

УДК 004.75

АЛГОРИТМ БАЛАНСИРОВКИ САМОПОДОБНОЙ НАГРУЗКИ

К. т. н. Т. А. Радивилова, И. Н. Иванисенко, д. т. н. Л. О. Кириченко

Харьковский национальный университет радиозлектроники

Украина, г. Харьков

tamara.radivilova@gmail.com

В работе предложен динамический алгоритм балансировки нагрузки, который учитывает самоподобную структуру потоков и состояние элементов распределенной системы. Имитационное моделирование показало, что данный алгоритм обеспечивает более равномерное распределение нагрузки на серверах, высокие показатели производительности и низкое время отклика.

Ключевые слова: балансировка нагрузки, самоподобный трафик, распределенная система.

Современные сети связи характеризуются значительными слабо предсказуемыми колебаниями нагрузки. Отсутствие встроенных механизмов инжиниринга трафика ставит вопрос о необходимости разработки методов, позволяющих эффективно использовать возможности существующей сетевой инфраструктуры. Одним из наиболее перспективных вариантов решения указанных проблем на сегодняшний день являются динамические механизмы балансировки трафика, вызывающие пристальный интерес научного сообщества [1—3]. Существует много алгоритмов балансировки нагрузки, которые учитывают интенсивность, приоритетность запросов, однако они не учитывают самоподобные свойства входящего трафика. Самоподобный трафик имеет особую структуру, которая сохраняется на многих масштабах – в реализации всегда присутствует некоторое количество очень больших выбросов при относительно небольшом среднем уровне трафика. Мультифрактальный трафик определяется как расширение самоподобного трафика за счет учета масштабируемых свойств статистических характеристик второго и выше порядков [4]. Самоподобный трафик вызывает значительные задержки и потери пакетов, даже если суммарная интенсивность всех потоков далека от максимально допустимых значений.

Целью работы является разработка динамического алгоритма балансировки нагрузки в компьютерных сетях связи на основе мониторинга загруженности серверов распределенной системы с учетом самоподобной структуры трафика и проведение сравнительного анализа различных алгоритмов балансировки нагрузки.

На основании анализа мультифрактальных свойств входящего трафика предлагается динамический алгоритм балансировки трафика (MDA-multifractal dynamic algorithm), который имеет следующие этапы работы.

1. В трафике, поступающем на вход коммутатора, выделяем окно фиксированной длины, в котором определяются основные фрактальные и статистические характеристики входящих потоков.

2. Проводим сбор и анализ статистической информации — доступной пропускной способности $Net_i(t)$, состояния серверов $CPU_i(t)$, $Ram_i(t)$ — объем свободного ЦПУ и объем свободной оперативной памяти i -го сервера в момент времени t соответственно.

3. На основе мультифрактальных свойств и интенсивностей потоков вычисляем необходимое количество ресурсов для каждого q -го класса обслуживания трафика.

4. Проводим расчет распределения потоков по узлам сети с учетом классификации трафика и загруженности серверов и каналов связи. На основе полученных данных рассчитывается загруженность серверов на следующем шаге.

5. Распределяем трафик по серверам в пределах каждого класса потока.

6. Проводим распределение недооценки рассчитанного количества ресурсов $Net_i(t)$, $CPU_i(t)$, $Ram_i(t)$.

7. Проводим сбор данных о загруженности серверов $Net_i(t)$, $CPU_i(t)$, $Ram_i(t)$ и передачу их в

систему балансировки нагрузки для расчета нового распределения потоков.

9. Передвигаем окно вперед на заданную величину сдвига и осуществляем анализ трафика и прогноз следующего значения загруженности серверов.

В работе было проведено имитационное моделирование распределенной системы, которая состояла из четырех серверов различной производительности. Сервера соединены с балансировщиком линиями связи с разной пропускной способностью. В качестве входящих потоков генерировались потоки запросов, обладающие мультифрактальными свойствами [4], и обеспечивающие среднюю нагрузку серверов, равную 0,7. Входной поток был разбит на классы обслуживания и трудоемкости запросов. С помощью имитационного моделирования выполнен анализ разработанного алгоритма MDA и еще трех основных динамических алгоритмов балансировки нагрузки [2, 3]:

1. Task Scheduling based on LB (TSB) состоит из двухуровневого механизма планирования заданий и обеспечивает высокую эффективность использования ресурсов. Алгоритм выполняет балансировку нагрузки путем первичного распределения задач в виртуальных машинах, а затем всех виртуальных машин на ресурсы хостов, таким образом улучшая время отклика, использование ресурсов и эффективность работы среды облачных вычислений.

2. Two phase scheduling load balancing algorithm (TFS) — это сочетание алгоритмов планирования, которое обеспечивает более высокую эффективность работы и поддержку системы балансировки нагрузки. TFS держит каждый узел в рабочем состоянии, чтобы достичь цели балансировки нагрузки, и используется, чтобы минимизировать время выполнения каждого задания на узле.

3. Shortest Response Time First (SJF). Идеей этого алгоритма является прямая переадресация. В нем каждому процессу назначается приоритет для его запуска, который является обратным по отношению к следующему всплеску процессора: если всплеск процессора увеличивается, то приоритет понижается. В этом алгоритме короткие задачи выполняются перед длинными задачами.

Было проведено сравнение приведенных алгоритмов и предложенного алгоритма MDA по основным показателям эффективности — времени отклика, использованию ресурсов, затратам и производительности. Имитационное моделирование показало, что алгоритм TSB дает самое низкое время отклика, у MDA — на 2% выше, у SJF — на 2,5% и на 3% выше у алгоритма TFS. Затраты самые низкие у алгоритма TFS, на 3% выше у MDA, у SJF выше на 3,5% и у TSB — на 4%. По эффективности использования ресурсов алгоритм MDA лидирует, следом идет TSB, за ним алгоритмы SJF и TFS. По показателю производительности также лидирует алгоритм MDA, следом идет TSB, за ним алгоритмы TFS и SJF.

Благодаря анализу и учету самоподобных свойств входного потока с помощью работы предложенного динамического алгоритма балансировки нагрузки увеличивается средняя производительность распределенной системы на 5%, увеличивается степень использования ресурсов на 6%, уменьшается время отклика на 3% при тех же самых затратах.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Dharmesh Kashyap, Jaydeep Viradiya. A survey of various load balancing algorithms in cloud computing // international journal of scientific & Technology Research.— 2014.— Vol 3(11).— P. 115—119.
2. Wenhong Tian, Yong Zhao. Optimized cloud resource management and scheduling. Theories and Practices // Morgan Kaufmann.— 2014. P. 135—148.
3. Игнатенко Е. Г., Бессараб В. И., Турупалов В. В. Алгоритм адаптивной балансировки нагрузки в кластерных системах // Моделювання та інформаційні технології: Зб. наук. пр.— Київ.— 2010.— № 58.— С. 142—150.
4. Kirichenko L., Radivilova T., Kayali E. Modeling telecommunications traffic using the stochastic multifractal cascade process // Problems of Computer Intellectualization.— Kiev–Sofia: ITHEA.— 2012.— P. 55–63.

Т. А. Radivilova, I. N. Ivanisenko, L. O. Kirichenko

The algorithm of self-similar load balancing

The paper presents a dynamic load balancing algorithm based on the self-similar structure of flows and monitoring state of elements of a distributed system. The authors compare algorithms for the key performance indicators: response time, resource usage, cost and performance. An analysis of the proposed algorithm showed that it provides a more uniform load distribution on the servers, high performance and low response time.

Keywords: *load balancing, self-similar traffic, distributed system.*