

Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра автоматизации систем вычислительных комплексов

Отчет по решению задания 3

Разработка и реализация алгоритма решения прикладной задачи и проведение его экспериментального исследования

Задание выполнил студент 421 группы
Алексей Ершов

Москва
2019

1 Содержательная постановка задачи	2
2 Математическая постановка задачи	3
3 Алгоритм решения задачи	4
3.1 Выбор жадного критерия	4
3.2 Сортировка	4
3.3 Размещение ВМ	4
3.4 Ограниченный перебор	5
3.5 Останов	5

1 Содержательная постановка задачи

Для выполнения задания требуется реализовать алгоритм для решения задачи распределения виртуальных машин (ВМ) по серверам ЦОД. В условии задания была представлена следующая содержательная постановка задачи.

Дано N заявок на размещение виртуальных машин в ЦОД. Каждая виртуальная машина характеризуется количеством запрашиваемых ядер и требуемым объемом оперативной памяти. Каждый сервер характеризуется количеством ядер и объемом оперативной памяти. Виртуальная машина должна полностью разместиться на одном сервере (т.е. ядра и память, выделенные виртуальной машине должны быть на одном и том же сервере).

Задача заключается в отыскании отображения из множества номеров заявок на размещение ВМ во множество номеров серверов, которое ставит в соответствие номеру заявки номер сервера, на котором данная ВМ будет размещена. Для некоторых номеров заявок такое отображение построить невозможно - это означает, что данную заявку алгоритм разместить не смог.

В качестве алгоритма был предложен алгоритм, сочетающий жадные стратегии и ограниченный перебор.

2 Математическая постановка задачи

Дано:

- $S = \{ \langle C_1, R_1 \rangle, \dots, \langle C_n, R_n \rangle \}$ - множество серверов, где пара $\langle C_i, R_i \rangle$ определяет характеристики i -го сервера, а именно: количество ядер и количество оперативной памяти в гигабайтах.
- $V = \{ \langle c_1, r_1 \rangle, \dots, \langle c_k, r_k \rangle \}$ - множество заявок на размещение виртуальных машин, где пара $\langle c_j, r_j \rangle$ определяет характеристики j -ой ВМ, аналогично характеристикам сервера.

Функция $M(i) = j$, если i -ая ВМ размещена на j -ом сервере и -1 , если i -ая ВМ не размещена.

Отображение M корректно \Leftrightarrow

$$\forall j \in [1, n] \sum_{i \in [1, k]: M(i)=j} c_i \leq C_j \ \& \ \forall j \in [1, n] \sum_{i \in [1, k]: M(i)=j} r_i \leq R_j.$$

Функция $D(i) = 1$, если $M(i) > 0$ и 0 , если $M(i) < 0$.

Целевая функция: $f_M = \sum_{i=1}^k D(i).$

Требуется найти: M , при котором достигается $\max f_M$, где максимум берётся по всем возможным M .

3 Алгоритм решения задачи

Для решения задачи будем использовать алгоритм, сочетающий жадные стратегии и ограниченный перебор. Опишем его схему.

3.1 Выбор жадного критерия

Первым шагом нужно определить жадный критерий: количество ядер или объём оперативной памяти.

Для этого сравним следующие 2 числа:

$$\rho_1 = \left(\sum_{i=1}^n C_i \right) / \left(\sum_{i=1}^k c_i \right)$$
$$\rho_2 = \left(\sum_{i=1}^n R_i \right) / \left(\sum_{i=1}^k r_i \right).$$

Если $\rho_1 < \rho_2$, то количество ядер суть жадный критерий. В противном случае - объём оперативной памяти.

3.2 Сортировка

Для простоты будем считать, что множества S и V суть списки.

На данном этапе мы сортируем эти 2 списка следующим образом: если количество ядер есть жадный критерий, то сортируем список S по убыванию значения C_i , а список V - по возрастанию c_j . Иначе - аналогично, но по параметрам R и r соответственно.

3.3 Размещение ВМ

На данном шаге алгоритм для $\forall j \in [1, n]$ в порядке возрастания j и для $\forall i \in [1, k]$ пытается “разместить” i -ую виртуальную машину на j -ом сервере, проверяя критерий корректности отображения M (для $\forall i \in [1, k]$, для которых соответствующая i -ая ВМ уже размещена). Иными словами, зададим новый список пар $\langle CL_m, RL_m \rangle$, $m \in [1, n]$, где CL_m суть сумма всех c_i , размещённых в данный момент на сервере с номером m , а RL_m - аналогично сумма всех таких r_i .

Если же для данного j критерий корректности не выполняется, то алгоритм пытается “разместить” i -ую ВМ на $(j+1)$ -ом сервере и т.д.

При успешном размещении всех ВМ, алгоритм завершается. Если алгоритм не может найти такой сервер, на котором можно разместить текущую заявку, то он переходит к п. 3.4

3.4 Ограниченный перебор

Пусть для ВМ с номером p не нашлось доступного для размещения сервера. Тогда алгоритм “откатывается” на один шаг назад, то есть “переразмещает” ВМ с номером $(p-1)$ на сервер с номером $(M_{p-1} + 1)$, где M_{p-1} - есть номер сервера, на котором до этого была размещена ВМ с номером $(p-1)$. При невозможности разместить данную ВМ на этот сервер алгоритм пытается разместить её на сервер с номером $(M_{p-1} + 1)$ и т.д. Если данную машину не удаётся разместить ни на один из серверов, то алгоритм возвращает начальное отображение номеров ВМ в номера серверов, а для ВМ-инициатора отката с номером p : $M(p) = -1$.

При удачном переразмещении алгоритм пытается вновь разместить ВМ с номером p уже на сервер номер M_{p-1} . Если алгоритм снова терпит неудачу, то совершается откат для ВМ с номером $(p-2)$ и т.д. Количество откатов ограничено параметром алгоритма λ .

3.5 Останов

Алгоритм заканчивает работу, когда для $\forall i \in [1, k]$ определено значение $M(i)$.