

Література

1. Ушаков К.З. Газовая динамика шахт. – 2-е изд., перераб и доп. / Ушаков К.З. – М. : Издательство Московского государственного горного университета, 2004. – 481 с.

Анотація

Розглядаються основні статичні характеристики шахтних газодинамічних процесів: закони розподілу, кореляційна функція, стаціонарність та нерівномірність процесу газовиділення.

Аннотация

Рассматриваются основные статические характеристики шахтных газодинамических процессов: законы распределения, корреляционная функция, стационарность и неравномерность процесса газовыделения.

Abstract

Discusses the basic static characteristics of coal mine gas-dynamic processes: the laws of distribution, correlation function, stationarity and irregularity of the process of gassing.

АЛГОРИТМЫ БАЛАНСИРОВКИ НАГРУЗКИ

*Хаст В.Д., ст. гр. ТКС-12, mr.khast@gmail.com;
Ступак Г.В., ст. пр. каф. АТ, e-mail, stupakgv@gmail.com
Донецкий национальный технический университет,
г.Красноармейск, Украина*

Высокие темпы роста сетевого трафика, а следовательно и нагрузки на центры обработки данных, приводят к необходимости изучения и анализа существующих алгоритмов балансировки нагрузки с целью исследования их эффективности, возможностей их применения и оптимизации.

Исходя из статистических прогнозов компании Cisco [1], темпы роста сетевого трафика составляют 150% в год. В своем пятом ежегодном докладе Cisco Global Cloud Index (2014–2019) компания продемонстрировала современные объемы облачного трафика, а также трафик центров обработки данных, которые в настоящее время составляют 2,1 и 3,4 ЗБ соответственно. В Cisco составили прогноз роста глобального трафика по обоим направлениям и заявили, что к концу 2019 года объем облачного трафика возрастет вчетверо и составит 8,6 ЗБ, а трафик центров обработки данных, который за этот же период станет втрое больше по сравнению с текущим моментом, составит 10,4 ЗБ.

Прогнозы темпов роста трафика подчеркивают необходимость поиска и разработки эффективных методов его балансировки и управления. Разделяют аппаратную и программную балансировку. Аппаратная балансировка является эффективным, но дорогостоящим и низко-мобильным решением. Исходя из этого, программная балансировка нагрузки является более доступным и универсальным решением с широким применением.

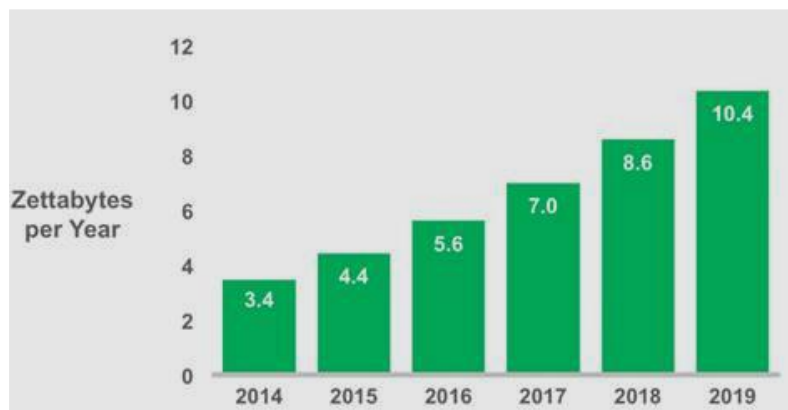


Рисунок 1. Прогнозируемый рост объема трафика

Программная балансировка осуществляется при помощи специальных алгоритмов на определенных уровнях модели OSI:

- сетевом;
- транспортном;
- прикладном.

Балансировка на *сетевом уровне* предполагает соответствие одному конкретному IP-адресу сервера разных физических машин и может осуществляться с помощью нескольких методов:

- DNS-балансировка;
- объединение серверов в балансировочный кластер;
- балансировка по территориальному признаку.

Балансировка на *транспортном уровне* предполагает, что клиент обращается к балансировщику, который по определенному алгоритму определяет конкретный сервер, и перенаправляет на него запрос.

В отличие от сетевого уровня, где балансировщик просто распределяет пакеты между серверами, а взаимодействие между клиентом и сервером происходит непосредственно, на транспортном уровне балансировщик сам взаимодействует с сервером от своего имени, передавая информацию о клиенте.

При балансировке на *прикладном уровне* балансировщик также выступает посредником между клиентом и сервером, анализирует клиентские запросы и перенаправляет их на разные серверы по определенным алгоритмам в зависимости от характера этих запросов.

Существует много различных алгоритмов и методов для балансировки нагрузки, каждый алгоритм обладает своей спецификой, имеет свои недостатки и не является полностью универсальным [2]. Выбирая алгоритм, нужно исходить, во-первых, из специфики конкретной задачи, а во-вторых — из целей, которые необходимо достичь. Определяя цели, стоит опираться на следующие параметры:

- справедливость;
- эффективность;
- сокращение времени выполнения запроса;
- сокращение времени отклика.

Очень важно также, чтобы алгоритм балансировки обладал свойством предсказуемости (чёткое понимание, в каких ситуациях алгоритм будет эффективным для решения поставленных задач), и при этом был масштабируемым (алгоритм должен сохранять работоспособность и эффективность при росте нагрузки).

Рассмотрим некоторые наиболее распространенные алгоритмы.

Round Robin (алгоритм кругового обслуживания) представляет собой перебор серверов по кругу. Первый запрос передается первому серверу, следующий запрос передаётся второму, и так далее до последнего сервера, после чего перебор начинается сначала.

У этого алгоритма имеется ряд достоинств, однако обратим внимание на его недостатки. Чтобы распределение нагрузки по этому алгоритму отвечало критериями справедливости и эффективности, нужно, чтобы у каждого сервера был равный набор ресурсов. Это труднодостижимое условие. Кроме того, при балансировке по алгоритму *Round Robin* не учитывается загруженность сервера и количество активных на данный момент подключений. Исходя из этого, на практике используется усовершенствованная версия алгоритма — *Weighted Round Robin* (круговое обслуживание с весовыми коэффициентами). Суть усовершенствований заключается в назначении каждому серверу весового коэффициента в соответствии с его производительностью. Это помогает распределять нагрузку эффективнее, алгоритм распределяет запросы таким образом, что серверы с большим весом обрабатывают больше запросов. Однако всех проблем это не решает, так как для более эффективной балансировки необходимо учитывать намного больше параметров.

Алгоритм *Least Connections* позволяет решить проблему, когда к одному серверу обращается меньшее количество пользователей, однако этот сервер оказывается более загруженным, чем другой (с большим количеством пользователей) за счет долгих активных подключений. Алгоритм учитывает количество подключений, поддерживаемых серверами в текущий момент времени. Каждый следующий запрос передаётся серверу с наименьшим количеством активных подключений. Различные усовершенствованные варианты этого алгоритма включают весовые коэффи-

циенты серверов (Weighted Least Connection); объединение клиентов в группы, обращающихся к определенному серверу (Locality-Based Least Connection Scheduling); объединение клиентов в группы, обращающихся к конкретным группам серверов (Locality-Based Least Connection Scheduling with Replication Scheduling).

Алгоритм Least Connections и его модификации позволяет решить некоторые проблемы неравномерной нагрузки, однако наименьшее количество активных соединений далеко не всегда означает наименьшую загрузженность сервера [3]. Это является весомым недостатком алгоритма.

В алгоритме *Destination Hash Scheduling*, сервер, обрабатывающий запрос, выбирается из статической таблицы по IP-адресу получателя. Алгоритм *Source Hash Scheduling* основывается на тех же самых принципах, но сервер, который будет обрабатывать запрос, выбирается из таблицы по IP-адресу отправителя.

Эти алгоритмы могут работать достаточно эффективно, однако создание правильной и оптимальной структуры hash-таблиц часто сопряжено с рядом проблем и является узким местом этих алгоритмов.^[4]

С помощью алгоритма *Sticky Sessions* запросы распределяются по серверам на основе IP-адреса клиента. Запросы одного и того же клиента будут передаваться на один и тот же сервер. Если закреплённый за конкретным адресом сервер недоступен, запрос будет перенаправлен на другой сервер. Алгоритм также может включать весовые коэффициенты серверов. Однако применение этого алгоритма также сопряжено с некоторыми проблемами. Если клиент использует динамический IP, часто возникают проблемы с привязкой сессий. Во избежание подобных проблем необходимо использовать cookies. В ситуации, когда большое количество запросов проходит через один прокси-сервер, балансировка не является эффективной и справедливой.

Как видно, существует большое количество алгоритмов балансировки, которые находят то или иное применение. Исходя из высокого темпа роста нагрузки, очевидно, что в ближайшее время остро станет рассмотрение вопроса и изучение возможностей оптимизации существующих алгоритмов (а при необходимости — создания новых усовершенствованных моделей), анализ их эффективности и тестирование их работы на смоделированных и реальных нагрузках центра обработки данных. Для эффективной реализации поставленных задач необходим доступ к статистике трафика центра обработки данных. Приветствуется заинтересованность и помощь в изучении и реализации поставленных задач.

Литература

1. Cisco Global Cloud Index: Forecast and Methodology, 2014–2019 White Paper [Электронный ресурс] — режим доступа: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html

2. Selectel. Балансировка нагрузки: основные алгоритмы и методы [Електронний ресурс] — режим доступа: <https://blog.selectel.ru/balansirovka-nagruzki-osnovnye-algoritmy-i-metody/>

3. F5 Network Inc. technical article. Back to Basics: Least Connections is Not Least Loaded [Електронний ресурс] — режим доступа: <https://devcentral.f5.com/articles/back-to-basics-least-connections-is-not-least-loaded>

4. Wikibooks. Data Structures/Hash Tables. Problems with hash tables [Електронний ресурс] — режим доступа: https://en.wikibooks.org/wiki/Data_Structures/Hash_Tables#Problems_with_hash_tables

Анотація

Розглянуто прогнози темпів зростання мережевого трафіку і навантажень на центри обробки даних, основні методи і алгоритми балансування навантаження, їх недоліки. Поставлені задачі щодо вивчення можливостей оптимізації алгоритмів балансування.

Ключові слова: алгоритми балансування навантаження, центр обробки даних, обсяги трафіку.

Аннотация

Рассмотрены прогнозы темпов роста сетевого трафика и нагрузок на центры обработки данных, основные методы и алгоритмы балансировки нагрузки, их недостатки. Поставлены задачи по изучению возможностей оптимизации алгоритмов балансировки.

Ключевые слова: алгоритмы балансировки нагрузки, центр обработки данных, объемы трафика.

Abstract

Considered forecasts growth of network traffic and data center loading, basic load balancing methods and algorithms. Tasked to consider the possibilities of load balancing algorithms optimization.

Keywords: load balancing algorithms, data center, traffic.

ОБҐРУНТУВАННЯ СТРУКТУРИ ВИМІРЮВАЛЬНОГО ПРИБОРУ КОНЦЕНТРАЦІЇ МІДІ У СТІЧНИХ ВОДАХ

Прокопчук А.В., магістрант, prokopchuk.alina@yandex.ua;

Вовна О.В., к.т.н., доцент, Vovna_Alex@ukr.net

Державний вищий навчальний заклад «Донецький національний технічний університет», м. Красноармійськ, Україна

Найбільша екологічна катастрофа це забруднення природних вод. Використана людиною вода повертається до природного середовища. Але це вже не чиста вода, а побутові, промислові та сільськогосподарські стічні води, зазвичай неочищені або очищені недостатньо. Таким чином відбувається забруднення прісноводних водоймищ – річок, озер, суші та прибережних ділянок морів. До одних з основних і найбільш небезпечних