ИССЛЕДОВАНИЕ ЗАДАЧИ БАЛАНСИРОВКИ НАГРУЗКИ В КЛАСТЕРНЫХ СИСТЕМАХ С ОЧЕРЕДЬЮ

СУХОПЛЮЕВ Данил Игоревич
студент
ФГБОУ ВО «Московский государственный университет информационных технологий, радиотехники и электроники» (МИРЭА)
г. Москва, Россия

Современные кластерные системы с ростом трафика все чаще сталкиваются с проблемой перегрузки на различных узлах системы, что влечет за собой понижение стабильности, производительности, эффективности. Балансировщик нагрузки обеспечивает возможность оптимизировать использование ресурсов, с целью минимизации финансовых расходов на вертикальное масштабирование. Особый интерес данного подхода вызывает определение задачи балансировки нагрузки в кластерных системах с очередью, ее математическая модель, доказательство NP полноты и вычислимость методами математического программирования.

Ключевые слова: микросервисы, кластерная архитектура, динамическая балансировка нагрузки, математическое моделирование.

С целью разбиения процессов высоконагруженных веб приложений используется архитектурный подход в виде построения распределенных систем. Каждый функциональный элемент представляет собой доменный сервис или кластер.

Во время процесса обработки входного трафика кластер или микросервис, может сталкиваться с динамической нагрузкой. Эта нагрузка меняется со временем и характеризуется множеством параметров – ЦПУ, виртуальная память, скорость чтения-записи на персистентные хранилища и т. д., и т. п.

Для того, что обработать входной блок данных, то есть сообщение, кластер может использовать временное хранилище, организованное по принципу структуры – очередь. Для того что бы гарантировать однородность потока сообщений, система может предоставлять информацию по количеству элементов в очереди в течение дискретного интервала времени. Эта численная метрика позволяет определить нагрузку на каждый кластер в каждый плановый период по каждому параметру.

Если нагрузка по каждому параметру не превосходит верхней границы оценки, то микросервис находится в рабочем режиме. В противном случае кластер работает с перегрузкой, что нежелательно, возникает задержка между временем ответа от системы, повышается мгновенная плотность, способная привести к полному отказу системы.

Уменьшение времени задержки может компенсироваться как вертикальным, так и горизонтальным масштабирование, что в свою очередь влечет к высокой стоимости разрешения проблемы.

Оптимальным решением в таком случае может являться балансировщик нагрузки, который обеспечивает функцию распределения задач между кластерами с целью максимизации использования вычислительного потенциала микросервисной системы.

Благодаря такому заключению, мы можем дать определение задачи балансировки нагрузки, как следствие функции балансировщика.

И поскольку в данном вопросе не менее важным является так же и математическая модель, то и обратимся к ней.

Прежде чем определить её структуру, введем обозначения всех значащих факторов:

S – конечное множество серверов;

D – конечное множество остаточных очередей;

T – элементарный интервал времени;

R – конечное множество характеристик нагрузки;

C_{drt} – количество элементов в очереди d в момент t по характеристике r;

$\overline{C_{sr}}$ – пороговая нагрузка на сервере s по характеристике r;

x_{ds}^{0} – начальное распределение остаточных очередей по серверам s;

$b_{sdr}^{w} b_{sdr}^{e}$ – повышение нагрузки по характеристике r из-за переполнения очереди d на сервере s;

$B_{sr}^{w} B_{sr}^{e}$- критическая нагрузка по характеристике r из-за переполнения остаточной очереди d;

Так как большинство современных кластерных систем используют реплицирование, то мы имеем переменную дискретного типа, а именно:

$$x_{ds} = \begin{cases} 1 \\ 0 \end{cases}$$

где 1 – если очередь d соответствует серверу s и 0 в противном случае.

Оценка перегрузки в момент t по характеристике r представляется, как - y_{str}

С учетом введённых переменных, математическая модель задачи балансировки нагрузки может быть задана 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

$$y_{str} \geq \sum_{d \in D} c_{drt} x_{ds} - \overline{c_{sr}} \quad (2)$$

$$\sum_{s \in S} x_{ds} = 1 \quad (3)$$

$$\sum_{d \in D} b_{sdr}^{w}\, x_{ds} (1 - x_{ds}^{0}) \leq B_{sr}^{w} \quad (4)$$

$$\sum_{d \in D} b_{sdr}^{e}\, (1 - x_{ds}) x_{ds}^{0} \leq B_{sr}^{e} \quad (5)$$

$$y_{str} \geq 0, x_{ds} \in \{0, 1\} \quad (6)$$

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 каждой очереди. Неравенства (4) и (5) ограничивают размер очереди по каждому серверу и каждой характеристике при переполнении.

Минимизация суммы перегрузок(1) по индексу параметров обеспечивает решение задачи балансировки нагрузки с остаточной очередью при заданных ограничениях(3)-(6) с помощью методов математического программирования.

Поэтому, для того, чтобы утверждать, что задача является вычислимой за полиномиальное время, требуется доказать, что она является NP полной.

Аппроксимация исходной задачи(1)–(6) с заменой целочисленных переменных x_{ds} на непрерывные переменные $x_{ds} \geq 0$ дает точное решение посредством CPLEX метода, который позволяет найти нижнюю оценку оптимума, что по своей сути является вероятнейшей точкой для оценки всех приближенных алгоритмов.

CPLEX сокращает размерность задачи до тех пор, пока оставшаяся подзача не станет достаточно малой и ее точно решение может быть найдено методом ветвей и границ.

Рассмотрим следующею NP полную задачу о разбиении. Положим, что заданы множества K и целые положительные размеры $w(K)$ для всех $k \ \in K$. Вопрос, а существует ли такое подмножество $K' \cup K$, что $\sum_{k \in K'} w(k) = \sum_{k \in K \setminus K'} w(k)$? Приведем сведения этой задачи к (1)–(6) при стандартных $|T| = 1, |R| = 1, |S| = 2$ и нулевых накладных расходах, то есть без оценки погрешности. Тогда решение (1)–(6) с нулевой перегрузкой дает ответ «да» в задаче разбиения и наоборот, из чего по индукции следует, что сама задача является NP полной.

Для численной оценки задачи, были проведены эксперименты в символьной машине Matlab на процессоре Ryzen с 8 физическими ядрами, 8 потока, 16 Гб оперативной памяти и 4 ГГц заявленной тактовой частоты.

Таблица 1

ЧИСЛЕННЫЙ ЭКСПЕРИМЕНТ

c_{sr}	F_{ILP}	F_{LP}	t_{ILP}	t_{LP}
830	0	0	17’	2’’
820	0	0	70’	2’’
810	1834	0	50’ (остан.)	2’’
800	2480	0	50’ (остан.)	2’’
750	-	0	-	2’’
700	188306	96911	50’ (остан.)	7’3’’
650	407792	382399	10’7’’	2’40’’

В таблице 1 представлены результаты численных экспериментов, где первый столбец – это пороговая нагрузка на сервер s в момент времени t, время в минутах и часах на решение задачи методом линейного программирования – t_{LP} и соответственно целочисленно-линейного программирования – t_{ILP}. F_{LP}, F_{ILP} – перегрузки при заданых методах.

Важно отметить, что при нагрузке в 750 перегрузка отсутствует в обоих методах, но само решение дает достаточно много единиц для распределения остаточных очередей x_{ds}.

Так же, как мы можем видеть, при нагрузке в 810, 800 и 700 программа была остановлена, это связано с тем что задача становится слишком тяжелой и расчеты в течении 50 минут не приводят к практически значимым результатам. Из этого следует, что для подобного класса тестовых примеров требуются новые подходы и алгоритмы, например, основанные на локальном поиске или различных эвристиках.

В рамках исследований были выявлены 3 основные группы алгоритмов балансировки нагрузки, а именно:

1) эвристические методы и мета эвристики, которые по сути являются общими схемами для методов поиска глобального оптимума, но их эффективность во многом зависит от того, на сколько удачно подобран алгоритм согласно специфики решаемой задачи;

2) CPLEX методы;

3) методы вычислений нижних оценок оптимума;

Все эти группы алгоритмов представляют из себя большой интерес для дальнейших исследований с целью ведения задачи оптимизации балансировки нагрузки в кластерных системах с очередью.

СПИСОК ЛИТЕРАТУРЫ

1. *Кочетов Ю.А.* Методы локального поиска для дискретных задач размещения. Модели и алгоритмы. Saarbrucken: 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.