Нижние границы для динамической задачи упаковки в контейнеры с начальным размещением

Турнаев А.М.

Институт математики им. С.Л. Соболева СО РАН (Омский филиал)

Исследование выполнено за счет гранта Российского научного фонда № 22-71-10015-П

27.09.2025

Постановка задачи

1. Имеется потенциально неограниченное количество идентичных стоек.
2. Каждая стойка содержит заданное число идентичных серверов.
3. Каждый сервер имеет архитектуру неравномерного доступа к памяти (NUMA) с заданным количеством NUMA-узлов (2 или 4).
4. Каждый NUMA-узел имеет ограничения на два ресурса: ЦПУ и память. NUMA-узлы не являются однородными и могут отличаться.

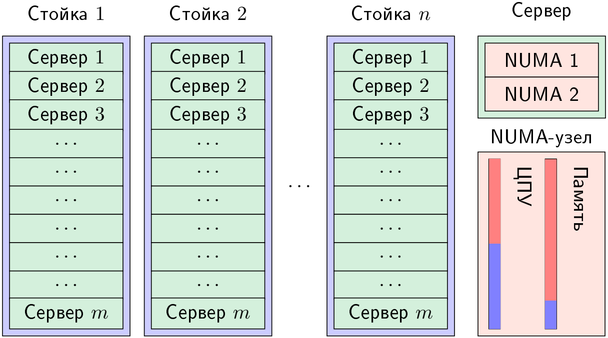


Рис. 1: Структура облачного центра

- Для предоставления пользователю вычислительных мощностей, в облачном центре аллоцируются виртуальные машины, то есть размещаются на соответствующей стойке, сервере и NUMA-узле. Виртуальные машины типизированы, а именно тип задает следующие характеристики:
  1. Ресурсные требования – использование ЦПУ и объем оперативной памяти, необходимые для запуска данной виртуальной машины;
  2. Размер – количество частей, из которых состоит данная виртуальная машина. Части виртуальной машины имеют одинаковые ресурсные требования и должны быть размещены на одном сервере, но на разных NUMA-узлах. Мы рассматриваем два размера ВМ: малые (1 часть) и большие (2 части).
- Вычислительные мощности предоставляются на определенные периоды времени, задавая время создания α и время удаления ω для каждой виртуальной машины.

Дополнительные ограничения

- Пользователь может потребовать отказоустойчивости предоставляемых ему вычислительных мощностей.
- Для этого каждой виртуальной машине ставится в соответствие группа размещения. Каждая группа размещения состоит из непересекающихся наборов виртуальных машин, называемых партициями.
- Виртуальные машины из одной группы размещения, но разных партиций, не могут быть размещены на одной стойке, даже если соблюдаются остальные ограничения.

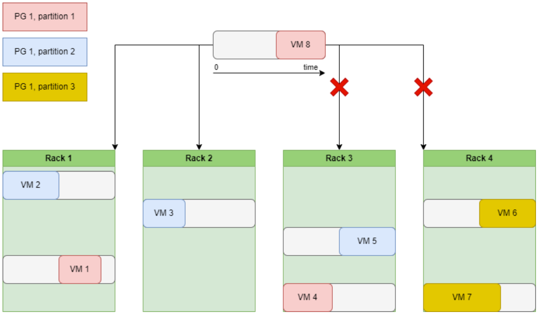


Рис. 2: Иллюстрация ограничений на размещение типа “партиции”

Дополнительные ограничения

- Или наоборот, пользователь может потребовать максимальной производительности. В таком случае виртуальные машины из одной группы должны быть упакованы на одной стойку (или даже сервер), если их времена работы пересекаются.

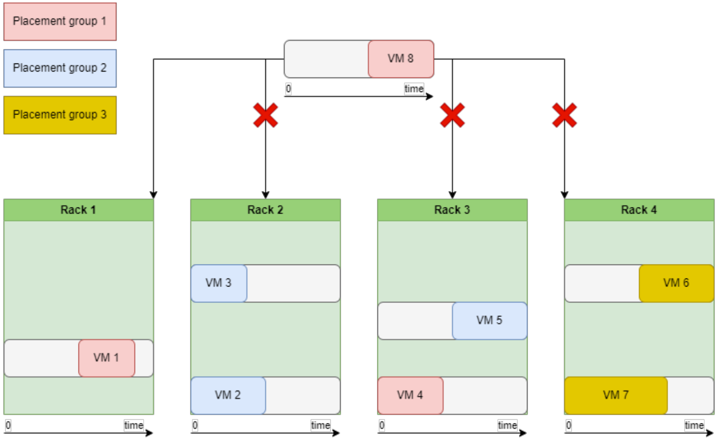


Рис. 3: Иллюстрация афинных ограничений на размещение

Постановка задачи

Задача (Базовая)

Задача заключается в минимизации количества активных стоек при заданном наборе запросов (виртуальных машин), ограничений на ресурсы и размещение.

Задача (С начальным размещением)

Задача заключается в минимизации количества дополнительных активных стоек при заданном наборе запросов, ограничений на ресурсы и размещение, начальном размещении ВМ.

- Задачи являются NP-трудными, если количество типов виртуальных машин не ограничено, поскольку это обобщение классической задачи упаковки в контейнеры.
- В работе[1] сказано, что увеличение плотности размещения виртуальных машин даже на 1% позволяет экономить до 100 миллионов долларов в год облачной платформе Microsoft Azure.

[1] Hadary O. et al. Protean:VM allocation service at scale //14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20). 2020. Ñ. 845-861.

Нижняя оценка для базовой задачи. Алгоритм генерации столбцов.

Для упрощения рассмотрим классическую задачу упаковки в контейнеры, с емкостью контейнера W и весами предметов w_i, $i \in I$ – типы предметов.

- Будем называть произвольный вектор $(a_i)_{i \in I}$ шаблоном упаковки, если он удовлетворяет ограничению: $\sum\limits_{i \in I} a_i \cdot w_i \leq W$.
- Рассмотрим множество всех возможных шаблонов J. Тогда задача упаковки в контейнеры может быть описана как:

$$
\begin{cases}
\sum\limits_{j \in J} x_j \to \min \\
\sum\limits_{j \in J} a_{ij} x_j \geq n_i, \ \ i \in I \\
x_j \in \mathbb{Z}_+, \ \ j \in J
\end{cases}
$$

Нижняя оценка для базовой задачи. Алгоритм генерации столбцов.

- Релаксируем переменные x_j, тогда решение задачи:

$$\begin{cases} \sum\limits_{j \in J} x_j \to \min \\ \sum\limits_{j \in J} a_{ij} x_j \geq n_i, \ \ i \in I \\ x_j \geq 0, \ \ j \in J, \end{cases}$$

будет нижней оценкой для исходной задачи, но $|J|$ экспоненциально большое. Поэтому рассмотрим подмножество $J' \subseteq J$, при котором данная задача будет разрешимой(можно получить с помощью жадных алгоритмов FF, BF, $\ldots$).

Нижняя оценка для базовой задачи. Алгоритм генерации столбцов.

- Пусть x^* – оптимальное решение для J', а $\lambda^* \geq 0$ – оптимальное решение двойственной задачи для J':

$$\begin{cases} \sum\limits_{i \in I} n_i \lambda_i \to max \\ \sum\limits_{i \in I} a_{ij} \lambda_i \leq 1, \ \ j \in J' \\ \lambda_i \geq 0, \ \ i \in I \end{cases}$$

- Если для всех $j \in J$ справедливо:

$$\sum_{i \in I} a_{ij} \lambda_i^* \leq 1,$$

то λ_j^* – оптимальное решение двойственной задачи для всего J, и нижняя оценка сохраняется.

Нижняя оценка для базовой задачи. Алгоритм генерации столбцов.

- Для того чтобы быстро проверить, что:

$$\sum_{i\in I} a_{ij}\lambda_i^* \leq 1, \forall j \in J, \qquad (*)$$

рассмотрим задачу генерации нового шаблона:

$$\begin{cases} \alpha = \sum\limits_{i\in I} \lambda_i^* y_i \to \max \\ \sum\limits_{i\in I} w_i y_i \leq W \\ y_i \in \mathbb{Z}_+, \end{cases}$$

Легко заметить, что если $\alpha^* \leq 1$, то выполняются все ограничения $(*)$, иначе мы получили новый шаблон, который добавляем в J' и повторяем итерацию.

Нижняя оценка для базовой задачи. Алгоритм генерации столбцов.

Алгоритм (Нижней оценки для базовой задачи)

1. *Выбрать множество самых нагруженных моментов времени* T.
2. *Вычислить нижние границы в фиксированный момент времени* $t \in T$ *с помощью алгоритма генерации столбцов:* $LB_t = \mathrm{CG}(VM_t)$
3. *Нижняя граница:* $LB = \max_{t \in T}\{LB_t\}$

Кочетов Ю. А., Ратушный А. В. Верхние и нижние оценки оптимума для задачи динамической упаковки в контейнеры //Труды Института математики и механики УрО РАН. – 2024. – Т. 30. – №. 1. – С. 109-127.

Применение алгоритма генерации столбцов к задаче с начальным размещением

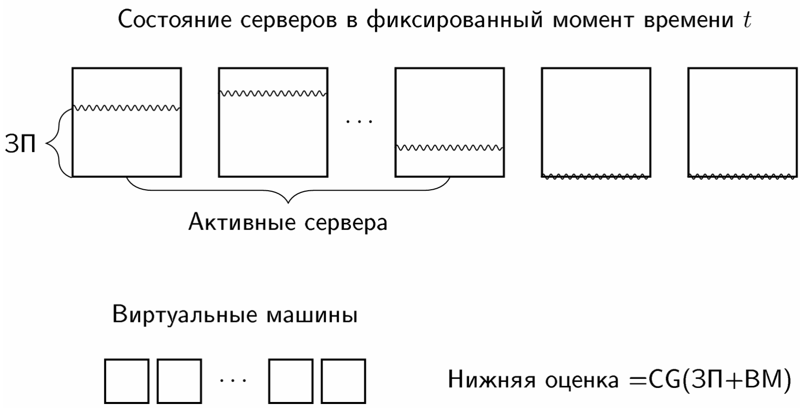
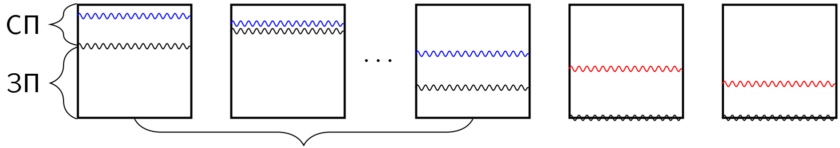


Рис. 4: Нижняя оценка с помощью модификации CG

Применение алгоритма генерации столбцов к задаче с начальным размещением

Рис. 5: Редукция типов

Оптимальная упаковка оставшихся ВМ в фиксированный момент времени t



Активные сервера

СП – свободные пространства

Виртуальные машины

$$\mathbf{N}(\text{ВМ}) \le \mathbf{N}(\text{СП})$$

$$\mathbf{N}(\underbrace{\text{ВМ} \cup \text{ВМ}}_{\text{ВМ}}) \le \mathbf{N}(\text{ВМ}) + \mathbf{N}(\text{ВМ})$$

$$\mathbf{N}(\text{ВМ}) \ge \mathbf{N}(\text{ВМ}) - \mathbf{N}(\text{СП})$$

Кол. доп. серверов $\ge$CG(ВМ)$-$UB(СП)

CG(ВМ)$\le \mathbf{N}$(ВМ) , $\mathbf{N}$(СП)$\le$UB(СП)

$\mathbf{N}(X)$ – мин. кол. **только пустых** серверов необходимых для упаковки X

Рис. 6: Нижняя оценка с помощью ECG

Нижняя оценка для задачи с начальным размещением

Дополнительно определяются преобразования над СП, сохраняющие инвариант "встраивания", то есть неравенство:

$$\mathbf{N}(\text{ВМ}) \leq \mathbf{N}(\text{СП})$$

Такими преобразованиями являются:

- **Горизонтальная нарезка**. Если на конкретном нума-узле никаким образом нельзя полностью заполнить один из ресурсов, то мы можем отрезать это лишнее свободное пространство.
- **Вертикальная нарезка**. Если на конкретный сервер нельзя упаковать ВМ большого размера, то мы можем рассматривать каждый нума-узел как отдельный предмет.

Кроме прочего, мы можем редуцировать само множество ВМ, убрав некоторые типы (малозатратные), тогда горизонтальная и вертикальная нарезки будут более эффективными, но может уменьшиться CG(ВМ').

$$LB \geq \text{CG(ВМ)} - \text{UB(СП')} \rightsquigarrow LB \geq \text{CG(ВМ')} - \text{UB(СП'')}$$

Нижняя оценка для задачи с начальным размещением

Алгоритм (Нижней оценки для задачи с начальным размещением, ECG)

1. *Выбрать множество самых нагруженных моментов времени* T. *Отсортировать множество типов ВМ* $I \subset \mathbb{N}$ *по возрастанию ресурсных требований. Введем обозначение:* $I_i = \{i, \dots, |I|\} \subset I,\ i \in I$. *А множество виртуальных машин данных типов в момент времени* t *обозначим как* M_{tI_i}
2. *Вычислить с помощью алгоритма генерации столбцов для всех* $t \in T$ *и* $i \in I$ *оценку снизу* LB_t^i :
$$LB_t^i = \mathrm{CG}(M_{tI_i}) \leq \mathbf{N}(M_{I_i})$$
3. *Для всех* $t \in T$ *и* $i \in I$ *вычислить оценку сверху* $UB_t^i : \mathbf{N}(E_{tI_i}) \leq UB_t^i$. *Здесь* E_{tI_i} *– нарезанное свободное пространство в момент времени* t, *учитывающее только типы* I_i.
4. *Нижняя граница:* $LB = \max\limits_{i \in I} \max\limits_{t \in T} \{LB_t^i - UB_t^i\}$.

Численные эксперименты

Открытый синтетический набор данных из библиотеки бенчмарков “Discrete Location Problems”[2] был использован для проведения экспериментов. Набор данных содержит в общей сложности 450 тестов, которые равномерно распределены в соответствии со следующими параметрами:

- Типы групп размещений:
  1. LPO, только большие группы с партициями;
  2. MP, большие и маленькие группы с партициями;
  3. DMP, группы с партициями и группы без ограничений на размещения.
- Общее количество запросов:
  1. Small, тесты, содержащие в среднем 14702 ВМ;
  2. Medium, тесты, содержащие в среднем 26545 ВМ;
  3. Large, тесты, содержащие в среднем 51301 ВМ.
- Количество используемых NUMA-узлов (2,4).

На основе данного набора данных были сгенерированы два набора тестовых примеров для задачи с начальным размещением, путем упаковки первых 30% и 70% виртуальных машин с помощью алгоритма First Fit.

[2] http://old.math.nsc.ru/AP/benchmarks/Temporal%20Bin%20Packing/binpack.html

Численные эксперименты

Таблица 1: Результаты работы алгоритмов на нижнюю границу (GAP)

	70%				30%			
Семейство	Trivial	CG	ECG	Both	Trivial	CG	ECG	Both
LPO	2.10%	1.31%	**1.25%**	1.15%	5.98%	**4.97%**	5.23%	4.80%
MP	4.16%	2.53%	**2.27%**	2.14%	5.85%	**5.02%**	5.05%	4.73%
DMP	3,24%	2,47%	**1,58%**	1,57%	4,23%	**3,67%**	3,71%	3,35%
small	4,04%	2,44%	**2,22%**	2,05%	5,77%	**4,45%**	4,75%	4,18%
medium	3,37%	2,13%	**1,67%**	1,59%	4,70%	**3,95%**	4,11%	3,73%
large	2,09%	1,74%	**1,22%**	1,21%	5,61%	5,25%	**5,13%**	4,98%
2	1,63%	0,89%	**0,75%**	0,63%	4,99%	4,15%	**3,87%**	3,64%
4	4,71%	3,32%	**2,65%**	2,61%	5,73%	**4,95%**	5,45%	4,95%
Итого	3,17%	2,10%	**1,70%**	1,62%	5,36%	**4,55%**	4,66%	4,29%
Среднее время	-	4.61 мин.	5.75 мин.	-	-	16.2 мин.	8.0 мин.	-
Максимальное время	-	65.7 мин.	21.2 мин.	-	-	154.0 мин.	33.3мин.	-

Верхяя граница получена с использованием алгоритма из статьи[3].

$$GAP = \frac{UB - LB}{UB} \cdot 100\%$$

[3] Turnaev A., Panin A. Stochastic greedy algorithms for a temporal bin packing problem with placement groups //International Conference on Mathematical Optimization Theory and Operations Research. – Cham : Springer Nature Switzerland, 2024. – С. 199-211.

Спасибо за внимание!