выполняет задачи управления ресурсами кластера, то есть, распределения поступающих на обработку вычислительных задач. Разработано множество алгоритмов функционирования модуля балансировки нагрузки, учитывающих существенное количество параметров, которые являются отражением особенностей системы, результатом их интерпретации конкретным проектировщиком.

В статье рассматриваются основные алгоритмы балансировки, а именно, их влияние на производительность при распределении задач. Результирующая оценка производительности получается по итогам построения упрощенных моделей кластеров и имитации их работы. При выборе алгоритма балансировки для системы, в первую очередь, учитываются особенности конкретного набора решаемых задач, а также стратегии работы системы. В общем случае выделяют следующие возможные стратегии:

- минимизация средней длины очереди запросов;
- сокращения периодов простоя серверов;

– минимизации времени обработки запросов.

В статье основной анализируемой стратегией и опорой для получения базовой метрики производительности, используемой для сравнения алгоритмов, является минимизация времени обработки запросов.

Для проведения исследования построена упрощенная модель кластера с модулем балансировки нагрузки с использованием программного обеспечения для имитационного моделирования AnyLogic. С целью оптимизации возможности унифицированного использования модели при ее проектировании учтены несколько важных аспектов, оказывающих влияние на объективность получаемых в дальнейшем результатов работы.

Во-первых, имитационная модель учитывает, что узлы, которые включены в кластер, могут иметь различную производительность, а, соответственно, и разную продолжительность выполнения абсолютно идентичных вычислительных задач.

Во-вторых, задачи, поступающие на обработку в кластер, также могут иметь различные условные параметры сложности решения, влияющие на продолжительность их выполнения. Таким образом, несколько задач могут иметь различное время обработки на абсолютно идентичных с точки зрения производительности узлах.

Стоит отметить, что построенная модель кластера является лишь примером по причине применения различных допущений и, в том числе, аппроксимации логики работы его элементов. В работе рассмотрен частный случай реализации кластера – он имеет фиксированное количество узлов, в то время как варианты с иным количеством узлов не затрагиваются. Этот факт может повлиять на показания производительности алгоритмов при больших расхождениях в количестве узлов. Следовательно, результаты, показанные алгоритмами, следует оценивать как приближенные, и в условиях, когда характеристики реального кластера схожи с экспериментальной моделью.

В моделируемый кластер включено 5 узлов, каждый из которых имеет свою вычислительную мощность, выраженную в модели коэффициентами, со значениями, принадлежащими интервалу (0; 1]. В качестве измеряемых характеристик использованы: среднее время обработки запроса кластером и средний размер очереди задач, ожидающих обработки. Совокупность этих параметров позволяет показать, насколько быстро кластером производится обработка задач.

Так, средняя длина очереди показывает, существуют ли ситуации, когда определенный узел, или целая группа узлов, не справляются с потоком поступающих задач, а время обработки – степень оптимальности выбора вычислительного ресурса модулем балансировки нагрузки. Управление моделируемой нагрузкой на кластер производится одним, внешним по отношению к кластеру параметром – частотой поступления запросов (количеством запросов в секунду).

Построенная в среде AnyLogic модель представлена на рисунке.

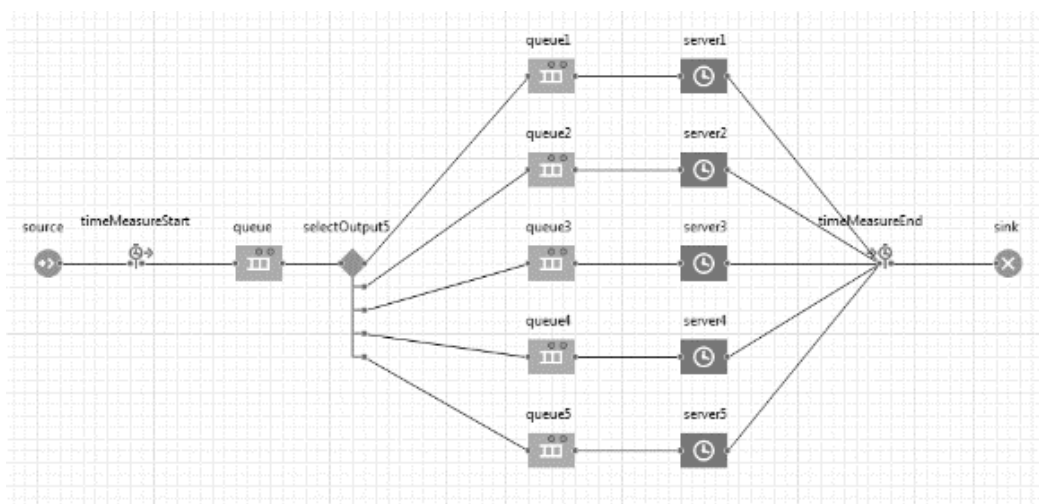


Рис.1. Модель кластера в среде AnyLogic

Приведем описание основных параметров, имеющих значение в имитационной модели:

frequency – максимальное количество запросов, поступающих на обработку в секунду.

serversPerformance – массив коэффициентов производительности узлов.

Также приведем основные элементы модели:

Source – источник запросов; запросы поступают согласно коэффициенту интенсивности.

SelectedOutput5 – модуль балансировки нагрузки представлен в виде элемента, распределяющего запросы.

Узлы имитируются связкой Queue (очередь запросов) и Delay (задержка на обработку запросов).

Task – тип, представляющий собой задачу. Имеет поле duration – коэффициент продолжительности выполнения (вычислительной сложности) задачи, случайно генерируемый в пределах [0.000001, 1] [1].

Далее рассмотрим принципы работы алгоритмов балансировки нагрузки. Для сравнения выбраны четыре алгоритма балансировки:

1. алгоритм равновероятной балансировки (РВБ);
2. алгоритм Round-robin (RR);
3. алгоритм Weighted Round-robin (WRR);
4. алгоритм Least Connections (LC).

Алгоритм равновероятной балансировки

Под равновероятной балансировкой понимается, что распределение модулем балансировки нагрузки поступающих задач производится на основании вероятности передачи задачи на обработку на один из узлов кластера. То есть, вероятность выбора любого сервера представляется одинаковой. Поскольку количество серверов в кластере равно 5, то вероятность выбора каждого из них равна 0,2.

Алгоритм Round-robin

Это циклический алгоритм распределения нагрузки в вычислительных системах, игнорирующий состояние узлов и соединения. Является одним из самых простых для реализации.

Так, например, имеется 5 узлов, способных выполнять множество задач. Первая поступившая на обработку задача отдается на исполнение первому узлу, вторая – второму и т.д. Шестая же задача отдается на исполнение уже первому (т.к. узлов всего 5), и далее по циклической логике [2].

Алгоритм Weighted Round-robin

Данный алгоритм, как и его базовая версия – RR, имеет циклическую природу, однако, имеет и существенные отличия. Он был создан специально для улучшения производительности кластеров, узлы которых имеют различную производительность. Так, каждому узлу назначается весовой коэффициент – целое число, отражающее его производительность. Узлы с большими весами получают большее количество задач, по сравнению с узлами с меньшими весовыми коэффициентами. В то же время узлы, чьи весовые коэффициенты равны, получают одинаковое количество задач [3].

Алгоритм Least Connections

Рассмотренный выше алгоритм RR, в исходном варианте, не учитывает один из существенных факторов – текущую загруженность выбранного узла. Алгоритм LC, в отличие от RR, в свою очередь, учитывает текущую нагрузку на узлы. Работа алгоритма заключается в том, что при поступлении новой задачи на обработку в кластер производится поиск наименее загруженного узла, которому задача и передается на обработку [3].

Сравнение производительности алгоритмов

Для получения результатов моделирования использовались два набора коэффициентов мощности узлов – {0,8, 1, 0,7, 0,5, 0,7} и {1, 1, 1, 1, 1}. Первый представляет собой ситуацию, когда узлы имеют различную мощность, второй – одинаковую мощность.

На рисунке 2 представлена сравнительная диаграмма продолжительности обработки задач при различных параметрах входной нагрузки и кластера. На рисунке 3 – сравнительная диаграмма средней длины очереди задач. Столбцы диаграмм соответствуют порядковому номеру набора данных (с 1-го по 4-й) для каждого, отдельно взятого, алгоритма. Следует отметить, что максимальные значения, равные 1,5 у алгоритмов РВБ и RR не являются реальными – они составляют 378,497 и 408,676 соответственно (рисунок 2). Аналогично, значение 0,6 для РВБ и RR составляет 2322,413 и 2400,474 соответственно (рисунок 3). Усечение значений сделано с целью повышения читаемости представленных диаграмм.

Первый набор результатов (столбцы №1 на диаграммах) получен при различных коэффициентах производительности узлов и максимальной частоте поступления запросов, равной 50. Первые два места по продолжительности обработки задач (в порядке от меньшей к большей) занимают алгоритмы, в том или ином виде учитывающие текущую нагрузку на узлы кластера. LC, занявший первое место, отличается от WRR совсем незначительно – разница составляет около процента. Алгоритмически, разница между LC и WRR состоит лишь в том, что второй отдает предпочтение самым производительным узлам, а первый – самым незагруженным.

Последние два места, в свою очередь, занимают алгоритмы, не опирающиеся на параметры кластера, и, следовательно, схожие по принципу работы – РВБ и RR. Разница в результатах между ними составляет около

7%, при этом, теоретически, РВБ может вести себя абсолютно аналогично RR.

Из первого набора экспериментальных данных можно сделать вывод, что крайне существенным параметром для алгоритмов балансировки нагрузки является текущая загруженность узла.

Второй набор результатов (столбцы №2 на диаграммах) получен при моделировании кластера, имеющего узлы с совершенно одинаковой (максимальной) производительностью. Наилучший результат вновь показал алгоритм LC, имеющий на 2.8% лучшую производительность, чем RR, занявший второе место.

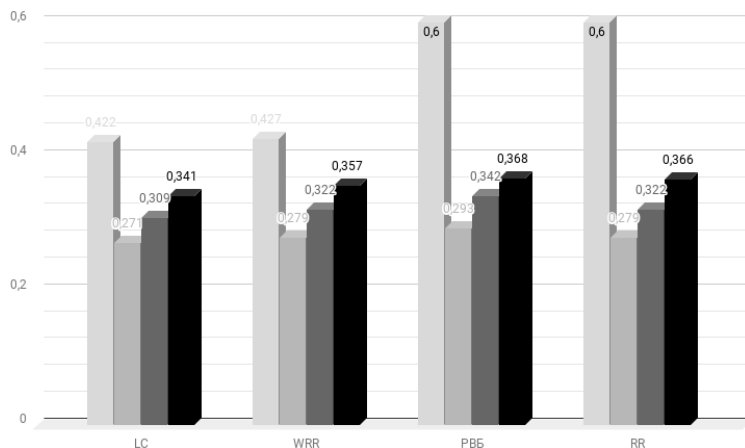


Рис. 2. Продолжительность обработки задач

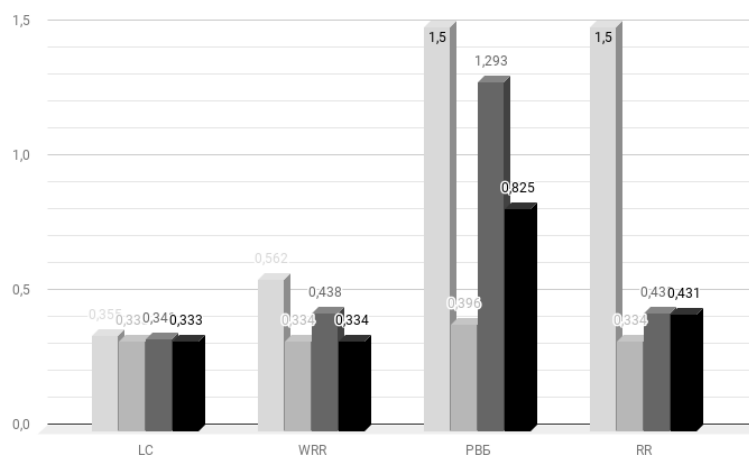


Рис. 3. Средняя длина очереди задач

В целом, все алгоритмы данного набора показали схожие результаты. Это можно объяснить высокой производительностью узлов кластера – не возникает ситуаций, при которых какой-либо узел кластера полностью загружен, а, соответственно, задержка между поступлением запроса на обработку и его обработкой минимальна. В этом можно убедиться, рассмотрев значения средней длины очереди, которая у всех алгоритмов близка к 0.333. Также следует отметить, что алгоритмы RR и WRR показали абсолютно идентичную производительность, поскольку при одинаковых значениях весовых коэффициентов узлов принцип их работы полностью идентичен.

Третий набор результатов (столбцы №3) получен при моделировании кластера с идентичными параметрами производительности узлов, но большей поступающей нагрузке (при значении коэффициента 70). Как и во всех предыдущих результатах, первое место занимает алгоритм LC с отрывом в 4% от второго места – WRR. Алгоритмы RR и WRR показали абсолютно одинаковые результаты, поскольку WRR, в случае, когда производительность всех узлов одинакова, представляет собой RR [4].

В четвертом наборе (столбцы №4) результатов моделирования, полученном при пониженном значении поступления запросов (20) и различной производительности узлов, можно отметить, что, как и в прошлых тестах, первое место занимает LC.

На рисунке 4 представлена сравнительная диаграмма средней продолжительности выполнения задач кластером при применении рассматриваемых алгоритмов балансировки при различных параметрах кластера и нагрузке. Алгоритмы отсортированы по средней продолжительности выполнения задач, а значения результатов для каждого алгоритма – в следующей очередности тестов (наборов результатов): 2, 3, 4, 1. Такая сортировка позволяет легко сравнить производительность рассмотренных алгоритмов.

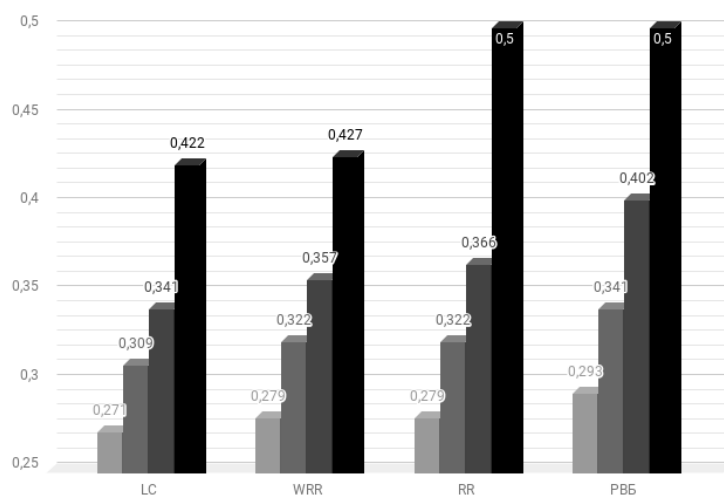


Рис. 4. Средняя продолжительности выполнения задач кластером при применении алгоритмов балансировки нагрузки с различными входными параметрами (меньше значение – выше производительность)

Полученные результаты имитационного моделирования алгоритмов балансировки нагрузки показали, что наибольшую производительность показывают алгоритмы, учитывающие текущую нагрузку на узлы вычислительного кластера. В основном, учет данного параметра позволяет максимально сократить время простоя узлов, и, при этом, уменьшить средний размер очереди запросов, что также положительно отражается на общей производительности кластера. Также, на основании полученных данных, можно сделать вывод, что для кластеров, со схожими с имитационной моделью характеристиками, лучшие результаты, как в производительности, так и в стабильности, показывает алгоритм LC.

Литература

1. Боев В. Компьютерное моделирование: Пособие для практических занятий, курсового и дипломного проектирования в AnyLogic7 / В.Д. Боев. – СПб.: ВАС, 2014. – 432 с.
2. Tai-hoon K., Soksoo K. Analysis of Security Session Reusing in Distribution Server System // Computational Science and Its Applications - ICCSA 2006. / Springer, 2006. - 1045 с.
3. Хританков А. Модели и алгоритмы распределения нагрузки. Алгоритмы на основе сетей СМО // Журнал Информационные технологии и вычислительные сети. – 2009. – №3.
4. Иванисенко И., Кириченко Л., Радивилова Т. Методы балансировки с учетом мультифрактальных свойств нагрузки // International Journal "Information Content and Processing". – 2015. – Vol. 2. – Num. 4. – С. 345 – 368.

References

1. Boev V. Kompjuterное modelirovanie: Posobie dlja prakticheskikh zanjatij, kursovogo i diplomnogo proektirovanija v AnyLogic7 [Computer modeling: A manual for practical classes, course and diploma projects in AnyLogic7] St. Petersburg, VAS Publ., 2014, 432p. (In Russian)
2. Tai-hoon K., Soksoo K. Analysis of Security Session Reusing in Distribution Server System . Computational Science and Its Applications - ICCSA 2006. Springer, 2006, 1045 p.
3. Khritankov A. Modeli i algoritmy raspredelenija nagruzki. Algoritmy na osnove setej SMO [Models and algorithms of load balancing. Algorithms on the basis of networks of queuing systems]. Informacionnye tehnologii i vychislitel'nye seti – Information technologies and computer networks. 2009, vol. 3. (In Russian)
4. Ivanisenko I., Kirichenko L., Radivilova T. Metody balansirovki s uchetom multifraktal'nyh svojstv nagruzki [Balancer multifractal methods considering load characteristics]. International Journal "Information Content and Processing". 2015, vol. 2, no. 4, pp. 345–368. (In Russian)