



ИССЛЕДОВАНИЕ ЗАДАЧ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ ДОПУСТИМЫХ НАГРУЗОК НА СЕРВЕРЕ

ГВОЗДИНСКИЙ А.Н., СТОЛЯРЕНКО К.С.

Для минимизации возможных сбоев и корректной работы с данными исследуется алгоритм, с помощью которого решаются задачи для распределения допустимых нагрузок на сервере.

1. Введение

Актуальность исследования. Последние десятилетия XX века отмечены событиями, существенным образом трансформировавшими современную технологическую реальность. Речь идет об активном вхождении в жизнь общества новейших информационных технологий, произошедшем в результате бурного развития электроники, а также необходимости использования автоматизации бизнес-процессов как в коммерческой, так и некоммерческой деятельности предприятия. Внедрение IT-технологий и автоматизация рабочих мест кадров необходима для того, чтобы всегда оставаться конкурентоспособным на рынке как внешнем, так и внутреннем. Внедрение таких технологий и их использование дают массу преимуществ. Это может быть и быстрый поиск информации во внутренних базах или хранилищах, и исключение дублирования информации. К преимуществам можно отнести:

- увеличение объемов обработки информации;
- снижение времени на ее обработку;
- повышение качества обработки;
- автоматическое создание отчетов.

Все это автоматизируется с помощью серверов. Слово «сервер» произошло от английского глагола *serve*, что означает «служить, обслуживать», т.е. серверные устройства обслуживают пользователей и являются аппаратными устройствами и программными средствами. Как аппарат сервер представляет собой компьютер, который обслуживает другие компьютеры, а также принтеры, факсы и иные технические средства. Пользователи формируют для сервера задачи, а он их быстро и без ошибок решает с помощью определенных методов. В наше время невозможно себе представить работу банков, различных предприятий и научно-исследовательских учреждений без объединения информационных ресурсов в единую сеть, т.е. без сер-

веров. Даже факс можно отправить с помощью компьютера, а следовательно, и с помощью сервера. Поэтому нужно понимать всю важность и ценность серверов, их возможности и преимущества, а также различные назначения.

На любом предприятии используют серверы, создают многопользовательские центры, которые объединяют информацию всех работников-пользователей, что обеспечивает моментальный и очень удобный доступ к ней. Сервер обрабатывает огромный массив данных и предоставляет обмен информацией между всеми участниками одного проекта. Работа в учреждении или на предприятии становится более эффективной и слаженной, так как увеличивается скорость выполнения задач и повышается ее надежность. Соответственно уменьшается количество ошибок и просчетов. С помощью серверов можно объединить также и материальные ресурсы, такие как факсы, принтеры и др. Поэтому понятно, что все это приводит только к огромной экономии времени и денег. Благодаря серверу мы общаемся по электронной почте, в сбербанках сотрудники распечатывают нам документы, а в супермаркетах кассиры очень быстро нас обслуживают. Понятно, что совокупность серверов, которые обеспечивают доступ к удаленным собраниям ресурсов, дает такое многообразие наполнения Интернета. Все сайты в Интернете хранятся на серверах. А доступ к серверам предоставляют хостинговые компании. Все это называется виртуальным хостингом. Безусловно, чем надежнее, быстрее и безопаснее работает сервер, тем дороже обходится и серверное оборудование, а также программное обеспечение. Поэтому сервер является очень выгодным компонентом при работе в сети. Статистика гласит о том, что с каждым годом Интернет разрастается все быстрее и быстрее. В 2012 году web вырос более чем в два раза: год начался с 330 млн зарегистрированных веб-сайтов, а к декабрю эта цифра возросла к 743 млн. Сайт «internetlivestats.com» в режиме онлайн фиксирует каждый новый сайт, блог сообщение и твиты, которые появляются во всемирной паутине. Количество созданных сайтов к 2016 году перевалило за 1 млрд.

Сущность исследования заключается в том, чтобы найти оптимальное распределение допустимых нагрузок серверных приложений или служб на сервере, учитывая критерии передачи данных для точной их передачи и хранения.

Цели исследования заключаются в минимизации числа допустимых ошибок на сервере, а также минимизации времени и затрат на обработку запросов приложений (служб); распределении допустимых нагрузок серверных приложений (служб) между *m* серверами компьютерной сети. Для достижения данных целей необходимо решить задачи анализа алгоритмов оптимизации, выбрать конкретный алгоритм, который бы мог справиться с проблемой.

Индустрия электронной коммерции продолжает развиваться, и сегодня все новые компании начинают

общаться со своими клиентами с помощью Web. Высокопроизводительный сайт, способный предоставлять материалы быстро и без сбоев, не только помогает привлекать новых клиентов, но и становится важнейшей предпосылкой успешной деятельности предприятий электронной торговли.

2. Постановка задачи и математическая модель

Предприятие, работа которого зависит от компьютеров и компьютерной сети, нуждается в наличии серверов, для которых нужно определить оптимальность их работы. Каждый сервер имеет характеристики производительности. При обнаружении неполадки одного из серверов выходом из ситуации будет взаимозаменяемость их при обработке сетевых приложений (служб). Таким образом, для того, чтобы выполнить задачу, нужно составить план распределения приложений (служб) на серверы с заданными ограничениями при условии:

- минимум времени на обработку запросов;
- минимум затрат на обработку запросов приложений (служб);
- минимум ошибок.

Исходя из общей математической модели задачи распределения нагрузок, с учетом всех особенностей поставленной перед нами задачи построим конкретную математическую модель, с помощью которой можно будет вычислить экстремумы по трем условиям:

- минимум времени на обработку запросов:

$$F(x) = 60x_1 + 25x_2 + 15x_3 \rightarrow \min \quad (1)$$

при условиях:

$$0.3x_1 + 0.5x_2 + 0.4x_3 \leq 300; \quad (2)$$

$$0.5x_1 + 0.7x_2 + 0.8x_3 \leq 800; \quad (3)$$

$$1.3x_1 + 0.1x_2 + 1.2x_3 \leq 200; \quad (4)$$

$$x_1 \geq 500; x_2 \geq 300; \quad (5)$$

- минимум затрат на обработку запросов приложений (служб):

$$F(x) = 55x_1 + 30x_2 + 8x_3 \rightarrow \min \quad (6)$$

при условиях:

$$0.8x_1 + 0.5x_2 + 0.3x_3 \leq 250; \quad (7)$$

$$0.3x_1 + 0.5x_2 + 0.6x_3 \leq 500; \quad (8)$$

$$0.8x_1 + 0.1x_2 + 1.1x_3 \leq 150; \quad (9)$$

$$x_1 \geq 400; x_2 \geq 150; \quad (10)$$

- минимум ошибок:

$$F(x) = 55x_1 + 30x_2 + 8x_3 \rightarrow \min \quad (11)$$

при условиях:

$$0.6x_1 + 0.7x_2 + 0.4x_3 \leq 200; \quad (12)$$

$$0.6x_1 + 0.75x_2 + 0.8x_3 \leq 700; \quad (13)$$

$$1.4x_1 + 0.5x_2 + 1.4x_3 \leq 300; \quad (14)$$

$$x_1 \geq 550; x_2 \geq 350; \quad (15)$$

Целевые функции $F(x)$, представленные в (1), (6), (11), выражают возможные значения нагрузок, при которых сервер может корректно обрабатывать, передавать, принимать данные, определенные пользователем. Таким образом, для получения оптимального плана необходимо минимизировать целевые функции $F(x)$ с соблюдением всех условий задачи, которые накладывают ограничения на $F(x)$.

В выражениях (1)-(15) описана задача для нахождения экстремумов линейных функций в условиях системы линейных ограничений.

3. Исследования методов решения

Исходя из полученной обобщенной математической модели решения, изложенных в постановке задач, для решения обычно используют математические методы поиска экстремумов. Среди них наиболее развитыми и широко используемыми являются методы математического программирования, методы эволюционной оптимизации (генетические алгоритмы) и классические итерационные методы достижения экстремумов функций.

К математическому программированию относятся: 1. Линейное программирование. 2. Нелинейное программирование. 3. Теория графов. 4. Целочисленное программирование. 5. Стохастическое линейное программирование. 6. Задачи теории массового обслуживания. 7. Динамическое программирование. 8. Геометрическое программирование.

Ввиду того, что задача распределения нагрузок на сервер имеет вид задачи на нахождение экстремумов функций и состоит из трех систем уравнений, что описано в ее условиях, и данные три функции цели сводятся к нахождению минимумов, то решать задачу нужно универсальным методом линейного программирования – симплекс-метод и стохастический Байесов принцип.

Симплекс-метод. Данный метод решения задачи линейного программирования является универсальным. С его помощью можно решить любую задачу такого типа. В его основу положена идея последовательного улучшения получаемого решения.

Геометрический смысл данной задачи состоит в том, чтобы последовательно перейти от одной вершины многогранника ограничения нагрузок на сервер к соседней таким образом, чтобы целевая функция имела наилучшее значение до тех пор, пока не будет найдено оптимальное решение распределения нагрузок на сервер. В нашем случае – тах нагрузок.

Процесс нахождения оптимального плана предполагает реализацию 3-х элементов:

- 1) определение первоначального допустимого базисного решения задачи распределения нагрузок;
- 2) правило перехода к лучшему решению для нахождения \max нагрузок;
- 3) критерий проверки оптимальности для решения данной задачи.

Эта задача касается анализа линейного программирования. Но существуют еще модели стохастического характера, которые можно свести к детерминированным линейным моделям большей размерности; при этом увеличение размерности в реальных задачах настолько значительно, что нахождение оптимального решения подчас оказывается практически невозможным. Стоит рассмотреть стохастический Байесов принцип. Данный метод в условиях частичной неопределенности предполагает, что возможным состояниям природы можно приписать вероятность их наступления и, определив математическое ожидание выигрыша для каждого решения, выбрать то, которое обеспечит наибольшее значение выигрыша.

При этом предполагается, что ситуация, в которой принимается решение, характеризуется следующими обстоятельствами:

- вероятности появления состояния F_j известны и не зависят от времени;
- решение реализуется (теоретически) бесконечно много раз;
- для малого числа реализаций решения допускается некоторый риск.

При достаточно большом количестве реализаций среднее значение постепенно стабилизируется. Поэтому при полной (бесконечной) реализации какой-либо риск практически исключён.

4. Алгоритмы решения задачи

Для конкретных методов решений стоит сформулировать следующие примеры:

Симплексный метод включает в себя ряд этапов и может быть сформулирован в виде алгоритма.

Шаг 1. Формулировка ЗЛП (формирование целевой функции и системы ограничений распределения нагрузок на сервер). Из этого следует, что решается задача на отыскание максимума. Следует привести общую постановку данной задачи для достижения минимума времени на обработку запросов:

$$F(x) = 60x_1 + 25x_2 + 15x_3 \rightarrow \min \quad (16)$$

при условии, что:

$$0.3x_1 + 0.5x_2 + 0.4x_3 \leq 300; \quad (17)$$

$$0.5x_1 + 0.7x_2 + 0.8x_3 \leq 800; \quad (18)$$

$$1.3x_1 + 0.1x_2 + 1.2x_3 \leq 200; \quad (19)$$

$$x_1 \geq 500; x_2 \geq 300; \quad (20)$$

В этих выражениях x_1, x_2, x_3 – критерии, по которым рассчитываются нагрузки на серверы.

Шаг 2. Приведение задачи к канонической форме, что является переводом функции ограничений в систему уравнений. Для этого нужно вводить дополнительные неотрицательные переменные. Следует знать, что эти переменные должны иметь тот же знак, что и свободные члены системы ограничений. Если это условие не выполняется, то нужно использовать метод искусственного базиса (М-метод).

В результате получим каноническую форму задачи для распределения нагрузок приложений на сервере:

$$F(x) = 60x_1 + 25x_2 + 15x_3 \rightarrow \min \quad (21)$$

при тех же ограничениях:

$$0.3x_1 + 0.5x_2 + 0.4x_3 - x_4 = 300; \quad (22)$$

$$0.5x_1 + 0.7x_2 + 0.8x_3 - x_5 = 800; \quad (23)$$

$$1.3x_1 + 0.1x_2 + 1.2x_3 - x_6 = 200; \quad (24)$$

В результате этого можно получить систему ограничений данной задачи по 3-м критериям.

Шаг 3. Построение исходной симплекс-таблицы, в результате чего можно получить первоначальный план решения задачи. Он показан в табл. 1.

Таблица 1

	C_j		$C_1=60$	$C_2=25$	$C_3=15$	$C_4=0$	$C_5=0$	$C_6=0$
Базис	C_i	$P_0=X_0$	P_1	P_2	P_3	P_4	P_5	P_6
P_4	0	300	0,3	0,5	0,4	-1	0	0
P_5	0	800	0,5	0,7	0,8	0	-1	0
P_6	0	200	1,3	0,1	1,2	0	0	-1
F_j		0	0	0	0	0	0	0
Δ_j		-	-60	-25	-15	0	0	0

Итак, в левом столбце записываются основные (базисные) переменные, в первой строке перечисляются все переменные задачи. Крайний правый столбец содержит свободные члены системы ограничений b_1, b_2, \dots, b_m . В последней строке таблицы, которая называется оценочной, записываются коэффициенты целевой функции, а также ее значение. В рабочую область таблицы занесены коэффициенты a_{ij} при переменных системы ограничений.

Шаг 4. Проверка условия $c_{ij} \geq 0$. Если условие не действительно – осуществляется переход к шагу 5, если да – задача решена. Таким образом, на данном шаге проверяется наличие положительных элементов в последней строке симплексной таблицы.

Шаг 5. Выбор разрешающего столбца и строки. Столбец выбирается в соответствии со следующим условием:

$$\theta = x_j / x_{ik}, x_{ik} \geq 0, \quad (25)$$

где k – номер разрешающего столбца.

Таким образом, при определении разрешающего столбца просматривается выбранная строка симплексной таблицы и в ней отыскивается положительный наибольший элемент. Для данной задачи этим элементом будет $x_k = 1.3$.

Шаг 6. Проверка условий:

– если $x_k \geq 0$, то опорный план и есть решением задачи;

– $x_k < 0$ следует рассмотреть шаг 7.

Проверять первый раз это условие не стоит, так как изначально опорный план задачи распределения ресурсов не имеет отрицательных значений, поэтому нужно переходить к шагу 7.

Шаг 7. Пересчет элементов симплекс-таблицы и переход к новому базисному решению. Для элементов разрешающей строки используются следующие формулы:

$$x'_{rj} = x_{rj} / x_{rk}, \quad (26)$$

$$x'_{ij} = x_{ij} - x_{ri} * x_{ik} / x_{rk}, \quad (27)$$

$$j = \overline{0, (r + m)}, \quad (28)$$

где r – номер разрешающей строки; k – номер разрешающего столбца; x'_{rj}, x'_{ij} – новые значения пересчитываемых элементов.

По окончании пересчета осуществляется возврат к шагу 4.

Если в результате выполнения шага 4 были получены положительные значения строки j с, то оптимальный план задачи был достигнут.

Таким образом, с помощью симплекс-метода было найдено решение задачи распределения нагрузок приложений (служб) на сервере.

Базовый план выглядит так:

$$x^* = \|112.9; 532.26; 0; 0; 370.97; 0\|.$$

В результате получили значение целевой функции:

$$F(x) = 25 * 532.26 + 0 * 370.97 + 60 * 112.9 = 20080.$$

Допустимые нагрузки на данном сервере с ограничением на минимальное время выполнения действий (приложений, служб) на сервере могут составить не больше 20080 запросов.

Аналогично вычислениям на минимальное время выполнения служб можно получить значение критерия на минимум затрат при обработке запросов приложений (служб).

Оптимальный план будет выглядеть так:

$$x^* = \|156.5; 250; 0; 0; 328.13; 0\|,$$

$$F(x) = 30 * 250 + 0 * 328.13 + 55 * 156.5 = 16093.$$

Минимальное время выполнения служб будет при подаче 16093 запросов на сервер.

При достижении минимума ошибок во время выполнения обработки информации на сервере с помощью служб или приложений будут другие данные:

$$x^* = \|0; 0; 214.29; 114.29; 528.57; 0\|,$$

$$F(x) = 0 * 114.29 + 0 * 528.57 + 30 * 214.29 = 6428.$$

Минимум ошибок будет достигнуто при выполнении 6428 запросов.

Критерий Байеса-Лапласа применяют к ситуации, в которой принимается решение о распределении нагрузок приложений на сервер.

Расчет критерия Байеса-Лапласа выглядит так:

$$K_i^{BL} = \sum_j p_j u_{ij}, \quad (29)$$

$$i^{BL} = \arg \max_i (K_i^{BL}). \quad (30)$$

Шаг 1. Определение вероятностей состояний сервера для данной задачи. Они будут равны $p_1=0.3$; $p_2=0.5$; $p_3=0.2$.

Шаг 2. Составление матрицы ценности.

Соответственно она будет состоять из альтернативных решений, которые включают в себя состояния сервера и решений.

Это показано в табл. 2.

Таблица 2

Альтернативные решения	Состояния сервера		
	1	2	3
Решение 1	10000	15366	9870
Решение 2	8750	14800	7650
Решение 3	9630	15560	9320

Шаг 3. Определение критериев по Байесу-Лапласу. Значения критериев будут равны:

$$K_1^{BL} = 0.3 * 10000 + 0.5 * 15366 + 0.2 * 9870 = 12657;$$

$$K_2^{BL} = 11555;$$

$$K_3^{BL} = 12533;$$

Шаг 4. Определение наилучшего решения – максимального. Для данной задачи теперь можно определить, что наилучшим решением является первое:

$$i^{BL} = \arg \max(12657, 11555, 12533) = 1,$$

$$K^{BL} = 1.$$

Таким образом, найдено решение для распределения нагрузок на сервере: из трех решений первое оказалось наилучшим по критерию Байеса-Лапласа.

5. Выводы

В ходе выполнения работы был рассмотрен метод оптимизации распределения допустимых нагрузок приложений (служб) на сервер с использованием определенных условий. Основным инструментом при решении данной задачи стало математическое моделирование – формальное описание изучаемого явления и исследование с помощью математического аппарата. Была составлена математическая модель задачи. Для нахождения решений выбран симплекс-метод и стохастический принцип Байеса-Лапласа.

Научная новизна. Важность использования сервера, а также регулярной проверки наличия доступа к нему в век информационных технологий и глобального присутствия любого бизнеса в Интернете трудно переоценить. При неработающем хостинге, кроме недоступности некоторого сайта посетителям, могут быть и более серьезные проблемы. Если робот поисковой системы попадает на необрабатываемый сайт, то он может быть исключен из выдачи поисковика. Соответственно есть возможность в потере очень многих посетителей и потенциальных клиентов. Таким образом, данное понятие означает тот временной промежуток, в течение которого сервер работает без неполадок.

Практическая значимость. Полученные научные результаты данного исследования имеют большое практическое значение для разработки плана нагрузок с точки зрения минимизации ошибок, времени на обработку запросов приложений (служб) и затрат работы технического оборудования.

Вопрос обслуживания сервера очень часто встает перед разработчиками или владельцами крупных высоко посещаемых проектов, ведь масштабов обычного виртуального хостинга уже не хватает, а VIP хостинг дорогой и не выгоден с точки зрения эффективности. В этой связи приходится арендовать свой сервер, однако его необходимо администрировать, управлять, защищать и так далее. Серверы используют различные компании в своих офисах для обеспечения эффективной работы персонала, финансовой и бухгалтерской службы, которые могут работать в одной программе 1С, например, чтобы всегда иметь свежие

обновленные данные. Серверы в компаниях обеспечивают выход сотрудников в Интернет, регламентируют доступ к определенным ресурсам, чтобы персонал занимался работой, а не социальными сетями и другими развлечениями.

Работа сотрудников любой фирмы во многом зависит от эффективности работы сервера. Хорошо отлаженный сервер увеличивает производительность и эффективность работы компании в целом в несколько раз. Сбои на сервере могут привести не только к отсутствию доступа в сеть Интернет, но и к потере ценных данных, простою в работе, потере заявок, просрочке выполнения работ и многое другое. Сегодня почти все компании, нацеленные на долгосрочную и плодотворную деятельность, не могут обойтись без развитой компьютерной сети, а значит и без сервера, который эту сеть обслуживает и организует. Здесь очень важно подумать не только об установке и настройке сервера, но и о его дальнейшем качественном и квалифицированном обслуживании.

Литература: 1. Гвоздинский А.Н., Губин В.А., Шергин В.Л. Методы оптимизации в организационном управлении», ХНУРЭ, 2014. 395с.2. Бондаренко М.Ф. Оптимізаційні задачі в системах прийняття рішень: підручник / М.Ф. Бондаренко, А.М. Гвоздинський. Харків: ХТУРЕ, 1998. 219 с.

Transliterated bibliography:

1. Gvozdzinskiy A.N., Gubin V.A., Shergin V.L. Metody optimizatsii v organizatsionnom upravlenii», HNURE, 2014. 395s.
2. Bondarenko M.F. Optimizatsiyni zadachi v sistemah priynyattya rishen: pdruchnik / M.F. Bondarenko, A.M. Gvozdzinskiy. Harkiv: HTURE, 1998. 219 s.

Поступила в редколлегию 23.06.2016

Рецензент: д-р техн. наук, проф. Путятин Е.П.

Гвоздинский Анатолий Николаевич, канд. техн. наук, профессор кафедры искусственного интеллекта ХНУРЭ. Научные интересы: оптимизация процедур принятия решения в сложных системах управления. Адрес: Украина, 61166, Харьков, ул. акад. Ляпунова, 7, кв. 9, тел. 702-38-33.

Столяренко Константин Сергеевич, студент группы СА-14-1 кафедры прикладной математики ХНУРЭ. Научные интересы: проектирование клиент-серверных приложений под мобильную платформу iOS, разработка баз данных и их комплексное взаимодействие. Адрес: Украина, 61072, Харьков, пер. Отакара Яроша, 12А, кв. 155, тел. +38066-706-58-69.