УДК 519.687.1

А.И. Ролик, М.Е. Боданюк

АЛГОРИТМ ПОСЛЕДОВАТЕЛЬНОГО АНАЛИЗА ВАРИАНТОВ В ЗАДАЧЕ РАСПРЕДЕЛЕНИЯ ВИРТУАЛЬНЫХ МАШИН В ЦЕНТРЕ ОБРАБОТКИ ДАННЫХ

Аннотация: Предложен алгоритм последовательного анализа вариантов для решения задачи распределения виртуальных машин по физическим серверам в центрах обработки данных. Приведены результаты экспериментальных исследований для задач различной размерности. Проведено сравнение предложенного алгоритма с эвристическими и генетическим алгоритмами. Даны рекомендации по использованию алгоритма последовательного анализа вариантов для решения задачи распределения виртуальных машин.

Ключевые слова: алгоритм последовательного анализа вариантов, центр обработки данных, распределение ресурсов.

Введение

Технология виртуализации вычислительных ресурсов является неотъемлемым атрибутом современных центров обработки данных (ЦОД). Содержание ИТ-инфраструктуры ЦОД требует значительных финансовых затрат, что делает актуальной задачу эффективного управления распределением ресурсов. Максимально плотное размещение виртуальных машин (ВМ) на физических серверах позволяет снизить эксплуатационные затраты и обслужить большее количество пользователей серверным парком ЦОД. Поэтому данная работа, посвященная решению задачи оптимизации размещения ВМ по физическим серверам, является актуальной.

Анализ предыдущих публикаций

Задачу распределения ВМ по серверам можно решать с помощью классического генетического алгоритма (ГА) [1]. В [2, 3] показано, что применение управляемого генетического алгоритма позволяет повысить качество решения и контролировать ход решения на каждой итерации алгоритма. Для распределения ВМ можно применять предложенные в [1, 2] эвристические алгоритмы. Там же доказано, что эффективность эвристических алгоритмов зависит от структуры алгоритма и параметров размещаемых ВМ. Однако в этих работах не предложен алгоритм нахождения оптимального решения задачи размещения ВМ.

Цель работы

Целью работы является разработка и определение целесообразности применения алгоритма последовательного анализа вариантов (ПАВ) для решения задачи размещения ВМ на физических серверах в ЦОД.

Математическая модель

ЦОД содержит упорядоченное множество физических серверов $N = \{N_1, \ldots, N_n\}$, где n — количество серверов;

подлежит распределению упорядоченное множество виртуальных машин $K = \{K_1, \ldots, K_m\}$, где m — количество ВМ;

каждый сервер N_i, $i = \overline{1,n}$, характеризуется параметрами: Ω_i — производительность процессора; Γ_i — емкость оперативной памяти сервера N_i;

каждая ВМ K_j, $j = \overline{1,m}$, запрашивает ω_j процессорного времени и γ_j оперативной памяти;

матрица $R = ||r_{ji}||$ размером $m \times n$ задает распределение ВМ по серверам, причем

$$r_{ji} = \begin{cases} 1, & \text{если виртуальная машина } K_j \text{ располагается на сервере } N_i, \\ 0 - & \text{в противном случае.} \end{cases}$$

Матрица R является решением задачи и определяет распределение множества K ВМ на множестве N физических серверов.

Постановка задачи

Если все серверы множества N имеют идентичные технические характеристики, то можно считать, что $\Omega_i = 1$ и $\Gamma_i = 1$, $\forall i = \overline{1,n}$, т. е.

$$\{\Omega_i,\ \Gamma_i\}|_{N_i} = \{1,\ 1\}, \quad i = \overline{1,n}. \tag{1}$$

Здесь сделан переход от измерения ресурсов серверов в абсолютных единицах, когда память измеряется в МБ, а частота процессора в МГц, к относительным. Тогда потребности в ресурсах ВМ из множества K определяются как нормированная часть ресурсных возможностей сервера.

В статье не рассматривается задача кластеризации, поэтому требования к ресурсам каждой ВМ из множества K не могут превышать возможностей сервера

$$\omega_j \leqslant 1 \text{ и } \gamma_j \leqslant 1,\ i = \overline{1,n}. \tag{2}$$

При решении задачи распределения ВМ для всех серверов множества N должно выполняться следующее ресурсное ограничение:

$$\sum_{j=1}^{m} r_{ji}\omega_j \leqslant 1 \text{ и } \sum_{j=1}^{m} r_{ji}\gamma_j \leqslant 1,\ \forall i = \overline{1,n}. \tag{3}$$

Введем вектор $\overline{y} = \langle y_i \rangle$, $i = \overline{1,n}$, где

$$y_i = \begin{cases} 1, \text{ если на сервере } N_i \text{ располагается хотя бы одна ВМ,} \\ 0 - \text{ в противном случае.} \end{cases}$$

Тогда критерием оптимальности при решении задачи размещения ВМ на физических серверах будет:

 ISSN 1560-8956

$$\min \sum_{i=1}^{n} y_i, \tag{4}$$

т. е. сервера должны заполняться виртуальными машинами таким образом, чтобы было задействовано минимальное количество серверов.

Такой подход позволяет минимизировать суммарные затраты S на обслуживание и энергоснабжение парка серверов ЦОД. Целевую функцию можно представить следующим образом:

$$S = s \sum_{i=1}^{n} y_i, \tag{5}$$

где s — стоимость поддержания и затраты на энергоснабжение для одного сервера.

Тогда постановка задачи может звучать в следующей интерпретации. ВМ необходимо размещать на серверах ЦОД таким образом, чтобы выражение (5) достигало минимального значения.

Алгоритм последовательного анализа вариантов распределения виртуальных машин

Алгоритм ПАВ относится к комбинаторным методам решения задач [4]. Основная идея комбинаторных методов состоит в использовании конечности множества допустимых решений и замене их полного перебора сокращенным (направленным перебором). Общая схема последовательного анализа вариантов разработана Михалевичем В.С. [5] на основе идей теории последовательных решений и динамического программирования. Методы ПАВ основаны на пошаговом конструировании решений — последовательном уточнении значений компонент решения и отсеве в процессе конструирования решений, которые не могут быть достроены до оптимальных.

Пусть имеется задача минимизации:

$$\min f(x_1, \ldots, x_m) = \min s \sum_{i=1}^{n} y_i, \tag{6}$$

при ограничениях вида:

$$\left(\sum_{j=1}^{s} (r_{j1}\omega_j) y_1, \ldots, \sum_{j=1}^{s} (r_{ji}\omega_j) y_n \right) \leqslant 1, \ \forall j = \overline{1,m}, \ i = \overline{1,n}, \tag{7}$$

$$\left(\sum_{j=1}^{s} (r_{j1}\gamma_j) y_1, \ldots, \sum_{j=1}^{s} (r_{ji}\gamma_j) y_n \right) \leqslant 1, \ \forall j = \overline{1,m}, \ i = \overline{1,n}, \tag{8}$$

$$x_j \in Q, \ \forall j = \overline{1,m}, \tag{9}$$

где $Q = \{q_{1_1 j}, \ldots, q_{\alpha_m j}\}$ — заданные конечные множества [6]. Каждая переменная $x = (j; i)$ для всех $j = \overline{1,m}$ выражает размещение j-ой виртуальной машины на i-ом сервере.

Вектор $\bar{x}_{(m)} = (x_1, \ldots, x_m)$ является решением, если его компоненты $x_j \in Q, \quad \forall j = \overline{1,m}$. Каждый элемент множества $q_{\alpha_m i}$ представляет собой одну из возможных компонент вектора $\bar{x}_{(m)}$.

Множество всех решений обозначим через Z. Решение называется допустимым, если оно удовлетворяет неравенствам (7) и (8). Множество всех допустимых решений обозначаем через Z_f. Вектор $\bar{x}_{(p)} = (x_1, \ldots, x_p)$, $p < m$ называется частичным решением [6], если $\bar{x}_{(p)} \in Q$. Если при этом вектор $\bar{x}_{(p)}$ может быть достроен до допустимого решения $(x_1, \ldots, x_p, x_{p+1}, \ldots, x_m)$, $\bar{x}_{(m)} \in Z_f$, то он называется допустимым частичным решением.

Множество всех решений задачи изображается в виде дерева, где пути от корня к вершинам дерева соответствуют частичным решениям, а пути из вершины x_0 в вершины x_m соответствуют полным решениям (см. рисунок 1).

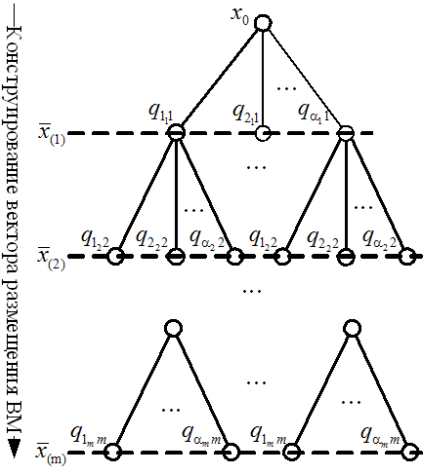

Рис. 1 – Дерево решений для алгоритма ПАВ

Для задачи, когда все сервера могут предоставить одинаковое количество ресурсов, целесообразно исключить из рассмотрения зеркальные решения. Например, если есть два идентичных сервера, количество возможных вариантов размещения ВМ можно сократить в два раза, поскольку перенос всех ВМ с первого сервера на второй не повлияет на объем задействованных ресурсов. Для серверов с одинаковой конфигурацией важно не абсолютное размещение ВМ на этих серверах, а их расположение одна относительно другой (см. рисунок 2). При увеличении количества серверов с одинаковой конфигурацией количество зеркальных комбинаций значительно возрастает от $(m! - 1)$ для случая размещения всех ВМ на одном сервере до $(m \times n - 1)!$ при размещении ВМ на разных серверах.

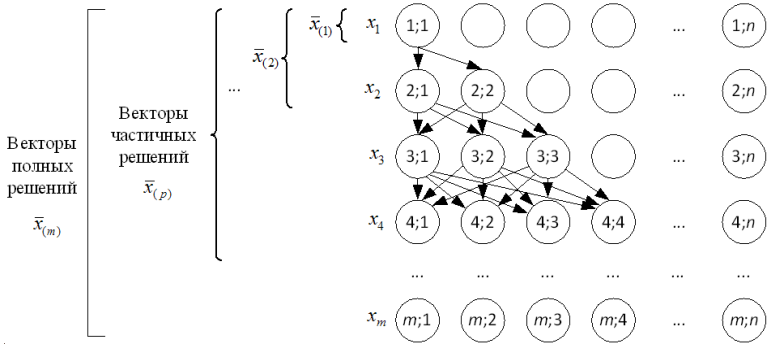

Рис. 2 – Размещение ВМ на серверах с помощью алгоритма ПАВ

Для разработки алгоритма необходимо определить набор правил выбора решений, которые будут развиваться на каждом шаге. Правила задаются в виде элиминирующих тестов, определяющих стратегию развития частичных решений. Определение набора элиминирующих тестов $\xi = \{\xi_l\}$ необходимо для отсева частичных решений, которые не могут быть достроены ни до допустимых, ни до оптимальных [6].

Пусть h — некоторое множество частичных решений. Тогда в упорядоченном множестве $\sigma(h) = \{\xi_0(h), \xi_1(h), \ldots, \xi_l(h)\}$ через $\xi_l(h)$ обозначается множество частичных решений, исключаемых тестом ξ_l.

Введем следующий набор элиминирующих тестов:

ξ_0 — анализ допустимости решений;

ξ_1 — сравнение допустимых решений по значению целевой функции;

ξ_2 — оценка оптимального значения γ_f целевой функции на каждом из множеств (вычисление нижней границы).

Тест ξ_0 позволяет исключить из рассмотрения недопустимые решения. Во множество недопустимых решений входят все решения, для которых количество размещенных на сервере ВМ превышает возможности сервера. Тест ξ_1 призван находить оптимальное распределение ВМ и исключить из рассмотрения неоптимальные допустимые решения. Тест ξ_2 и применение правила отсеивания зеркальных решений позволяет избежать полного перебора допустимых решений.

Определим элиминирующие тесты ξ_0, ξ_1, ξ_2 следующим образом:

$$\xi_0(h) = \{\overline{x}_{(p)} | \overline{x}_{(p)} \in h, \sum_{j=1}^{p} x_{ji}\omega_j > 1, \sum_{j=1}^{p} x_{ji}\gamma_j > 1, \forall\, i = \overline{1,n}\}, \qquad (10)$$

$$\xi_1(h) = \{\overline{x}_{(m)} | \overline{x}_{(m)} \in Z_f,\ f(x) \leqslant f(\bar{x}_{(m)})\}, \qquad (11)$$

$$\xi_2(h) = \{\overline{x}_{(p)} | \overline{x}_{(p)} \in h,\ \gamma_f(\bar{x}_{(p)}) > f^*\}, \qquad (12)$$

где f^*— верхняя граница для минимума задачи (6). Тест ξ_2 вычисляет нижнюю границу для оптимального значения целевой функции $f(\bar{x}^0_{(p)})$

на каждом из множеств $\mathrm{H}(\bar{x}^0_{(p)})$, элементами которых являются решения, продолжающие частичное решение $\bar{x}^0_{(p)}$.

На результаты ПАВ оказывает влияние порядок применения элиминирующих тестов и правил $U = (u_1, \ldots, u_q)$ выбора частичных решений. В алгоритме ПАВ для задачи распределения ВМ множество правил U задается следующими тремя правилами u_1, u_2, u_3:

Правило u_1. На каждом шаге алгоритма развиваются все частичные решения, построенные на предыдущем шаге, кроме решений, которые являются зеркальными отражениями уже рассмотренных решений.

Правило u_2. На каждом шаге алгоритма из частичных решений, полученных на предыдущих шагах, развиваются те, для которых достигается минимум целевой функции.

Правило u_3. Выбор кандидата на каждом шаге схемы осуществляется из множества $h_d = h_k - h_{k-1}$ частичных решений, полученных на предыдущем шаге, и только когда h_d не содержит частичных решений, необходимо обратиться ко всему списку.

Множество правил $U = (u_1,\ u_2,\ u_3)$ и тестов $\xi = \{\xi_0, \xi_1, \xi_2\}$ определяют стратегию перебора решений. Порядок применения правил и элиминирующих тестов соответствует индексам. В данном случае правила и тесты упорядочены по возрастанию сложности.

Результаты экспериментальных исследований

Особенности проведения исследований

Заявки на выделение множества ВМ в ЦОД поступают последовательно. Количество запрашиваемых ресурсов для каждой ВМ выбирается случайным образом. Соотношение требуемой оперативной памяти и процессорного времени распределено равномерно.

При проведении экспериментальных исследований осуществлялась генерация задаваемого количества множеств ВМ. Потребности ВМ в вычислительных ресурсах по оперативной памяти и процессорному времени сервера генерировались случайным образом в пределах 0.05—0.6 от возможностей сервера.

Результаты экспериментальных исследований

Эффективность работы алгоритма ПАВ оценивалась по времени, необходимому для нахождения оптимального решения в зависимости от размерности задачи m. Результаты работы алгоритма приведены на рисунке 3. Для сравнения показано время работы УГА [2, 3], а также эвристических алгоритмов “Т” и “М”, предложенных в [2].

На рисунке 3 видно, что алгоритм ПАВ показывает приемлемые результаты времени размещения на задачах малой размерности, когда количество ВМ не превышает 4—16. При дальнейшем увеличении количества ВМ время размещения алгоритмом ПАВ возрастает по экспоненциальной зависимости, в то время как эвристические и генетический алгоритмы демонстрируют линейный рост затрат времени.

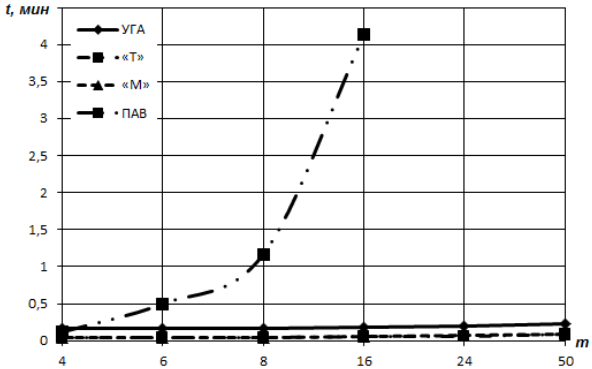

Рис. 3 – Зависимость времени работы алгоритмов от размерности задачи

Большое различие во времени работы алгоритма ПАВ и времени эвристических алгоритмов объясняется их различной природой. ПАВ является комбинаторным алгоритмом, а эвристические алгоритмы — модификацией сортировочных алгоритмов.

Произведен анализ погрешности работы алгоритмов, предложенных в [2]. Значение целевой функции S_T, S_M, $S_{УГА}$, полученное в результате работы эвристических алгоритмов “Т”, “М” и управляемого генетического алгоритма (УГА) соответственно, сравнивалось с оптимальным решением, обеспечиваемым ПАВ. На рисунке 4 по оси ординат отложена ошибка распределения ВМ, которая показывает, насколько больше серверов задействуется при распределении ВМ по алгоритмам “Т”, “М” и УГА, чем при оптимальном размещении. По оси абсцисс откладывается количество ВМ, подлежащих распределению на серверах ЦОД. Значения ошибки распределения усреднены по результатам десяти экспериментов.

Алгоритм ПАВ дает незначительный выигрыш по точности даже на задачах малой размерности. При увеличении размерности ошибка результатов эвристических и генетического алгоритмов возрастает, что свидетельствует о зависимости погрешности результатов работы этих алгоритмов от размерности задачи.

Стоит отметить, что для успешного применения алгоритма ПАВ необходимо располагать значением количества серверов $N_{\min}$, для которого точно известно, что на них могут быть размещены все ВМ из набора K_j, $j = \overline{1,m}$. Значение $N_{\min}$ может быть получено экспертным путем или с помощью одного из эвристических алгоритмов.

Исследования позволили сделать следующие рекомендации по выбору алгоритма распределения ВМ по физическим серверам. Для задач, где необходима высокая точность распределения при малом количестве ВМ, следует использовать алгоритм ПАВ, поскольку он находит оптимальное решение. ПАВ можно применять и тогда, когда время получе-

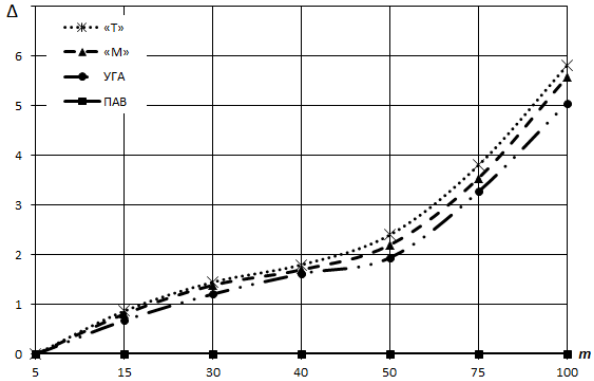

Рис. 4 – Зависимость ошибки распределения ВМ Δ относительно оптимального решения в зависимости от размерности задачи

ния результата не является критичным. Эвристические и эволюционные алгоритмы позволяют быстро получить решение. Скорость работы таких алгоритмов мало зависит от размерности задачи, однако качество решения уступает алгоритму ПАВ.

Выводы

Разработан алгоритм дискретной оптимизации на основе схемы ПАВ для решения задачи размещения виртуальных машин по физическим серверам в ЦОД. Оценены временные затраты работы алгоритма ПАВ в зависимости от размерности задачи. Произведен сравнительный анализ эффективности различных алгоритмов. Показано, что эвристические алгоритмы позволяют быстро получать решение, благодаря которому можно снизить затраты на поддержание серверов. Алгоритм ПАВ требует существенных затрат времени, но позволяет находить наилучшие решения.

Литература

1. Теленик С. Управління навантаженням і ресурсами центрів оброблення даних при віртуальному хостингу / С. Теленик, О. Ролік, М. Букасов, С. Андросов, Р. Римар // Вісн. Тернопільского держ. техн. ун-ту. — 2009. — Том 14. — № 4. — С. 198—210.
2. Теленик С.Ф. Управляемый генетический алгоритм в задачах распределения виртуальных машин в ЦОД / С.Ф. Теленик, А.И. Ролик, П.С. Савченко, М.Е. Боданюк // Вісник ЧДТУ. — 2011. — № 2. — С. 104—113.
3. Теленик С.Ф. Генетичні алгоритми вирішення задач управління ресурсами і навантаженням центрів оброблення даних / С.Ф. Теленик,

 ISSN 1560-8956

О.І. Ролік, М.М. Букасов, С.А. Андросов // Автоматика. Автоматизація. Електротехнічні комплексі та системі. — 2010. — №1 (25). — С. 106—120.

4. Сигнал И.Х. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: Учеб. пособие / И.Х. Сигнал, А.П. Иванова. — Изд. 2-е, испр. — М.: ФИЗМАТЛИТ, 2003. — 240 с.
5. Михалевич В.С. Последовательные алгоритмы оптимизации и их применение/ В.С. Михалевич // Кибернетика.— 1965.— №1.— С. 45—55; №2.— С. 85—88.
6. Сипко Е.Н., Метод последовательного анализа вариантов решения задачи составления расписания занятий/ Е.Н. Сипко // Искусственный интеллект. — 2011. — № 1. — С. 243 — 250.
7. Амосов А.А. Вычислительные методы для инженеров / А.А. Амосов, Ю.А. Дубинский, Н.П. Копченова. — М.: Высшая школа, 1994. — 544 с.
8. Бахвалов Н.С. Численные методы. / Н.С. Бахвалов, Н.П. Жидков, Г.Г. Кобельков. — 8-е изд. — М.: Лаборатория Базовых Знаний, 2000.— 624 с.
9. Волков Е.А. Численные методы/ Е.А. Волков. — М.: Физматлит, 2003. — 400 с.
10. Зайченко О.Ю. Дослідження операцій / О.Ю. Зайченко, Ю.П. Зайченко. — К.: Видавничій Дім "Слово". —2007. — 472 с.

Отримано 27.11.2012 р.