

ЗАДАЧА КОМПОЗИЦИИ ВЕБ СЕРВИСОВ ПРИ ПАКЕТНОМ ЗАПРОСЕ

Нгуен Х.К., Иванов Н.Н.

Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь (220013, Республика Беларусь, г. Минск, ул. П. Бровки, 6), e-mail: khanh29bk@yahoo.com

С развитием распределенных информационных систем возникли задачи организации обработки информации с использованием удаленных ресурсов. Веб-сервисы применяются для обмена сообщениями в сети, из них можно комбинировать сложные сервисы для реализаций различных функций. Рассматривается оптимизационная задача композиции пакета сервисов из доступных веб-сервисов. Ставится задача многокритериальной оптимизации с ограничениями на ациклическом ориентированном графе. Векторный критерий агрегируется в единственную минимизируемую целевую функцию. Если целевая функция сепарабельная и монотонно возрастает, то для оптимального решения задачи выполняется принцип динамического программирования. Задача оптимизации с ограничениями на графе решается методом динамического программирования с коррекцией решения на узлах графа. Построен также эвристический алгоритм решения задачи с псевдолинейной оценкой сложности. Проведено экспериментальное сравнение алгоритма с методом динамического программирования.

Ключевые слова: задача композиции веб сервисов, оптимизация на сетевом графике, динамическое программирование.

COMPOSITION OF WEB-SERVICES ON BATCH QUERY

Nguyen K.Q., Ivanov N.N.

Department of electronic computing machines, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus (6 Brovki St., Minsk, 220013, Republic of Belarus), e-mail: khanh29bk@yahoo.com

A challenge of information processing with remote resources exploiting is arose under the progress in information technologies. Web-services are used for message exchanges, they may be combined for realizing complex activity in a network. An optimization problem of a batch query compounding from available web-services is under consideration. Multi-objective optimization problem with restrictions on acyclic directed graph is formulated. Objective functions are aggregated into single monotonic increasing goal function. If this function is separable, then principle of dynamic programming holds true for optimal solution of the problem. Formulated optimization problem with constrains is solved with dynamic programming exploiting, corrections of services assignment are performed on each node in searching procedure. Heuristic algorithm for the problem solution is presented, complexity of the algorithm is pseudo linear. The algorithm was compared with dynamic programming method, experimental results are shown.

Keywords: Web-service composition, optimization problem on net, dynamic programming method.

Введение

Веб-сервисы выполняют обмены в сети и должны удовлетворять критериям качества обслуживания Quality of Service (QoS), которые зависят от вида сообщения. Провайдеры предлагают широкий выбор веб-сервисов с дублированием функций сервисов. Возникает задача выбора сервисов, выполняющих необходимую функцию и обеспечивающих требуемый уровень QoS. Задача выбора композиции сервисов по своей сути является комбинаторной. Большинство математических моделей этой проблемы приводят к NP-трудным задачам. Потребителю сервисов для построения композиции требуются линейные или хотя бы полиномиальные алгоритмы.

Материалы и методы исследования

При создании сложного веб-сервиса в реальных системах можно комбинировать его различными способами из имеющегося набора элементарных сервисов, первое упоминание задачи композиции веб-сервисов появилось в 2004 году [4]. В математической постановке задачи композиции сервисов строится векторная целевая функция по количеству критериев QoS. Ставится многокритериальная задача с ограничениями. В литературе приводится около 40 различных критериев качества для задачи композиции. Если указаны все возможные последовательности выполнения сервисов для реализации заданной функции, то рассматривается оптимизация на графе.

Задача оптимальной композиции сервисов входит в класс комбинаторных задач оптимизации. Моделируются задачи булева программирования, значения 1 или 0 переменных указывают, какие сервисы выбраны для композиции функции. Из Парето-оптимальных решений выбирается одно, для этого критерии задачи взвешенной суммой приводятся к единственному. В [5, 6] такая задача решается в два этапа, на первом, локальном, из предложений провайдеров выбирается лучший набор сервисов, на втором, глобальном, по двум критериям выбирается Парето-оптимальное множество. Модели на графах принимают во внимание связи сервисов и ищут оптимальную цепь сервисов на сети. В [8] учтены 5 критериев качества: стоимость, затраты времени, надежность, доступность, рейтинг провайдера сервисов. На решение накладывают ограничения по времени и по стоимости сервисов. Для решения задачи на основе информации брокеров выбирается множество сервисов, выполняющих требуемую функцию, из него создается сеть выполнения работ. Из пяти критериев качества создается единственная целевая функция. Строится критический путь, сервисы которого дают решение. Модель оптимизации на графе рассматривается и в [7], здесь строится модель дискретного программирования в виде многомерной задачи о рюкзаке. Кроме графовой модели и дискретного программирования предлагаются другие подходы, например, моделирование сетью Петри, иерархическую модель доступа к сервисам.

Здесь рассматривается задача одновременной композиции нескольких сервисов.

Постановка задачи и математическая модель

Ставится задача композиции сложных веб-сервисов из наборов элементарных сервисов. Обычно информационная система имеет избыточный набор сервисов, которые частично или полностью дублируют функции друг друга. Сервисы публикуются в реестрах UDDI своих провайдеров, брокеры, программы поиска и предоставления услуг, выбирают требуемые сервисы, по критериям качества QoS формируя сложный сервис. Здесь решается задача композиции нескольких сложных сервисов (далее просто сервисов), для которых требуется выбрать оптимальные наборы атомарных сервисов (атомов). Даны структурные схемы наборов атомов, каждый реализует сервис. Математическую модель задачи представим как

оптимизацию векторного критерия на ориентированном взвешенном графе без контуров. Задача может моделироваться и в рамках дискретного программирования.

Без потери общности будем рассматривать минимизацию критериев. Предполагается, что целевые функции определены на всех допустимых решениях задачи и монотонно возрастают. Если есть хотя бы одно допустимое решение, то существует и оптимальное решение. В этой работе многокритериальная задача взвешенной суммой критериев приводится к оптимизации с одной целевой функцией.

Формализуем задачу композиции. Далее будем называть ориентированный конечный граф, не содержащий контуров, сетью. Пусть задан пакет сервисов $\{S_1, S_2, \dots, S_N\}$, которые требуется скомпоновать из доступных атомов. Атомы, связанные в структуры, выполняют заданные сервисы S_k . Структуры представлены сетями, узлы соответствуют событиям, дуги – атомам. Сеть не содержит контуров, то есть атомам нельзя составить цикл. Для повторов фрагмент графа копируется несколько раз.

На основании структур представления сервисов формируется сеть $G = (\mathcal{N}, \mathcal{A})$, где \mathcal{N} – узлы, \mathcal{A} – дуги. Узлы обозначим t_i , дугу, связывающую узлы t_i и t_j обозначим (i, j) , ей соответствует атом $s(i, j)$. Сеть G сформирована из структурных связей атомов, которые формируют сервисы S_k , $k = 1, 2, \dots, N$. Для каждого k задана сеть G_k атомов, из которых можно сконструировать сервис S_k . Сеть G объединяет сети G_k , $k = 1, 2, \dots, N$. Если сети G_k попарно не пересекаются, то задача распадается на N более простых независимых. Положим, что сеть G состоит из одной компоненты. Если у нее несколько начальных узлов, то дополнением фиктивных дуг создадим единственный начальный узел, аналогично создадим единственный конечный узел. Атомы $s(i, j)$ пометим числом $T_{\text{exec}}(i, j)$ – продолжительностью его выполнения, эта величина не зависит от S_k , и N -мерным вектором $I(i, j)$, его компонента k равна 1, если атом $s(i, j)$ принадлежит G_k , иначе компонента равна 0.

При распараллеливании сервисов, исходящих из события, возможны два различных случая, в одном случае необходимо выполнить все параллельные ветви (связь типа AND), в другом случае выполняется одна из параллельных ветвей (связь типа OR). Связи типа AND можно преобразовать в единую дугу, параллельные ветви атомов заменяются одним атомом. Будем предполагать, что в сети существуют только связи типа OR.

В сетевой постановке получаем задачу оптимизации нелинейной целевой функции на потоке заданной величины, с ограничениями на дугах. Допустимыми решениями задачи являются N полных цепей C_k , по числу сервисов S_k . Вводится NM булевых функций $x_{ij}^k(t)$ для дуг (i, j) , где M – число дуг сети G . Отсчет времени начинается в момент $t = 0$, интервал

определения функций $[0, T_x]$ выбирается так, что в момент T_x заканчивается выполнение всех сервисов S_k . Если в момент t_0 сервис S_k использует атом $s(i, j)$, то $x_{ij}^k(t_0) = 1$, иначе $x_{ij}^k(t_0) = 0$. Атом не может одновременно обслуживать более одного сервиса, поэтому накладываем ограничения на пропускную способность дуг

$$\sum_{k=1}^N x_{ij}^k(t) \leq 1, t \in [0, T_x]. \quad (1)$$

Таким образом, суммарные потоки единиц продуктов k на цепях C_k порождают суммарный поток с величиной N . Цепь C_k описывается функцией $x_{ij}^k(t)$, которая повсюду равна 0, кроме интервала, на котором S_k использует сервис $s(i, j)$, обозначим этот интервал $[T_{\text{beg}}^k(i, j), T_{\text{end}}^k(i, j)]$, тогда

$$x_{ij}^k(t) = \begin{cases} 1, & t \in [T_{\text{beg}}^k(i, j), T_{\text{end}}^k(i, j)], \\ 0, & \text{иначе,} \end{cases} \quad (2)$$

$$T_{\text{end}}^k(i, j) - T_{\text{beg}}^k(i, j) \geq T_{\text{exec}}(i, j). \quad (3)$$

Необходимо учесть условие баланса потока в сети, для этого определим индикаторы атомов, зависящие от сервиса k и от дуги (i, j) . Если атом $s(i, j)$ был задействован при композиции S_k , то значение индикатора положим равным единице

$$\bar{x}_{ij}^k = \begin{cases} 1, & \exists t_0 | x_{ij}^k(t_0) > 0, \\ 0, & \text{иначе.} \end{cases} \quad (4)$$

Индикаторы позволяют ввести баланс потока в узлах сети, для любого узла n и любого сервиса k выполняется условие:

$$\sum_{i - \text{входящий}} \bar{x}_{in}^k - \sum_{j - \text{выходящий}} \bar{x}_{nj}^k = \begin{cases} -1, & k = s, \\ 1, & k = f, \\ 0, & \text{иначе.} \end{cases} \quad (5)$$

Это означает, что в промежуточных узлах поток, входящий в узел, равен выходящему.

Уточним критерии качества задачи. Пусть для каждого сервиса S_k имеем L целевых функций, которые взвешенной суммой с положительными весами преобразованы в целевую функцию сервиса S_k . Повторением операции взвешенной суммы получаем единственную целевую функцию $F(C_1, C_2, \dots, C_N)$ для всех сервисов S_k . Так как все исходные критерии являются суммой параметров на $s(i, j)$ и монотонно возрастают, то это свойство сохраняется и для $F(\cdot)$. Целевая функция является линейной

$$\min F(\bar{x}) = \sum_{k=1}^N \sum_{(i,j) \in N} c_{ij} \bar{x}_{ij}^k. \quad (6)$$

Ставится задача \mathcal{P} найти функции $x_{ij}^k(t)$, минимизирующие целевую функцию (6) при ограничениях (1) – (5) и дополнительным ограничением на время T_*^k завершения работы сервисов цепей C_k , если $C_k = ((i_{k1}, i_{k2}), \dots, (i_{k(m-1)}, i_{km}))$, то

$$T_{\text{end}(i_{k(m-1)}, i_{km})}^k \leq T_*^k. \quad (7)$$

Метод решения

В заданном объединении сетей G_k $k = 1, 2, \dots, N$, требуется построить N полных цепей C_k , каждая из которых принадлежит сети G_k , цепь k соответствует календарному плану работ выполнения сложного сервиса S_k . Календарные планы дают решение задачи \mathcal{P} . Задача композиции одного сервиса сводится к построению кратчайшей цепи, в модели дискретного программирования матрица ее ограничений унимодулярна, и она решается одним из стандартных методов теории сетей (см. [2], стр. 50).

При $N > 1$ между сложными сервисами возникает конкуренция за атомы и задача усложняется. Дискретная модель этой задачи содержит ограничения (1), (2) на функции, поэтому нельзя говорить о матрице ограничений [2].

Покажем, что к решению задачи \mathcal{P} можно применить метод динамического программирования [3]. Метод можно применять к задаче, если на ее решениях выполняется принцип оптимальности: сужение оптимального решения от любой его точки до конца процесса само является оптимальным решением с началом в данной точке. В задаче \mathcal{P} рассматриваются цепи, связывающие узлы t_s и t_f . Промежуточной точкой решения может служить произвольный разрез сети. Пусть имеем оптимальное решение задачи \mathcal{P} . Выполним разрез и удалим его дуги, выберем компоненту, содержащую узел t_f . Тогда сужением задачи построения полных циклов будет задача построения полных циклов на подсети из заданных узлов с заданными начальными условиями на исходные события. Очевидно, что ввиду суммарного представления целевой функции (6) сужение оптимального решения задачи \mathcal{P} на подсети даст оптимальное решение задачи для подсети, иначе исходное оптимальное решение задачи \mathcal{P} не оптимально. Таким образом, верно следующее утверждение.

Утверждение. Решение задачи \mathcal{P} удовлетворяет принципу динамического программирования.

Применение этого метода возможно, однако для реальных задач требует большого

перебора вариантов. Применение этого метода возможно, однако он требует большого перебора вариантов, сложность алгоритма экспоненциальна. Задачи небольшого размера можно решать методом динамического программирования. Здесь предлагается эвристический алгоритм решения задачи, основанный на методе параллельного расписания в классической задаче календарного планирования с ограничениями на ресурсы [1]. Алгоритм последовательно добавляет к каждой цепи C_k следующий атом, выбирая его как максимальный по целевой функции из допустимых исходящих дуг с текущим свободным резервом в заданном интервале. В случае если такой не найден, то выбирается наиболее ценный атом из тех, которые не нарушают ограничение на ресурс времени. Если же нет и таких работ, то делается шаг назад с удалением из рассмотрения пройденной дуги.

Пусть $SET = \{t_{i1}, t_{i2}, \dots, t_{iN}\}$ – текущие завершающие события уже найденных начальных дуг цепей C_k . На каждом шаге алгоритма оценивается возможность продолжения цепей из текущих состояний, то есть проверяются текущие критические пути из SET и строится новое множество SET с изменением всех его элементов.

Эвристический алгоритм решения задачи \mathcal{P} .

Вход: Граф \mathcal{G} , объединяющий сети \mathcal{G}_k , $k = 1, \dots, N$, вектор директивных сроков T^* , целевая функция $F(\cdot)$ (см. (6)).

Выход: Цепи C_k сетей \mathcal{G}_k , $k = 1, \dots, N$, удовлетворяющие директивным срокам и приближающие целевую функцию $F(\cdot)$.

1. Положить все элементы SET равными t_s .
2. Цикл по выбору следующей дуги цепи C_k . Положить $k = 1$.
3. Если $t_{ik} = t_f$, то перейти к 7.
4. Найти критический путь в сети \mathcal{G}_k с начальным узлом t_{ik} из множества SET , по заданному T_*^k вычислить полный резерв R на событии t_{ik} . Если нет выбора дуг для поиска критического пути, то это означает, что нет цепи C_k , удовлетворяющей ограничениям задачи, выдать сообщение «нет решения для сервиса k », перейти к 8. Если полный резерв отрицательный, то есть нарушен директивный срок, то перейти к 6.
5. На множества дуг сети \mathcal{G}_k , исходящих из t_{ik} , выбрать дуги, удовлетворяющие ограничениям задачи по критическому пути с обязательным условием включения рассматриваемой дуги, из этих дуг выбрать дугу с наименьшим значением целевой функции, пусть эта дуга ведет в узел j . Заменить SET t_{ik} на t_j , перейти к 7.
6. Из узла t_{ik} вернуться к предыдущему узлу, удаляя дугу перехода из дальнейшего

рассмотрения.

7. Положить $k = k++$, переходя к следующей цепи C_k .

8. Конец.

Эвристика алгоритма используется на шаге 5 выбора очередной дуги. Предложенный выбор гарантирует выполнение ограничений, но может не давать оптимального значения целевой функции. Сложность представленного алгоритма линейна, если гарантировано ограниченное применение шага 6. Таким образом, алгоритм имеет псевдолинейную сложность.

Обсуждение результата и заключение

Для сравнения двух методов были проведены 10 экспериментов с различными числами сервисов и атомов, в каждом эксперименте были случайным образом сгенерированы и решены по 50 задач. В таб. 1 приведены основные параметры задач и полученные результаты.

Таблица 1. Параметры задач и сравнение методов ДП и ЭМ

№	Колич. сервисов	Кол. атомов в сетях		Значение критерия		Отклонение критерия (%)	Изменение времени работы (%)
		G_k	G	ДП	ЭМ		
1	3	14	32	41	44	7.3	5.4
2	5	16	65	54	59	9.3	8.1
3	8	18	72	58	67	10.5	11.7
4	10	21	120	84	92	9.5	12.2
5	12	22	155	102	114	11.8	9.1
6	15	24	220	156	184	12.9	15.2
7	18	25	318	220	312	13.8	34.4
8	20	28	416	282	321	13.6	36.3
9	23	30	450	355	413	14.1	38.2
10	25	31	480	392	484	13.2	48.6

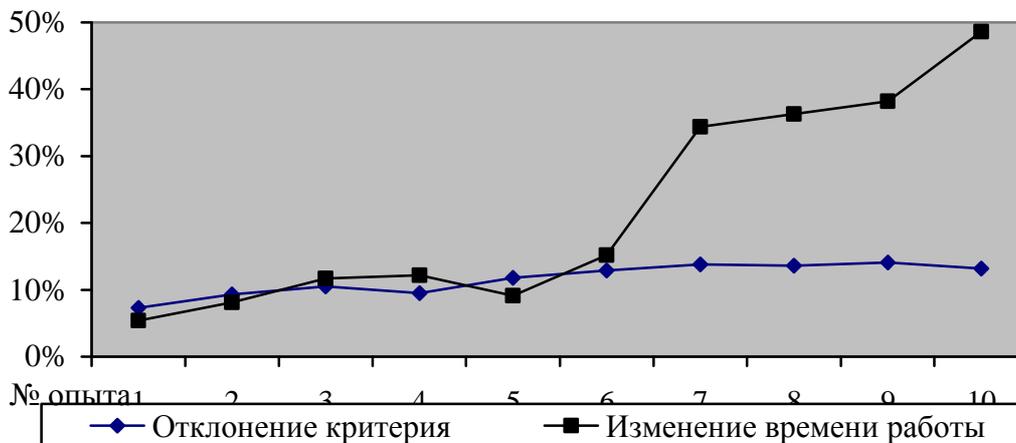


Рисунок 1. Сравнение отклонения критерия и изменения времени работы

Сравнивались 2 метода решения, метод динамического программирования (ДП), который дает точное решение и предложенный эвристический метод (ЭМ), который дает

приближенное решение. Сравниваются показатели «Отклонение критерия» и «Изменение времени работы» методов ДП и ЭМ. Результаты показывают (таб. 1 и рис. 1), что эвристический метод быстрее работает во всех случаях, разрыв во времени увеличивается с увеличением числа узлов сети, но при этом уменьшается точность полученных решений задачи.

Список литературы

1. Кочетов Ю.А, Столяр А.А Новые жадные эвристики для задачи календарного планирования с ограниченными ресурсами // Дискретный анализ исследования операций. – 2005. – Т. 12, № 1. – С. 12–36.
2. Лежнев А.В. Динамическое программирование в экономических задачах. – М.: Бином, Лаборатория знаний, 2010. – 176 с.
3. Филлипс Д., Гарсиф-Диас А. Методы анализа сетей. – М.: Мир, 1984. – 496 с.
4. Alonso G., Casati F., Kuno H., Machiraju, V. Web Services. Concepts, Architectures and Applications. – Berlin: Springer, 2004. – 340 p.
5. Alrifai M., Risse T. Combining Global Optimization with Local Selection for Efficient QoS-aware Service Composition. Proceedings of the International World Wide Web Conference. – Madrid, Spain, 2009. – P. 881–890.
6. Alrifai M., Risse T., Dolog P., Nejdl W. A Scalable Approach for QoS-based Web Service Selection. – Berlin: Springer, 2009. – Vol. 5472. – P. 190–199.
7. Yu T., Lin K-J. Service selection algorithms for Web services with end-to-end QoS constraints. IEEE Proceedings of International Conference on e-Commerce Technology (CEC 2004). – California, USA, 2004. – P. 129–136.
8. Zeng L., Benatallah B., Ngu A.H.H., Dumas M., Kalagnanam J., Chang, H. QoS-aware middleware for web services composition // IEEE Transactions on Software Engineering. – 2004. – Vol. 30, N. 5. – P. 311–327.

Рецензенты:

Татур М.М., д.т.н., профессор, заведующий кафедрой электронных вычислительных машин, Белорусский государственный университет информатики и радиоэлектроники, г. Минск.

Садыхов Р.Х., д.т.н., профессор кафедры электронных вычислительных машин, Белорусский государственный университет информатики и радиоэлектроники, г. Минск.