УДК 004.6: 004.75

МИНИМИЗАЦИЯ ТРАФИКА В ОБЛАЧНОЙ ИНФРАСТРУКТУРЕ

Суханов Владимир Иванович
д.т.н., доцент

Уральский федеральный универсистет, Екатеринбург, Россия

Приводится постановка задачи планирования избыточного размещения данных по узлам облака для минимизации пересылки информации

Ключевые слова: ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ, ПЛАНИРОВАНИЕ, ТРАФИК

UDC 004.6: 004.75

MINIMIZING TRAFFIC IN THE CLOUD INFRASTRUCTURE

Sukhanov Vladimir Ivanovich
Dr.Sci.Tech., associate professor

Ural Federal University, Ekaterinburg, Russia

The article presents formulation of the scheduling problem of excessive placement of data on cloud nodes to minimize the information traffic costs

Keywords: CLOUD COMPUTING, SCHEDULING PROBLEM, TRAFFIC

Облачные вычисления (Cloud Computing) – новое направление компьютерных технологий, которое решает много организационных, юридических и финансовых проблем пользователям, но требует от провайдеров предоставление широкого спектра сервисов и технологий, покрывающих возможные запросы со стороны клиентов. Облачные технологии основаны на инфраструктуре аппаратных и программных средств, содержащих набор вычислительных машин – узлов, объединенных локальной сетью передачи данных для приватных и сетью Интернет для публичных облаков. Организуют работу таких структур провайдеры, предоставляющие клиентам вычислительные услуги. Провайдер может располагать как собственной инфраструктурой, так и арендовать ее у поставщика IaaS.

Общая схема взаимодействия платформы с арендованной инфраструктурой и приложениями при предоставлении услуг конечным пользователям приведена на рисунке 1. Провайдер по контракту с предприятием предоставляет разработчикам его приложений и сервисов инструментарий для развертывания приложений в облаке, обеспечивая масштабирование за счет развертывания дополнительных копий приложения на виртуальных машинах облачной инфраструктуры и

последующую балансировку их загрузки запросами, поступающими от пользователей.

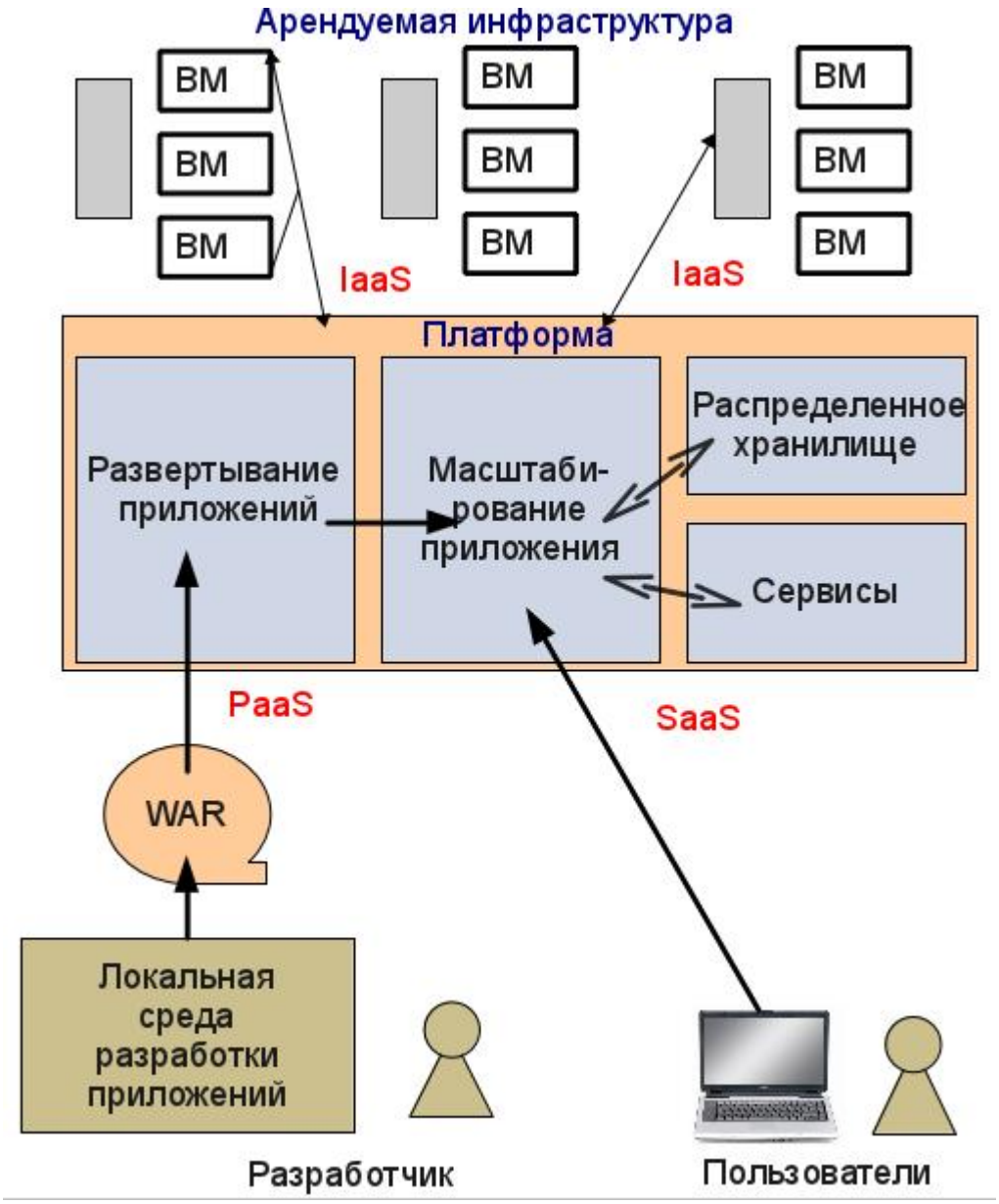

Рисунок 1 — Схема взаимодействия подсистем и пользователей облачной платформы

Эффективность предоставления услуг клиенту определяется во многом объемом накладных расходов на организацию вычислений на узлах

сети и пересылки данных между ними. Если быстродействие узлов легко наращивается установкой многоядерных процессоров, то скорость пересылки данных не может варьироваться в широких пределах по техническим причинам. Как правило, скорость пересылки данных существенно ниже скорости их обработки процессором узла и пересылка данных становится узким местом. Объем сетевого трафика во многом определяет задержки в обслуживании клиентов и стоимость услуг в целом. Поэтому актуальной является задача эффективного планирования размещения клиентских приложений по узлам сети, позволяющего уменьшить накладные расходы, включая задержки на транспортировку данных и оплату сетевого трафика.

Задачи распределения ресурсов для оптимизации размещения виртуальных машин и статического распределения потока запросов к серверам в инфраструктуре облака носят нелинейный комбинаторный характер. Нелинейными, как правило, являются ограничения и критерии качества принятия решений, а комбинаторным является множество, из которого эти решения выбираются. Решением таких задач является размещение набора объектов в позициях формируемого плана.

Для задач организации вычислительного процесса в сложных сетях облачной инфраструктуры объектами являются виртуальные машины, программы, наборы данных, заявки; позициями являются узлы, устройства памяти, место в очереди на исполнение. Критерии – стоимость и качество предоставления услуг.

Комбинаторные задачи с нелинейными критериями относятся к классу NP-полных задач (Nondeterministic Polinomial) с экспоненциальной оценкой времени их решения [1]. Точное решение таких задач можно получить перебором всех допустимых распределений объектов по позициям, но затраты времени будут неприемлемы для нужд практики. Аналогичную оценку сложности имеют точные методы, основанные на

методе ветвей и границ, из-за высокой сложности получения нижних (верхних) граней критерия качества, связанной с нелинейностью целевых функций и ограничений и, как следствие, возрастающим числом возвратов при выборе направления ветвления.

Математическая модель задачи планирования распределения приложений по узлам облачной инфраструктуры позволяет классифицировать сложность ее решения и наметить пути поиска инженерных методов реализации алгоритма. В нашем случае имеется n приложений, которые необходимо разместить по m узлам облака. Приложения требуют для своей работы вектор ресурсов

$$a_i = (a_{i,1}, \dots, a_{i,k})^T, i = 0, \dots, n-1. \quad (1)$$

Содержательно потребляемыми ресурсами являются внешняя память, размещаемые на узлах нужные пакеты прикладных программ, серверов, сервисов и библиотек, интенсивность поступающих запросов и степень загрузки центральных устройств, интенсивность межмашинного взаимодействия через сетевые адаптеры и другие параметры, критичные в той или иной конфигурации технических и программных средств узла. Потребление ресурсов может измеряться количественно, например, объем внешней памяти, загрузка процессора, так и качественно, например, наличие нужного пакета.

Машины узлов располагают ресурсами в объемах

$$b_j = (b_{j,1}, \dots, b_{j,k})^T, j = 1, \dots, m. \quad (2)$$

Как и для потребителей ресурсов все соглашения о их номенклатуре и предоставляемых объемах остаются в силе.

Приложения при своей работе требуют пересылки данных друг-другу в среднем в единицу времени в объемах, задаваемых матрицей

$$C = \{c_{ij}\}, i = 1, \dots, n; j = 1, \dots, n. \quad (3)$$

Матрица C не обязательно должна быть симметричной, что отражает возможную неравномерность пересылки данных в разных направлениях приложениями.

Для обеспечения надежности и живучести каждое приложение должно быть размещено в сети в p копиях. Определим матрицу переменных, определяющих размещение приложений в узлах сети следующим образом:

$$X = (x_{ij}),\ \ i = 0,\dots,n-1,n,\dots,2\times n-1,\dots,p\times n-1;\ \ j = 1,\dots,m \tag{4}$$

Переменная $x_{ij} \in \{0,1\}$ принимает значение 1, если приложение с номером i размещается в узле j, в противном случае – 0. Число элементов столбца матрицы размещения равно числу приложений, помноженному на число их копий. Для целей удобства записи номеров приложений и их копий, используемой в следующих конструкциях, нумерация строк начинается с нуля. Число столбцов равно числу узлов сети и их нумерация начинается с единицы.

Требование размещения каждой копии приложения точно на одном узле формулируется следующим условием:

$$\sum_{j=1}^{m} x_{ij} = 1,\ \ i = 0,1,\dots,n-1,n,\dots,2\times n-1,\dots,p\times n-1. \tag{5}$$

Запрет размещения нескольких копий приложения на одном узле будет выглядеть так:

$$x_{ij}\times x_{(i+n\times r)j} = 0,\ i = 1,\dots,n;\ j = 1,\dots,m;\ r = 1,\dots,p-1. \tag{6}$$

То есть, если какая-то копия i-го приложения размещена на узле j, то другая копия того же приложения не может быть на том же узле: произведение соответствующих индикаторов присутствия равно нулю.

Объем требуемых ресурсов приложениями, размещаемыми в узле, не должен превышать его возможностей, что формулируется следующим условием:

$$\sum_{j=0}^{p\times n-1} x_{iu} \times a_{(i \mod n),r} \le b_{ur}, r = 1,\dots,k; u = 1,\dots,m, \quad (7)$$

где $(a \bmod b)$ – остаток от a по модулю b.

Для принятого размещения X приложений объем трафика между парой узлов (u, v) будет вычисляться суммой всех пересылок данных во всех направлениях между всеми парами приложений, расположенных на этих узлах:

$$f(u,v) = \sum_{i=0}^{p\times n-1} \sum_{j=0}^{p\times n-1} x_{iu} \times x_{jv} \times c_{(i \bmod n)(j \bmod n)}. \quad (8)$$

Целевая функция задачи оптимального распределения приложений по узлам облака будет определяться суммой трафика между всеми парами узлов облака:

$$F(X) = \sum_{u=1}^{m} \sum_{v=u+1}^{m} f(u,v) = \sum_{u=1}^{m} \sum_{v=u+1}^{m} \sum_{i=0}^{p\times n-1} \sum_{j=0}^{p\times n-1} x_{iu} \times x_{jv} \times c_{(i \bmod n)(j \bmod n)}. \quad (9)$$

Таким образом, имеем поиск решения (4) задачи булевского комбинаторного программирования с нелинейными ограничениями (5 – 7) и нелинейной целевой функцией (9). Это хорошо известная проблема о разрезах в графе, точное решение которой имеет сложность NP-полной задачи [1]. Для получения решения в обозримое время с достаточной инженерной точностью приходится прибегать к эвристическим алгоритмам, не гарантирующим контролируемую погрешность.

Первая проблема – наличие и поиск допустимых решений, удовлетворяющих ограничениям (5 – 7). В реальных условиях коммерческих услуг со стороны провайдера инфраструктуры облака обычно имеется достаточный запас критических ресурсов, определяющих самое важное ограничение задачи, задаваемое системой неравенств (7). Остальные ограничения являются техническими и выполнимы всегда. В этих условиях проблема наличия и поиска допустимых решений не составляет труда.

Допустимый план **X** может удовлетворять кроме ресурсных еще ряду условий:

- декларациям, определяющим обязательные закрепления приложений;
- рекомендациям по диапазонам допустимых для приложений узлов;
- несовместимости приложений друг с другом.

Поиск допустимого решения будем вести последовательно. Начальным состоянием процесса построения решения является частичный план X_0, учитывающий декларативные закрепления. Допустимые планы задачи образуют множество R конечных состояний процесса построения решения.

Для элементов $X \in R$ определим расстояние от границы множества R:

$$r(X) = \min_{j,u}\left(b_{ju} - \sum_{i=1}^{n} a_{ju} x_{ij} \right)$$

Центрами в R будут планы X^o, для которых $r(X^o) = max\, r(X)$. Планы центра X^o обладают максимальным запасом устойчивости при изменении внешней обстановки при их реализации. Для сравнительной оценки состояния ресурсов целесообразно перейти к нормированной величине потребления ресурсов позициями:

$$\Delta_u(X) = \max_{j}\left(\sum_{i=1}^{n} a_{ju} x_{ij} \Big/ b_{ju} \right).$$

Шагом процесса j будет закрепление одного объекта в некоторой позиции плана. Выбор j на шаге s переводит процесс из состояния x_s в состояние $j(x_s)$. Состояние x' достижимо из x ($x \rightarrow x'$), если существует последовательность шагов $j_1, \ldots, j_p$, переводящая x в x'. Множество состояний R_g достижимое из R_f ($R_f \rightarrow R_g$), если из каждого состояния $x \in R_f$ достижимо некоторое $x' \in R_g$.

Пусть $\overline{R}(j)$ – множество допустимых планов, достижимых из позиции $j(x_s)$. Оценка

$$d(j) = \max_{u, x \ddot{e} \overline{R}(j)} |\Delta_u(x) - \Delta_u(x^*)|$$

характеризует близость множества $\overline{R}(j)$ к центру X^o совокупности R. Экстремальным переходом из позиции x_s будем называть выбор j^o из условия

$$d(j^o) = \min_{j \in \Phi_s} d(j).$$

Глубиной рефлексии r алгоритма поиска решения будем называть число подмножеств $\{x_s\} \to R_{s+1} \to \mathbf{K} \to R_{s+r}$, просматриваемых на каждом шаге s. Каждое R_{s+i} может быть представлено разбиением

$$R_{s+i} = R'_{s+i} \mathbf{U} R''_{s+i}, \quad R'_{s+i} \mathbf{I} R''_{s+i} = \varnothing.$$

Таким образом, каждое подмножество содержит состояния, из которых можно построить допустимый план. Оценка свойства непустоты R_{s+r} может быть произведена с ростом r с большей точностью. Ввиду ограниченных вычислительных возможностей ЛПР фиксирует величину r, например, полагая $r = 1$.

Рассмотрим случай (для больших значений r полученные результаты распространяются индуктивно), когда ЛПР затрачивает минимальное время на решение задачи. При этом он располагает знаниями на один шаг процесса: текущее состояние x_s, возможные управления $j \in \Psi_s$ и множество $\{x_{s+1} = j(x_s), \forall j \in \Psi_s\}$ достижимых состояний. Для всех x_{s+1} зададим нечеткую функцию $h(x_{s+1})$, определяющую степень

принадлежности x_{s+1} целевому множеству R'_{s+1}. Воспользуемся эвристической гипотезой: величина $h(x_{s+1})$ является неубывающей функцией от размера запаса $1-D(x_{s+1})$ ресурсов. Следовательно, максимизация запасов на шаге s способствует достижению целевого множества R. Свойства объектов и позиций определяют различную зависимость между распределяемыми ресурсами. В связи с этим рассмотрим следующие ситуации.

Построение начального допустимого плана может требовать значительного трудно прогнозируемого времени при больших значениях глубины рефлексии. Здесь сказывается экспоненциальный рост числа просматриваемых вершин дерева решений. Поэтому в инженерной практике целесообразно ориентироваться на алгоритмы с контролируемым временем работы. Таким алгоритмом может быть последовательное улучшение плана [2, 3] за счет перестановки приложений в узлах.

Выберем произвольным способом некоторое подмножество приложений $E \subset \{e_1, \dots, e_n\}$ небольшой мощности, например, $|E| \leq 3$. При небольшом числе объектов в E можно перебором найти их наилучшее расположение по узлам сети, не затрагивая позиций остальных приложений. Число проб в этом случае не превышает $m^{|E|}$, что вполне приемлемо для современных ЭВМ. Поиск их нового размещения следует выполнять с учетом ограничений (5 – 7) задачи. Если новое размещение улучшает старое, то заменяем им старое и продолжаем попытки улучшить план. Если все попытки не приводят к улучшению, то заканчиваем поиск решения основной задачи.

При отборе групп объектов следует учитывать мощность E, существенно влияющую на объем проделываемой работы. Во-первых, с увеличением числа объектов увеличиваются затраты времени алгоритма, часто нелинейно, как полином высокой степени. Во-вторых, число попыток

растет как число сочетаний C_n^h, где n – общее число объектов; $h = |E|$ – мощность группы. Поэтому часто из соображений экономии времени ограничиваются величиной $h = 2$.

Описанный подход удобен тем, что позволяет остановить процесс улучшения в любой момент по истечению отпущенного бюджета времени на поиск решения. В силу экспоненциальной сложности общей задачи (3 – 9) полный процесс может продолжаться трудно обозримое время, существенно зависящее от размерности задачи, задаваемой константами n, m, p, h.

Последнее замечание. Получаемый в результате последовательного процесса неулучшаемый план существенно зависит от начального допустимого плана. Описанный процесс аналогичен известным методам управляемого спуска, в частности, градиентному. Результатом такого процесса является попадание в локальный минимум целевой функции. Если у ЛПР есть достаточный ресурс времени, то целесообразно повторять процесс поиска для различных начальных планов, выбрав из всех локальных решений наилучшее.

Список литературы

1. Гэри М., Джонсон Д. Вычислительные машины и трудно решаемые задачи. – М.: Мир, 1982. – 416 с.
2. Михалевич В.С., Кукса А.И. Методы последовательной оптимизации в дискретных задачах оптимального распределения ресурсов. – М., :Наука, 1983. – 208 с.
3. Емеличев В.А., Комлик В.И. Метод построения последовательности планов для решения задач дискретной оптимизации. – М.: Наука, 1981. – 207 с.

Работа поддерживается Министерством образования и науки Российской Федерации, ГК 07.514.11.4026 (шифр 2011-1.4-514-104-009)