

УДК 004.75

Л.О. КИРИЧЕНКО, И.Н. ИВАНИСЕНКО, Т.А. РАДИВИЛОВА
Харьковский национальный университет радиоэлектроники

АНАЛИЗ ДИСБАЛАНСА РАСПРЕДЕЛЕННОЙ СИСТЕМЫ ПРИ САМОПОДОБНОЙ НАГРУЗКЕ

В работе предложен метод балансировки нагрузки с учетом оценивания загрузки узлов распределенной системы, который основан на расчете дисбаланса системы, при входных потоках данных, обладающих мультифрактальными свойствами. Проведено имитационное моделирование предложенного метода при работе различных алгоритмов балансировки, которое показало, что мультифрактальные характеристики трафика существенно влияют на дисбаланс системы. Использование предложенного метода позволяет распределять запросы по серверам так, чтобы отклонение загрузки серверов от среднего значения было минимальным, что позволяет получить более высокие показатели производительности системы и более быструю обработку потоков.

Ключевые слова: балансировка нагрузки, мультифрактальный поток, дисбаланс, оценка загрузки, распределенная система, самоподобие.

Л.О.КІРІЧЕНКО, І.М. ІВАНІСЕНКО, Т.А. РАДІВІЛОВА
Харківський національний університет радіоелектроніки

АНАЛІЗ ДИСБАЛАНСУ РОЗПОДІЛЕНОЇ СИСТЕМИ ПРИ САМОПОДІБНОМУ НАВАНТАЖЕННІ

У роботі запропоновано метод балансування навантаження з урахуванням оцінювання завантаження вузлів розподіленої системи, який засновано на розрахунку дисбалансу системи, при входних потоках даних, що володіють мультифрактальними властивостями. Проведено імітаційне моделювання запропонованого методу при роботі різних алгоритмів балансування, яке показало, що мультифрактальні характеристики трафіку істотно впливають на дисбаланс системи. Використання запропонованого методу дозволяє розподіляти запити по серверам так, щоб відхилення завантаженості серверів від середнього значення було мінімальним, що дозволяє отримати більш високі показники продуктивності системи і швидкість обробки потоків.

Ключові слова: балансування навантаження, мультифрактальний потік, дисбаланс, оцінка завантаження, розподілена система, самоподоба.

L. KIRICHENKO, I. IVANISENKO, T. RADIVILOVA
Kharkiv National University of Radio Electronics

ANALYSIS OF IMBALANCE OF DISTRIBUTED SYSTEM WITH A SELF-SIMILAR LOAD

In this paper a load balancing method based on estimating the load of distributed system nodes proposed, which is based on the calculation of the system imbalance, when the input data stream having multifractal properties. The simulation of the proposed method by using different load balancing algorithms conducted, it showed that characteristics of multifractal traffic significantly affect on the system imbalance. Using of proposed method allows to distribute requests across the servers so that the deviation of the load servers from the average value was minimal, which allows to get higher system performance metrics and faster processing flows.

Keywords: load balancing, multifractal flow, imbalance, imbalance, distributed system, self-similarity.

Постановка проблемы

В настоящее время вместе с увеличением скоростей передачи данных в телекоммуникациях увеличивается доля интерактивного трафика, крайне чувствительного к параметрам среды транспортировки. Для предоставления требуемого количества ресурсов при передаче различных видов трафика, предъявляющих различные требования к характеристикам телекоммуникационной сети используются различные механизмы обеспечения QoS (Quality of service). Одним из таких механизмов является балансировка нагрузки [1-3]. Система балансировки нагрузки решает задачи обеспечения качества обслуживания и повышения производительности распределенных систем за счет оптимального распределения заданий между узлами вычислительной системы.

Задача балансировки загрузки состоит в том, чтобы исходя из комплекта задач, включающих вычисления и передачу данных, и системы серверов различной ресурсоемкости, найти такое распределение задач по серверам, которое обеспечивает примерно равную вычислительную загрузку каждого сервера и минимальные затраты на передачу данных. Для выполнения этой задачи могут использоваться различные алгоритмы балансировки нагрузки, которые учитывают оценки загрузки вычислительного узла.

Наиболее разработкой и моделирование как King-Ming Z распределенной системы сетевого трафика имеет особую роль в некоторых случаях выбросы в потоках телекоммуникации D.V. Wilson, J. Self-similarity, ATM-сети, особенности сетевых сервисов самоподобия. вызывает необходимость для расчета времени. Такие нагрузки, которые Целью в данном потоке моделирования Для внешнего мониторинга Внешние Системы распределенной 1. Метод направляет управление сетью убедиться в готовности на выделении ответственной среды передачи загрузке узлов, 2. Метод более детально вычислительный 3. Метод узла и приложения между отправкой Основными вычислительными работами системы Недостатки а) данные которое может быть б) отсутствовать Внутренний Использовать телеком подходе и данные об узлах балансировщик

Анализ последних исследований и публикаций

Наиболее известными исследованиями в области балансировки, теоретических исследований и разработки фундаментальных основ распределения нагрузки, в создании математического аппарата, моделей и методов управления для распределения нагрузки в распределенных системах занимались такие ученые как Е.И. Игнатенко [3], В.Н. Тарасов [4], F. Wang [5], V. Cardellini [6], S.Keshav [7], Xing-Guo Luo, King-Ming Zhang [8], Hisao Kameda, Lie Li [9], а также другие исследователи, работающие над проблемами распределения нагрузки.

Многочисленные исследования процессов в информационных сетях показали, что реализации сетевого трафика обладают свойством масштабной инвариантности (самоподобием). Самоподобный трафик имеет особую структуру, сохраняющуюся на многих масштабах – в реализации всегда присутствует некоторое количество очень больших выбросов при относительно небольшом среднем уровне трафика. Эти выбросы вызывают значительные задержки и потери пакетов, даже когда суммарная потребность всех потоков далека от максимально допустимых значений. Исследованиями свойств самоподобного трафика в телекоммуникационных сетях занимались такие ученые как W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson, A.B. Лемешко [10], В.В. Турупалов, Е.Г. Игнатенко [3] и другие.

Самоподобные свойства обнаружены в локальных и глобальных сетях, в частности в трафиках Ethernet, ATM, приложений TCP, IP, VoIP и видеопотоках. Причина такого эффекта заключается в особенностях распределения файлов по серверам, их размерах, в типичном поведении пользователей. Оказалось, что изначально не проявляющие свойств самоподобия потоки данных, пройдя обработку на узловых серверах и активных сетевых элементах, начинают подавать ярко выраженные признаки самоподобия. Наличие у передаваемых клиентами информационных потоков свойств самоподобия оказывает большое влияние на эффективность работы распределенных систем. Особенно важную роль это играет для работы сервисов, обеспечивающих передачу мультимедийного трафика и трафика реального времени. Таким образом, актуальной является задача разработки и анализа алгоритма балансировки нагрузки, который учитывает самоподобие трафика и загрузку каждого узла и всей распределенной системы.

Цель работы

Целью данной работы является разработка метода балансировки нагрузки при самоподобном входном потоке с учетом загрузки узлов распределенной системы и анализ результатов имитационного моделирования при работе предложенного метода для различных алгоритмов балансировки.

Основная часть

Для получения информации о текущем состоянии вычислительного узла выделяют два подхода: внешний мониторинг состояния вычислительных узлов распределенной системы и внутренний мониторинг. Внешний мониторинг системы балансировки нагрузки.

Система балансировки централизованно собирает данные о состоянии всех вычислительных узлов распределенной системы. Существует несколько методов внешнего мониторинга.

1. Метод расчета системой балансировки нагрузки времени отклика вычислительного узла, для чего она направляет на узел служебный запрос и замеряет время ответа (например, используя протокол управления сообщениями Internet Control Message Protocol – ICMP). Данный метод позволяет системе убедиться в готовности вычислительного узла и определить количество времени, необходимое для передачи данных на вычислительный узел от системы балансировки нагрузки и обратно. Однако задержка в получении ответа может не зависеть от загрузки вычислительного узла. Здесь задержка может вносить среда передачи данных. Если нагрузка на сеть в распределенной системе неравномерна, то данные о загрузке узлов, основанные на времени ответа узла, могут быть сильно искажены.

2. Метод, основанный на отправке служебных пакетов подконтрольным узлам с целью получения более детальной информации о загрузке вычислительного узла (установление соединения с вычислительным узлом по протоколу TCP).

3. Метод, позволяющий обеспечивать мониторинг времени отклика и готовности вычислительного узла и приложений, работающих на нем [3]. Время отклика приложения определяется как интервал времени между отправкой запроса на предоставление данных и до момента объявления о готовности к передаче.

Основным преимуществом данного подхода является возможность учитывать производительность вычислительного узла и вводить весовые коэффициенты, которые можно динамически менять в процессе работы системы балансировки, что позволяет более гибко менять нагрузку на вычислительный узел.

Недостатки:

- а) данные о загрузке узла выражаются только временем ответа на запрос системы балансировки, которое может быть большим из-за перегрузок в сети, а не загрузки самого узла;
 - б) отсутствие данных о состоянии процессора, памяти, системы ввода/вывода и т.д.
- Внутренний мониторинг системы балансировки нагрузки.

Используется для получения более детальной информации о состоянии вычислительного узла. При таком подходе на каждый узел распределенной системы помещается программа-агент, которая собирает данные об узле, на котором она находится. Данные от программ-агентов могут передаваться на балансировщик либо после запроса центральной (диспетчерской) части системы балансировки нагрузки,

либо программы-агенты могут самостоятельно передавать сведения о загрузке подконтрольного узла на центральный узел через определенный интервал времени. При данном подходе существует проблема определения частоты передачи обработанной информации на балансировщик. Первый метод позволяет по запросу центрального узла получать данные от всех узлов в тот момент, когда необходимо поставить новый запрос клиента в очередь на исполнение. Второй метод позволяет постоянно иметь свежие данные о загрузке вычислительных узлов, но при этом повышается нагрузка на сеть из-за постоянной передачи служебных данных.

Основным недостатком данного подхода является использование служебными программами-агентами определенного количества ресурсов узла. Объем требуемых ресурсов узла программами-агентами зависит от того, по скольким параметрам идет оценка загруженности вычислительного узла [3, 4], идет ли обработка этих данных на самом узле или они передаются на центральную машину системы балансировки нагрузки. Поэтому при разработке системы балансировки нагрузки важным вопросом является определение критериев загрузки вычислительного узла. Потребуется большее количество ресурсов узла для обработки информации о загрузке компонентов (процессор, память) узла. Однако, оценка загрузки только одного компонента не дает объективных данных о состоянии всего узла. Поэтому в работе предложены критерии, учитывающие в совокупности несколько ресурсов узла.

Система балансировки состоит из активного балансировщика, резервного балансировщика и группы серверов [3, 11, 12, 13].

Критерии оценки загрузки распределенной системы.

1. Средняя загрузка каждого u -го процессора $CPU_i^u(T)$ i -го сервера определяется как средняя загрузка процессора в течение наблюдаемого периода T . Например, если период наблюдений составляет 1 мин., а загрузка процессора записывается через каждые 10 секунд, то есть CPU_i^u это среднее значение из шести записанных значений i -го сервера.

Аналогично определяется средняя загрузка каждой r -й памяти $RAM_i^r(T)$ i -го сервера и средняя загрузка k -го канала $Net_i^k(T)$ i -го сервера в течение наблюдаемого периода.

2. Введем средний коэффициент использования всех процессоров в системе. Пусть $CPU_i^{n_i}$ средняя загрузка ЦПУ i -го сервера,

$$CPU_u^{All} = \frac{\sum_i^N CPU_i^u CPU_i^{n_i}}{\sum_i^N CPU_i^{n_i}}, \quad (1)$$

где N – общее число физических серверов в системе, n_i – количество ЦПУ на i -м сервере.

Аналогичным образом, средний коэффициент использования памяти $RAM_i^{m_i}$, пропускной способности линии связи Net_i^k i -го сервера, вся память RAM_r^{All} , и вся пропускная способность сети в системе Net_k^{All} может быть определена.

$$RAM_r^{All} = \frac{\sum_i^N RAM_i^r RAM_i^{m_i}}{\sum_i^N RAM_i^{m_i}}, \quad (2)$$

$$Net_k^{All} = \frac{\sum_i^N Net_i^k Net_i^{k_i}}{\sum_i^N Net_i^{k_i}}, \quad (3)$$

3. Значение дисбаланса всех процессоров. Используя формулу дисперсии, значение дисбаланса всех процессоров в системе определяется как

$$ISL_{CPU} = \sum_i^N (CPU_i^u - CPU_u^{All})^2, \quad (4)$$

Точно так же могут быть рассчитаны значения дисбаланса памяти и пропускной способности сети.

$$ISL_{RAM} = \sum_i^N (RAM_i^r - RAM_r^{All})^2, \quad (5)$$

$$ISL_{Net} = \sum_i^N (Net_i^k - Net_k^{All})^2, \quad (6)$$

4. Введем комплексное значение дисбаланса нагрузки SIL_i i -го сервера, учитывающее все три ресурса сервера. Используя формулу расчета дисперсии как меры неравномерности, интегрированное значение дисбаланса нагрузки i -го сервера можем определить как:

$$SIL_i = a(CPU_i^u - CPU_u^{All})^2 + b(RAM_i^r - RAM_r^{All})^2 + c(Net_i^k - Net_k^{All})^2, \quad (7)$$

Параметры a, b, c обозначают весовые коэффициенты для процессора, памяти и пропускной способности сети, соответственно, которые выбираются экспериментальным путем таким образом, что $a + b + c = 1$, в зависимости от решаемых задач и структуры системы.

SIL_i применяется для обозначения уровня дисбаланса нагрузки путем сравнения коэффициентов использования процессора, памяти и пропускной способности сети.

5. Тогда суммарные значения дисбаланса всех серверов в системе записывается как:

$$ISL_{tot} = \frac{1}{N} \sum_i^N SIL_i, \quad (8)$$

6. Эффективность использования определяется как средняя нагрузка на любом сервере.

Таким образом, для планирования ресурсов была разработана методика комплексного измерения общего уровня дисбаланса системы, а также среднего уровня дисбаланса каждого сервера.

Самоподобные и мультифрактальные свойства трафика. Самоподобие случайных процессов заключается в сохранении статистических характеристик при изменении масштаба времени и характеризуется показателем Херста H , который является степенью самоподобия. Стохастический процесс $X(t)$ является статистически самоподобным, если процесс $a^{-H} X(at)$ обладает теми же статистическими характеристиками второго порядка, что и $X(t)$. Параметр H , называемый параметром Херста, представляет собой меру самоподобия стохастического процесса. Моменты самоподобного случайного процесса можно выразить как $M[X(t)^q] = C(q) \cdot t^{qH}$, где величина $C(q)$ – некоторая детерминированная функция. Для мультифрактальных процессов выполняется отношение $M[X(t)^q] = c(q) \cdot t^{qh(q)}$, где $c(q)$ – детерминированная функция; $h(q)$ – обобщенный показатель Херста, являющийся в общем случае нелинейной функцией. Значение $h(q)$ при $q = 2$ совпадает со значением степени самоподобия H [14].

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.

В работе характеристикой неоднородности мультифрактального потока данных предложено считать диапазон обобщенного показателя Херста $\Delta h = h(q_{min}) - h(q_{max})$. Для монофрактальных процессов обобщенный показатель Херста не зависит от параметра q и является прямой линией: $h(q) = H$, $\Delta h = 0$. Чем больше неоднородность процесса, т.е. чем большие выбросы присутствуют в трафике, тем больше диапазон Δh .

Метод балансировки с учетом мультифрактальных свойств трафика. Рассмотрим метод балансировки, который учитывает мультифрактальные свойства трафика. Система балансировки нагрузки состоит из компонентов, представленных на рис. 1. Балансировщик нагрузки имеет веб-интерфейс для отслеживания, настройки и администрирования распределения нагрузки.

Предлагаемый метод балансировки нагрузки основан на использовании комплекса методов внутреннего и внешнего мониторинга. Использование внешнего мониторинга системы позволяет периодически тестировать сеть для определения наиболее загруженных сегментов. Использование внутреннего мониторинга состояния вычислительного узла позволяет получить объективную картину загрузки узла и данные о загрузке отдельных компонентов узла. Критериями загрузки вычислительного узла предлагается считать загрузку процессора, памяти и пропускной способности канала для потоков разных классов обслуживания. Чтобы учитывать потенциальную вычислительную мощность каждого ресурса узла, предлагается использовать весовые коэффициенты. Данный метод учитывает мультифрактальные свойства входящего информационного потока. Использование комплексного подхода при балансировке нагрузки позволит добиться увеличения производительности распределенной вычислительной системы.

Процесс *pulse* – это управляющий процесс, который на активном балансировщике запускает процесс *lvs*, а на резервном балансировщике будет отслеживать состояние активного, периодически спрашивая его. Если активный балансировщик не отвечает в течение заданного периода времени, будет инициирован процесс передачи его функций резервному балансировщику. При этом *pulse* на резервном балансировщике отправляет процессу *pulse* на активном балансировщике команду остановки всех служб *lvs*, запускает *send_arp* для присвоения виртуальных IP-адресов MAC-адресу резервного балансировщика и запускает процесс *lvs*. Процесс *send_arp* рассылает широковещательные пакеты ARP при переходе виртуального IP-адреса от одного узла другому [15].

Процес *lvs* запускається на активному балансировщике по вызову *pulse*. Он вызывает службу *ipadm* для создания, добавления, изменения и удаления записей в таблице маршрутизации IP. Процесс *lvs* запускает процесс *status* для каждой настроенной службы распределения нагрузки. Если *status* сообщает о том, что реальный сервер отключен, *lvs* заставит утилиту *ipadm* удалить этот сервер из таблицы IP. Основным назначением процесса *status*, работающего на активном балансировщике, является наблюдение за нагрузкой серверов, сбор и анализ статистики о текущем состоянии системы и интенсивности трафика. Процесс *status* передает обработанную и проанализированную информацию в подсистему балансировки нагрузки процессу *inf_status*. Процесс *inf_status* в свою очередь передает информацию о текущем состоянии системы процессу *coun_parce*.

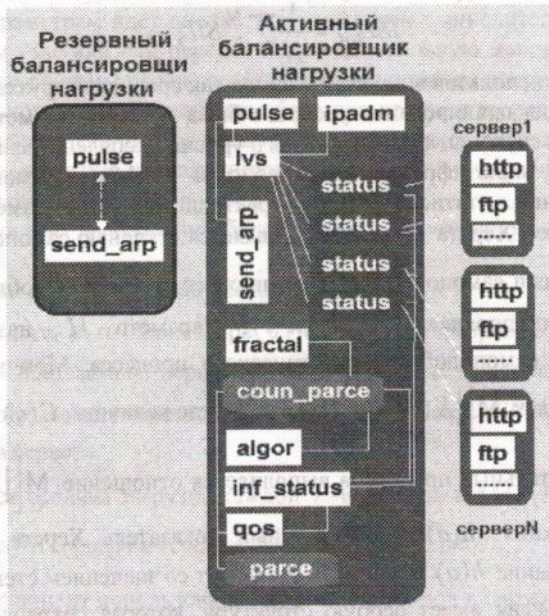


Рисунок 1. Компоненты системы балансировки нагрузки

Процесс *fractal* проводит расчет мультифрактальных свойств каждого входящего потока данных и передает эту информацию процессу *coun_parce*. Процесс *algor* выбирает алгоритм балансировки нагрузки и также передает эту информацию процессу *coun_parce*. Процесс *coun_parce* проводит расчет распределения потоков по узлам сети с учетом классификации трафика, загруженности серверов и каналов связи, дисбаланса серверов и всей системы. Результаты расчета передаются процессу *parce*, который осуществляет динамическое распределение трафика по различным каналам связи и узлам в зависимости от их текущего состояния. Также результаты расчета процесса *coun_parce* передаются процессу *qos* для гибкой настройки обеспечения качества обслуживания в соответствии с методами управления трафиком (управление пропускной способностью каналов и памяти, производительностью процессоров, кеш-памяти), если это необходимо. В случае применения методов управления трафиком процессом *qos*, информация об изменениях передается процессу *inf_status*.

Результаты имитационного моделирования. Был разработан программный продукт, который позволяет проводить имитационное моделирование работы системы балансировки нагрузки с помощью различных алгоритмов балансировки, используя предложенный метод. Рассмотрены три основных алгоритма балансировки нагрузки: Round Robin, Round Robin Weight и Compare Balance. Алгоритм Round Robin (кругового обслуживания), представляет собой перебор по круговому циклу: первая задача передается одному узлу, затем следующая задача передается другому и так до достижения последнего узла, а затем всё начинается сначала. В алгоритме Round Robin Weight каждому узлу присваивается весовой коэффициент в соответствии с его производительностью и мощностью. Это помогает распределять нагрузку более гибко, так как узлы с большим весом обрабатывают больше запросов. Алгоритм Compare Balance используется для достижения равновесного состояния и управления несбалансированной нагрузкой системы. В этом алгоритме балансировщик случайным образом выбирает сервера и сравнивает их нагрузку, после чего отправляет заявку на обработку сервера с меньшей нагрузкой.

Для анализа работы алгоритмов балансировки были проведены численные исследования работы системы балансировки при различных значениях нагрузки, количества серверов и параметров мультифрактального трафика. На рис. 2-4 показано изменение дисбаланса процессоров, памяти и пропускной способности каналов (4-6) в случае 2-х серверов, а также комплексное значение дисбаланса нагрузки каждого сервера (7) при работе алгоритма балансировки Compare Balance. Параметры *a, b, c*, которые обозначают весовые коэффициенты для процессора, памяти и пропускной способности сети, были выбраны равнозначными.

В:
 $H = 0.6$ и
 для трафик
 первом сл
 больших z
 $\Delta t = 2.5$.
 Параметр
 Потерян
 Среднее
 ожидани
 Ис
 характерис
 приходит
 Херста с
 равновесно
 балансирое

службу ipadm
Процес *lvs*
іs повідомляє о
таблиці IP.
аблюдення за
сти трафіка.
балансировки
ем состоянии

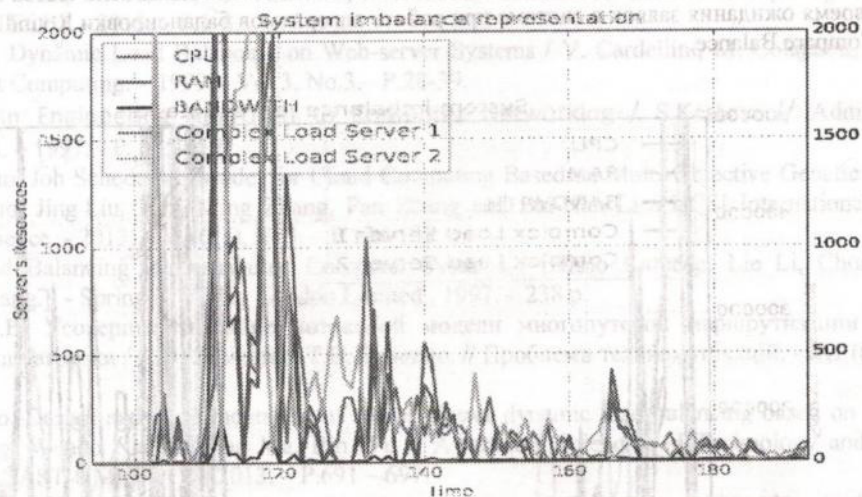


Рисунок 2. Дисбаланс системы при параметрах трафика $H = 0.6$ и $\Delta h = 1.5$.

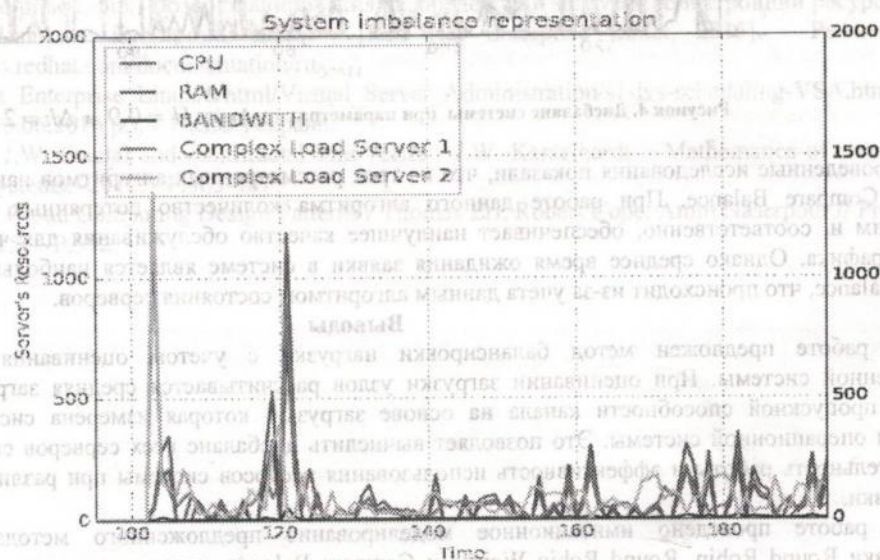


Рисунок 3. Дисбаланс системы при параметрах трафика $H = 0.9$ и $\Delta h = 1.5$.

ока данных и
ки нагрузки и
аспределения
каналов связи,
осуществляет
их текущего
ой настройке
(управление
ги), если это
формация об
кт, который
с помощью
и основных
оритм Round
на передаётся
а, а затем всё
эффицент в
более гибко,
льзуется для
мы. В этом
; после чего
ания работы
параметров
з, памяти и
е дисбаланса
етры a, b, c ,
и сети, были

В первом случае (рис. 2) на балансировщик поступает сгенерированный трафик с параметром $H = 0.6$ и диапазоном обобщенного показателя Херста $\Delta h = 1.5$. Рис. 2 демонстрирует дисбаланс системы для трафика с более сильной долгосрочной зависимостью ($H = 0.9$) и той же неоднородностью, что и в первом случае. На рис. 4 представлены результаты моделирования работы балансировщика в случае больших значений как параметра Херста $H = 0.9$, так и диапазона обобщенного показателя Херста $\Delta h = 2.5$.

Таблица 1

Сравнительные характеристики алгоритмов балансировки

Параметры	Round Robin	Round Robin Weight	Compare Balance
Потерянные данные, %	1,94	1,64	0,98
Среднее время ожидания, мс	4	6	8

Исследования показали, что дисбаланс системы существенно зависит от мультифрактальных характеристик трафика. При небольших значениях H и небольшой неоднородности система балансировки приходит в равновесное состояние и значение дисбаланса стремится к нулю. При увеличении показателя Херста с течением времени дисбаланс системы не затухает и система балансировки не приходит в равновесное состояние. При больших значениях показателя Херста и большой неоднородности система балансировки находится в неустойчивом состоянии и значение дисбаланса изменяется в несколько раз, что

приводит к максимальной загрузке ресурсов. В табл. 1 приведены значения количества потерянных данных и среднее время ожидания заявок в системе при работе алгоритмов балансировки Round Robin, Round Robin Weight и Compare Balance.

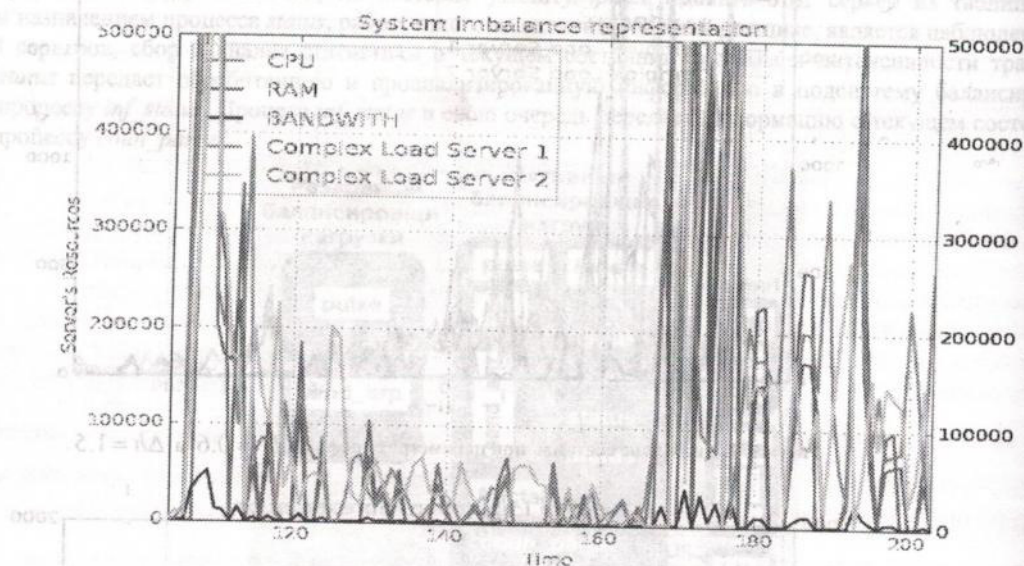


Рисунок 4. Дисбаланс системы при параметрах трафика $H = 0.9$ и $\Delta h = 2.5$

Проведенные исследования показали, что из трех рассмотренных алгоритмов наилучшим является алгоритм Compare Balance. При работе данного алгоритма количество потерянных данных является наименьшим и, соответственно, обеспечивает наилучшее качество обслуживания для чувствительного к потерям трафика. Однако среднее время ожидания заявки в системе является наибольшим у алгоритма Compare Balance, что происходит из-за учета данным алгоритмом состояния серверов.

Выводы

В работе предложен метод балансировки нагрузки с учетом оценивания загрузки узлов распределенной системы. При оценивании загрузки узлов рассчитывается средняя загрузка процессора, памяти и пропускной способности канала на основе загрузки, которая измерена системой учета или монитором операционной системы. Это позволяет вычислить дисбаланс всех серверов системы, среднюю продолжительность работы и эффективность использования ресурсов системы при различных алгоритмах балансировки.

В работе проведено имитационное моделирование предложенного метода и алгоритмов балансировки Round Robin, Round Robin Weight и Compare Balance, которое показало, что наименьшее количество потерянной информации получается при работе динамического алгоритма балансировки, а наименьшее время отклика - при статическом. Результаты моделирования показали, что мультифрактальные характеристики трафика существенно влияют на дисбаланс системы. При небольших значениях показателя Херста и небольшой неоднородности трафика значение дисбаланса стремится к нулю и система балансировки приходит в равновесное состояние. При больших значениях показателя Херста и неоднородности система балансировки постоянно находится в неустойчивом состоянии, что приводит к максимальной загрузке ресурсов. Использование предложенного метода при балансировке нагрузки, с учетом информации о состоянии серверов и всей системы, позволяет балансировщику выделить сервер, который способен наилучшим образом справиться с обработкой пришедшего мультифрактального потока задач.

Список использованной литературы

1. Чжоу Т. Системы балансировки нагрузки Web-серверов / Тао Чжоу // Журнал «Windows 2000 Magazine». – 2000. – № 03/2000.
2. Воеводин В.В. Параллельные вычисления / В.В. Воеводин // СПб: БХВ-Петербург. - 2003.
3. Игнатенко Е.И. Адаптивный алгоритм мониторинга загруженности сети кластера в системе балансировки нагрузки / Е.И. Игнатенко, В.И. Бессараб, И.В. Дегтяренко // Наукові праці ДонНТУ. – 2011. – Вип.21(183). – С.95-102.
4. Тарасов В.Н. Математические модели облачного вычислительного центра обработки данных с использованием Openflow / В.Н. Тарасов, П.Н. Полежаев, А.Е. Шухман, Ю.А. Ушаков, А.Л. Коннов // ВЕСТНИК ОГУ. – 2012. – №9 (145). – С.150-155.
5. Wang F. User-priority guided min-min scheduling algorithm for load balancing in cloud computing

Редакційна рада

ISSN 2078-4481

Міністерство освіти і науки України
Херсонський національний технічний університет

ВІСНИК

Херсонського національного
технічного університету



Рекомендовано до друку Вченою радою
Херсонського національного технічного університету
(протокол № 7 від 30 червня 2016 року)

Журнал включено до Переліку наукових фахових видань України
(Наказ МОН України № 820 від 11.07.2016 р.),
у яких можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів
доктора та кандидата технічних наук

Журнал включено до наукометричних баз, електронних бібліотек та репозитаріїв:
Національна бібліотека імені В.І. Вернадського, Google Scholar, РИНЦ (російський індекс
научного цитування)

Херсон 2016

Випуск

Вісник Херсонського національного технічного університету. Вип. 3(58). –
Херсон: ХНТУ, 2016. – 566 с.

Використання програмних засобів моделювання та розробки електронних пристроїв з використанням спеціальних пакетів програмного забезпечення дозволяють оптимізувати параметри електронних пристроїв, зокрема, розробки мікроконтролерних систем управління можуть з використанням спеціальних пакетів програмного забезпечення характеризувати особливості нелинійної управляючої характеристики. В результаті цього дозволять з більшою точністю стабілізувати потужність формуючого електронного пучка при роботі пучок БІІ в складі технологічного обладнання, в, тем самым, значительно повысить качество готовых изделий и снизить процент брака на производстве.

Список використаної літератури

1. Дідошки С.В., Левинский И.И., Чернышовой В.В. и др. Электронно-лучевая плазма в дуговом разряде. – Киев: «Судол», 2007. – 603 с.
2. Иванков А.А. Источники электронов высоковольтного тлеющего разряда. – М.: Энергоатомиздат, 1983. – 96 с.
3. Денисовський С.В., Костюченко К.О., Мельник В.І., Мельник Г.В., Тутай В.А. Моделирование процесса автоматического управления струей тлеющего разряда электровакуумного нагревателя. – Электроника та зв'язок, том 9, № 22, 2004. С.15–17.
4. Melnyk V. I., Melnyk G. V., Tutaev V. A. Model of control of glow discharge electron gun current for microelectronical devices. // Proceedings of SPIE, Sixth International Conference on "Material Science and Material Properties for Infrared Optoelectronics". – Vol. 5065 (2003).
5. Костюченко К.О., Мельник В.І., Мельник Г.В., Тутай В.А. Моделирование процесса автоматического управления струей тлеющего разряда электровакуумного нагревателя. // Микросистемная электроника. – М.: Издательский дом «Долгострой», 2008. – С. 15–17.
6. Мельник В.В. Алгоритмы автоматического управления технологических источников электронов высоковольтного тлеющего разряда с использованием средств системы Matlab // Вісник Херсонського національного технічного університету. – 2009. – № 2 (35). – С. 299–305.
7. Мельник В.В., Тутай В.В. Исследование возможности детального устройства нагревателя газа для газоразрядной электронной пушки. // Электроника и связь. № 5 (64), 2011. – С. 23–30.
8. Васильев В.П. Численные методы решения экстремальных задач. – М.: Наука, 1985. – 152 с.

Відповідальний за випуск

д.ф.-м.н., професор

Хомченко Анатолій Никифорович

Макетування, технічна
редакція

к.т.н.

Омельчук Антон Анатолійович

Типографські роботи

Грінь Дмитро Сергійович

Матеріали друкуються в авторській редакції.

Відповідальність за достовірність даних, зазначених у статтях,
несуть їх автори.

Підписано до друку 02.08.2016

Формат 60x84/8. Папір Офс.

Ум. арк. 66,03. Наклад 300 примірників.

Видання та друк: ФОП Грінь Д.С.,

73033, м. Херсон, а/с 15

e-mail: dimg@meta.ua

Свід. ДК № 4094 від 17.06.2011

© Херсонський національний технічний університет, 2016