

Институт математики им. С. Л. Соболева СО РАН
пр. Акад. Коптюга, 4, Новосибирск, 630090, Россия

Новосибирский государственный университет
ул. Пирогова, 2, Новосибирск, 630090, Россия

E-mail: jkochet@math.nsc.ru; nkochet@math.nsc.ru

ЗАДАЧА БАЛАНСИРОВКИ НАГРУЗКИ НА СЕРВЕРЫ *

Статья посвящена новой задаче балансировки нагрузки на серверы, возникающей при оптимизации облачного хостинга веб-приложений и пользовательского контента. Получена математическая модель в терминах частично-целочисленного линейного программирования. Показана NP-трудность задачи. Разработан приближенный алгоритм ее решения с апостериорной оценкой уклонения от оптимума. Проведены вычислительные эксперименты с числом серверов до 20.

Ключевые слова: балансировка загрузки, NP-трудные задачи, целочисленное программирование, приближенные алгоритмы.

Введение

Рассмотрим следующую прикладную задачу. Имеется набор серверов, на каждом из которых размещены диски (точнее, их образы). На дисках хранятся интернет-сайты с разнородной информацией. Пользователи посещают сайты, что создает определенную нагрузку на серверы. Эта нагрузка меняется со временем и характеризуется несколькими параметрами – CPU, RAM и др. По каждому диску известна активность пользователей в течение планового периода. Эта активность позволяет определить нагрузку на каждый сервер в каждый момент времени по каждому параметру. Время предполагается дискретным, т. е. измерение нагрузки производится, например, каждый час (полчаса, минуту). Если нагрузка по каждому параметру не превосходит заданного порога, то сервер находится в рабочем режиме. В противном случае сервер работает с перегрузкой, что нежелательно, так как возникают задержки на запросы пользователей. Чтобы избежать перегрузки, диски можно перемещать с одного сервера на другой. Такое перемещение требует определенных вычислительных затрат. Будем называть их накладными расходами. Предполагается, что для каждого диска известны накладные расходы по каждому параметру при его изъятии с сервера и подключении к любому другому серверу. Начальное распределение дисков по серверам считается известным. Задача состоит в том, чтобы перераспределить диски по серверам и достичь минимальной суммарной перегрузки на всем плановом периоде при ограничениях на накладные расходы по каждому серверу.

Решение этой задачи потребовалось в НИОКР, выполняемой Новосибирским государственным университетом по заказу Минобрнауки РФ (договор № 02.G25.31.0054).

* Работа выполнена при финансовой поддержке Минобрнауки РФ (договор № 02.G25.31.0054).

Математическая модель

Введем следующие обозначения:

S – множество серверов;

D – множество дисков;

T – плановый период;

R – множество характеристик нагрузки (CPU, RAM, ...);

c_{drt} – нагрузка диска d в момент t по характеристике r ;

\bar{c}_{sr} – пороговая нагрузка сервера s по характеристике r ;

x_{ds}^0 – начальное распределение дисков по серверам;

$b_{sdr}^w (b_{sdr}^e)$ – накладные расходы по характеристике r на вставку (изъятие) диска d в сервер s ;

$B_{sr}^w (B_{sr}^e)$ – предельно допустимые накладные расходы по характеристике r на вставку (изъятие) дисков в сервер s .

Переменные задачи:

$$x_{ds} = \begin{cases} 1, & \text{если диск } d \text{ ставится на сервер } s, \\ 0 & \text{в противном случае,} \end{cases}$$

y_{str} – перегрузка на сервере s в момент времени t по характеристике r .

С использованием введенных обозначений задача балансировки трафика может быть записана в виде задачи частично-целочисленного линейного программирования:

$$\min \sum_{s \in S} \sum_{t \in T} \sum_{r \in R} y_{str} \quad (1)$$

при ограничениях

$$y_{str} \geq \sum_{d \in D} c_{drt} x_{ds} - \bar{c}_{sr} \quad \text{для всех } s \in S, t \in T, r \in R; \quad (2)$$

$$\sum_{s \in S} x_{ds} = 1 \quad \text{для всех } d \in D; \quad (3)$$

$$\sum_{d \in D} b_{sdr}^w x_{ds} (1 - x_{ds}^0) \leq B_{sr}^w \quad \text{для всех } s \in S, r \in R; \quad (4)$$

$$\sum_{d \in D} b_{sdr}^e (1 - x_{ds}) x_{ds}^0 \leq B_{sr}^e \quad \text{для всех } s \in S, r \in R; \quad (5)$$

$$y_{str} \geq 0, x_{ds} \in \{0, 1\} \quad \text{для всех } d \in D, s \in S, t \in T, r \in R. \quad (6)$$

Целевая функция (1) задает суммарную перегрузку серверов на всем плановом периоде. Неравенство (2) позволяет определить эту перегрузку для каждого сервера в каждый момент времени по каждой характеристике. Равенство (3) гарантирует в точности один сервер для каждого диска. Неравенства (4) и (5) ограничивают накладные расходы по каждому серверу и каждой характеристике при изъятии и вставке дисков.

Представление задачи в терминах частично-целочисленного программирования позволяет использовать для ее решения коммерческое программное обеспечение, например, IBM ILOG CPLEX¹, GUROBI² и др. К сожалению, наличие большого разрыва целочисленности при переходе к непрерывным переменным $x_{ds} \geq 0$ не дает возможности найти точное решение на примерах даже средней размерности. При $|S|=20$, $|D|=200$, $|T|=150$, $|R|=2$ расчеты на современном компьютере могут занимать более суток без гарантии найти точное решение. Следующее утверждение отчасти объясняет причины этого и подталкивает к разработке специализированных методов. Покажем, что задача (1)–(6) является NP-трудной,

¹ <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>

² <http://www.gurobi.com/>

даже при $|T|=1$, $|R|=1$, $|S|=2$ и нулевых накладных расходах. Рассмотрим следующую NP-полную задачу о разбиении. Заданы множество A и целые положительные размеры $w(a)$ всех элементов $a \in A$. Вопрос: существует ли такое подмножество $A' \subset A$, что $\sum_{a \in A'} w(a) = \sum_{a \in A \setminus A'} w(a)$? Приведем сведение этой задачи к задаче (1)–(6) при $|T|=1$, $|R|=1$, $|S|=2$ и нулевых накладных расходах. Для этого по исходным данным задачи разбиения построим исходные данные задачи (1)–(6). Пусть $|D|=|A|$ и каждому элементу $a \in A$ соответствует диск $d(a)$. Положим $c_{drt} = w(a)$ для $d = d(a)$ и $\bar{c}_{sr} = \frac{1}{2} \sum_{a \in A} w(a)$, $s = 1, 2$. Тогда решение задачи (1)–(6) с нулевой перегрузкой соответствует ответу «да» в задаче разбиения и наоборот.

Приближенный метод. Рассмотрим задачу линейного программирования, которая получается из задачи (1)–(6) заменой целочисленных переменных x_{ds} на непрерывные переменные $x_{ds} \geq 0$. Такая задача полиномиально разрешима. Ее точное решение дает нижнюю оценку оптимума для исходной задачи (1)–(6), что позволяет получить апостериорную оценку точности приближенных алгоритмов.

Пусть x_{ds}^* – оптимальное решение такой задачи. Будем использовать его для приближенного решения исходной задачи. Пусть параметр k_s показывает число дисков, уже зафиксированных на сервере s , а вспомогательная 0–1 матрица $\{z_{ds}\}$ показывает их распределение по серверам. На каждом шаге алгоритма решается задача линейного программирования с дополнительным ограничением $x_{ds} \geq z_{ds}$ для всех $d \in D$, $s \in S$. По оптимальному решению x_{ds}^* этой задачи будем фиксировать на каждом сервере по одному диску, выбирая диск с наибольшим значением x_{ds}^* . При каждой такой фиксации проверяются ограничения (4)–(5). Если они не нарушаются, то соответствующая величина z_{ds} полагается равной 1. В противном случае переходим к следующему серверу. Таким образом, размерность задачи сокращается до тех пор, пока оставшаяся подзадача не станет достаточно малой и ее можно будет решить точно методом ветвей и границ (CPLEX). Общая схема алгоритма может быть представлена следующим образом.

0. Полагаем $k_s := 0$, $z_{ds} := 0$ для всех $d \in D$, $s \in S$.

1. Решаем задачу линейного программирования с дополнительным ограничением $x_{ds} \geq z_{ds}$ для всех $d \in D$, $s \in S$.

2. Для каждого $s \in S$ находим диск $d \in D$ с максимальным значением x_{ds}^* . Если $x_{ds}^* > 0,5$ и требование $z_{ds} = 1$ не нарушает условий (4) и (5), то полагаем $z_{ds} := 1$, $k_s := k_s + 1$.

3. Если $k_s \leq k_{\max}$ и число зафиксированных дисков возросло, то возвращаемся на шаг 1.

4. Решаем исходную задачу (1)–(6) с дополнительным ограничением $x_{ds} \geq z_{ds}$ для всех $d \in D$, $s \in S$.

На шаге 2 неравенство $x_{ds}^* > 0,5$ гарантирует, что один диск не будет установлен на несколько серверов одновременно и условие (3) будет выполняться. На шаге 4 требуется решение исходной задачи (1)–(6) с фиксацией части дисков, что при малом значении параметра k_{\max} может требовать больших вычислительных усилий. Тем не менее даже при остановке расчетов по времени, наилучшее найденное решение методом ветвей и границ может давать хорошее приближение к оптимуму в исходной задаче.

Экспериментальные исследования. Для проведения численных экспериментов рассматривался следующий класс тестовых примеров: $|S|=20$, $|D|=200$, $|T|=150$, $|R|=2$. Начальное распределение дисков по серверам, величины $x_{ds}^0 \in \{0, 1\}$ порождались случайным образом с равномерным распределением. Для каждого диска выбирался один сервер независимо от других дисков. Для вычисления нагрузки и накладных расходов сначала каждому диску

приписывалось значение средней нагрузки c_d , $d \in D$. Она бралась из интервала [50, 100] с равномерным распределением, а затем полагалось:

- $c_{drt} = c_d + \mu_{drt}$ для всех $d \in D$, $r \in R$, $t \in T$;
- $b_{sdr}^w = c_d + \eta_{sdr}$ для всех $s \in S$, $d \in D$, $r \in R$;
- $b_{sdr}^e = c_d + \theta_{sdr}$ для всех $s \in S$, $d \in D$, $r \in R$,

где величины $\mu_{drt} \in [-20, 20]$, η_{sdr} , $\theta_{sdr} \in [-3, 3]$ также выбирались независимо друг от друга с равномерным распределением из заданных интервалов. Эксперименты проводились при различных значениях пороговой нагрузки $\bar{c}_{sr} \in [600, 850]$ и $B_{sr}^w = B_{sr}^e = 50, 150$, $k_{\max} = 7$. Результаты расчетов представлены в таблице:

\bar{c}_{sr}	t_{LP}	F_{LP}	#1	t_{ILP}	F_{ILP}
$B_{sr}^w = B_{sr}^e = 150$					
850	2''	0	185	8''	0
840	2''	0	185	21''	0
830	2''	0	182	11''	0
820	2''	0	177	50' (остан.)	4
810	2''	0	157	35'	0
800	2''	0	146	50' (остан.)	1370
750	3'20''	0	40	–	–
700	12'14''	87 700	40	–	–
650	3'02''	382 399	80	50' (остан.)	469 294
600	1'04''	682 399	83	50' (остан.)	761 107
$B_{sr}^w = B_{sr}^e = 50$					
850	2''	0	146	6''	0
840	2''	0	143	7''	0
830	2''	0	145	17''	0
820	2''	0	136	1'10''	0
810	2''	0	143	50' (остан.)	1834
800	2''	0	123	50' (остан.)	2480
750	3'18''	0	40	–	–
700	7'3''	96 911	40	50' (остан.)	188 306
650	2'40''	382 399	70	10'7''	407 792
600	58''	682 399	98	1'25''	702 039

При пороговой нагрузке 850 задача легко решается точно. Линейное программирование требует всего две секунды расчетов на PC Pentium Core 2Duo, CPU 2.66 GHz, RAM 2 GB (столбец t_{LP}). Целочисленное линейное программирование занимает восемь секунд (столбец t_{ILP}). Перегрузка отсутствует в оптимальном решении как линейного программирования (столбец F_{LP}), так и целочисленного линейного программирования (столбец F_{ILP}). Интересно отметить, что линейное программирование дает много единиц в оптимальном решении (столбец #1). Всего 15 дисков оказались «размазанными» по серверам ($x_{ds}^* \notin \{0, 1\}$).

Значение $\bar{c}_{sr} = 600$ также оказалось простым для расчетов, хотя оно соответствует большой перегрузке и требует значительно большего времени для расчетов. Линейное программирование потребовало чуть больше одной минуты, а целочисленное линейное программирование было остановлено через 50 минут расчетов. Однако полученное решение не сильно отличалось от оптимума, относительная погрешность не превысила 12%. Заметим, что в случае $B_{sr}^w = B_{sr}^e = 150$ ограничения (4), (5) фактически оказываются несущественными, что, конечно, упрощает задачу (см. таблицу). Тем не менее она не становится тривиальной.

Значения $\bar{c}_{sr} = 700, 750$ соответствуют наиболее трудному случаю. Линейное программирование требует значительно большее время, а целочисленное линейное программирование оказывается задачей слишком большой размерности. Алгоритм не может сократить размерность задачи, зафиксировав часть дисков, так как для подавляющего числа пар (d, s) величина $x_{ds}^* \leq 0, 5$. В итоге задача на шаге 4 оказывается слишком тяжелой и расчеты в течение 50 минут не приводят к практически значимым результатам. Для этих значений пороговой нагрузки требуются новые подходы, например основанные на методах локального поиска и метаэвристиках [1].

В таблице представлены результаты расчетов для $B_{sr}^w = B_{sr}^w = 50$. Это более трудные примеры, они требуют больше времени, но значения $\bar{c}_{sr} = 850, 600$ тем не менее решаются сравнительно легко. Значения $\bar{c}_{sr} = 700, 750$, как и прежде, наиболее трудны для данного подхода.

Заключение

В работе представлена новая прикладная задача балансировки интернет-трафика, возникающая при оптимизации облачных сервисов. Построена математическая модель частично-целочисленного программирования, показана NP-трудность задачи. Для ее решения разработан приближенный алгоритм, основанный на точном решении соответствующей линейной релаксации. Проведены численные эксперименты, в ходе которых установлены наиболее трудные примеры, требующие разработки новых алгоритмических подходов. Так как исходная задача является NP-трудной, для дальнейших исследований можно выделить как минимум три перспективных направления.

1. Разработка эвристических методов и, в частности, метаэвристик, к которым относятся генетические алгоритмы [2; 3], локальный поиск с чередующимися окрестностями [4], имитация отжига [5], поиск с запретами [6] и др. Эти методы хорошо зарекомендовали себя при решении многих NP-трудных задач. Они не опираются на линейное программирование, их точность не зависит от разрыва целочисленности. По сути, это общие схемы для поиска глобального оптимума. Их эффективность во многом зависит от того, насколько удачно разработчики воспользовались спецификой решаемой задачи и сумели найти подходящие структуры данных для повышения агрессивности поиска.

2. Разработка точных методов, среди которых следует выделить как классические методы ветвей и границ, динамического программирования, методы генерации переменных и ограничений [7], так и новые гибридные схемы, сочетающие в себе классические методы и метаэвристики [8; 9]. Особый интерес здесь представляют точные методы, например, метод локальных ветвлений (local branching), где поиск наилучшего решения в окрестности осуществляется методами математического программирования.

3. Вычисление нижних оценок оптимума, получаемых как априорными, так и апостериорными методами [10; 11]. С теоретической точки зрения здесь любопытно получить границы аппроксимируемости рассматриваемой задачи, если такие существуют, или попытаться разработать аппроксимационные схемы. Они позволяют получать решения со сколь угодно малой относительной погрешностью ε при трудоемкости алгоритмов, полиномиально зависящей от величины $1/\varepsilon$.

В данной работе сделан только первый шаг в разработке таких методов.

Список литературы

1. Кочетов Ю. А. Методы локального поиска для дискретных задач размещения. Модели и алгоритмы. Saarbrücken: Lambert Academic Publishing, 2011. 259 с.
2. Plyasunov A. V., Panin A. A. The Pricing Problem. Part 1: Exact and Approximate Algorithms // Journal of Applied and Industrial Mathematics. 2013. Vol. 7. Is. 2. P. 241–251.

3. *Kochetov Yu. A., Plyasunov A. V.* Genetic Local Search for the Graph Partitioning Problem under Cardinality Constraints // *Computational Mathematics and Mathematical Physics*. 2012. Vol. 52. No. 1. P. 157–167.
4. *Davydov I., Kochetov Yu., Carrizosa E.* VNS Heuristic for the $(r|p)$ -Centroid Problem on the Plane // *Electronic Notes in Discrete Mathematics*. 2012. Vol. 39. P. 5–12.
5. *Diakova Z., Kochetov Yu.* A Double VNS Heuristic for the Facility Location and Pricing Problem // *Electronic Notes in Discrete Mathematics*. 2012. Vol. 39. P. 29–34.
6. *Давыдов И. А.* Локальный поиск с запретами для дискретной задачи о $(r|p)$ -центроиде // *Дискрет. анализ и исслед. операций*. 2012. Т. 19, № 2. С. 19–40.
7. *Alekseeva E., Kochetova N., Kochetov Yu., Plyasunov A.* Heuristic and Exact Methods for the Discrete $(r|p)$ -Centroid Problem // *Lecture Notes in Computer Science*. 2010. Vol. 6022. P. 11–22.
8. *Kononova P. A., Kochetov Yu. A.* The Variable Neighborhood Search for the two Machine Flow Shop Problem with a Passive Prefetch // *Journal of Applied and Industrial Mathematics*. 2013. Vol. 7. Is. 1. P. 54–67.
9. *Береснев В. Л., Гончаров Е. Н., Мельников А. А.* Локальный поиск по обобщенной окрестности для задачи оптимизации псевдодобулевых функций // *Дискрет. анализ и исслед. операций*. 2011. Т. 18, № 4. С. 3–16.
10. *Vasil'ev I. L., Klimentova K. B., Kochetov Yu. A.* New Lower Bounds for the Facility Location Problem with Clients' Preferences // *Computational Mathematics and Mathematical Physics*. 2009. Vol. 49. No. 6. P. 1010–1020.
11. *Kochetov Yu. A.* Computational Bounds of Local Search in Combinatorial Optimization // *Computational Mathematics and Mathematical Physics*. 2008. Vol. 48. No. 5. P. 788–807.

Материал поступил в редколлегию 05.11.2013

Yu. A. Kochetov, N. A. Kochetova

THE SERVERS LOAD BALANCING PROBLEM

The paper is devoted to a new load balancing problem originated from the cloud computing and optimal hosting for the web applications and user's contents. We present a mixed integer linear programming formulation and show that it is NP-hard problem. We design an approximation algorithm with a posterior bound for the deviation from the optimum. Computational results are conducted with number of servers up to 20.

Keywords: Load balancing, NP-hard problems, integer programming, approximation algorithms.