

Вертикальное фрагментирование в реляционных СУБД

Георгий Чернышев

Санкт-Петербургский Государственный Университет

chernishev@gmail.com
www.math.spbu.ru/user/chernishev/

30 апреля 2015 г.
Семинар ACM SIGMOD Chapter
BMK, МГУ

Overview

- 1 Введение
- 2 Постановка задачи
- 3 Методы решения
- 4 Эвристические методы
- 5 Стоимостные методы
- 6 Современное состояние

Вертикальное фрагментирование (1)

id	фамилия	комната	зарплата
1	Иванов	350	100
2	Петров	351	80
3	Сидоров	400	200

В первом приближении: “разрезание таблицы по вдоль атрибутов”:

id	фамилия	комната
1	Иванов	350
2	Петров	351
3	Сидоров	400

id	зарплата
1	100
2	80
3	200

Вертикальное фрагментирование (2): место в физическом дизайне

Плюсы: вертикальное фрагментирование позволяет уменьшить количество “ненужных” данных при считывании.

Принцип: “вместе запрашиваем — вместе храним”.

Сложности:

- 1 Не всегда влияние положительно: некоторым запросам придется считывать информацию из нескольких частей (и делать соединение);
- 2 При запросе на обновление производительность падает;
- 3 Если запросов несколько выгоду оценить еще сложнее;
- 4 Нормализация: можно потерять информацию, можно получить фантомы;

Что можно получить в случае успешного применения?

- **Внутризапросный параллелизм.** Успешное решение создаст необходимые условия для более широкого применения внутризапросного параллелизма, что позволит уменьшить время выполнения отдельного запроса.
- **Межзапросный параллелизм.** Ускорит выполнение набора запросов с помощью межзапросного параллелизма.
- **Увеличить мощность системы.** Организовать эффективное добавление новых узлов и устройств хранения в распределенную систему.
- **Понизить совокупную стоимость владения системы.** Упрощение системы в целом, упрощение или даже устранение задач администрирования, конфигурирования и настройки приведет к снижению затрат на персонал.

Вертикальное фрагментирование (3): как проводить

В отличие от горизонтального фрагментирования, не поддерживается в явном виде в языке [Jindal et al., 2012]:

```
create table BOOKSHELF_RANGE_PART
(Title  VARCHAR2 (100) primary key,
Publisher  VARCHAR2 (20),
CategoryName  VARCHAR2 (20),
Rating  VARCHAR2 (2),
constraint CATFK2 foreign key (CategoryName)
references CATEGORY (CategoryName)
)
partition by range (CategoryName)
(partition PART1 values less than ('B') tablespace PART1_TS,
partition PART2 values less than (MAXVALUE)
tablespace PART2_TS);
```

Это во-многом до сих пор искусство и отдается на откуп администратору БД.

Формальная постановка задачи настройки СУБД (1)

Поведение СУБД опишем как функцию
[Chaudhuri and Weikum, 2009]:

$$f : C \times W \rightarrow P,$$

где C обозначает конфигурацию системы:

- 1 аппаратная конфигурация;
- 2 программная конфигурация — настраиваемые параметры СУБД, вступающие в силу после загрузки;
- 3 конфигурацию физического уровня СУБД (использование фрагментирования, индексов, материализованных представлений и др.);

W — нагрузка, предполагаемый набор запросов и их характеристики: затрагиваемые атрибуты, типы запросов, интенсивность поступления в заданные периоды (рабочие или пиковые часы), распределение параметров этих запросов и многое другое.

Формальная постановка задачи настройки СУБД (2)

Поведение СУБД опишем как функцию
[Chaudhuri and Weikum, 2009]:

$$f : C \times W \rightarrow P,$$

Производительность P :

- 1 пропускная способность системы, измеряемая в количестве обработанных запросов в единицу времени (throughput);
- 2 время ответа на запрос (response time);
- 3 различные метрики измерения функциональной надежности (dependability): доступность (availability), надежность (reliability) и производительность в условиях ослабленных конфигураций (performability).

Формальная постановка задачи настройки СУБД (2)

Поведение СУБД опишем как функцию
[Chaudhuri and Weikum, 2009]:

$$f : C \times W \rightarrow P,$$

Задача: найти конфигурацию физического уровня (набор вертикальных фрагментов), обладающую заданной производительностью. Однако чаще всего требуемый уровень производительности заранее не известен, и необходимо найти максимально достижимый.

Итог: экстремальная задача, где в качестве максимизируемой функции выступает производительность P в зависимости от C и W , ограничения описываются набором уравнений и неравенств, а искомая конфигурация — набором параметров.

Уточнение рассматриваемой задачи

- 1 Конфигурация определяется как конфигурация физического уровня СУБД — набор вертикальных фрагментов. Все остальные компоненты конфигурации (аппаратная или программная часть) фиксированы и выступают в роли условий. Иногда часть конфигурации известна заранее;
- 2 Информация о нагрузке дополнена достаточно подробной информацией о данных, к которым обращаются запросы (количество и характеристики атрибутов таблиц и т.д.);
- 3 Пользовательские ограничения, не связанные напрямую с системой (могут считаться частью нагрузки). Например, передача данных между узлами в распределенной компьютерной сети осуществляется за плату, и для решения задачи выделен определенный бюджет.

О решении

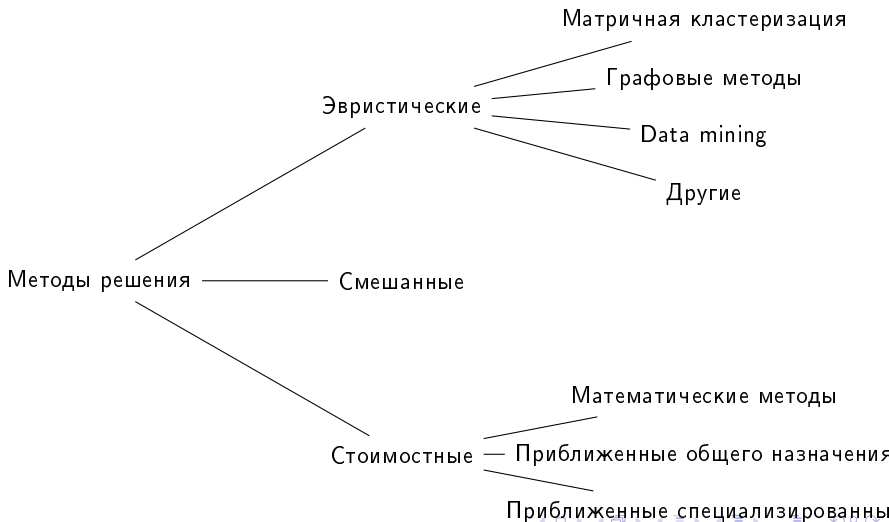
Для нахождения решения необходимы:

- 1 способ оценивать конфигурации → **стоимостная модель**, учитывающую характеристики аппаратной конфигурации, характеристики нагрузки и, возможно, другие пользовательские ограничения.
- 2 способ перебирать конфигурации
 - необходимо проверить экспоненциальное количество решений, например количество вариантов вертикального фрагментирования равно числу Белла $B_{n+1} = \sum_{k=0}^n \binom{N}{k} B_k$
 - это много:

$$B_1 = 1, \dots, B_5 = 15, \dots, B_{10} = 21147, \dots, B_{15} = 190899322$$

$$B_{20} = 5832742205057, \dots, B_{25} = 4638590332229999353$$

“Классификация” методов решения



Классификация методов решения (1)

- Эвристические — выбирается некоторая процедура создания решения, обосновывается почему она должна давать хорошее решение;
- Стоимостные — создается модель системы, формулируется некоторая стоимостная функция, минимизируется;
- Смешанные методы — комбинация вышеупомянутых.

Классификация методов решения (2): эвристические

- Близость атрибутов:
 - Методы матричной кластеризации — выделение фрагментов по матрице использования атрибутов, исторически первый способ решения задачи вертикального фргаментирования;
 - Графовые методы — поиск шаблонов и разрезание графа;
- Data mining — поиск совместно появляющихся атрибутов в запросах.

Методы матричной кластеризации (1)

Общая идея: составить на основании нагрузки матрицу близости атрибутов и кластеризовать ее — привести к блочно-диагональному виду путем перестановки столбцов и строк (Bond Energy Algorithm, BEA). Полученные блоки и будут фрагментами.

Самая ранняя работа данного типа: [Hoffer and Severance, 1975]

Attribute	Partition #1: (19,18,17,16,15,13,1,21)(8,7)(14)(6,9,20)(12,11,10)(5,3)(4)(2)(22)													
	Partition #2: (19,8,7,18,17,16,15,14,13,1,21,2)(6,9,20)(12,11,10)(5,3,4)(22)													
	19	8	7	18	17	16	15	14	6	13	12	11	10	5
19	1.00	0.76	0.76	0.62	0.65	0.65	0.62	0.60	0.44	0.50	0.25	0.25	0.25	0.3
8	0.76	1.00	0.92	0.77	0.73	0.72	0.79	0.90	0.93	0.67	0.42	0.42	0.42	0.5
7	0.76	0.92	1.00	0.75	0.72	0.72	0.77	0.84	0.79	0.65	0.43	0.48	0.48	0.5
18	0.62	0.77	0.75	1.00	0.74	0.70	0.66	0.62	0.50	0.45	0.27	0.27	0.27	0.3
17	0.65	0.73	0.72	0.74	1.00	0.92	0.59	0.56	0.40	0.45	0.24	0.24	0.24	0.3
16	0.65	0.72	0.72	0.70	0.92	1.00	0.63	0.58	0.40	0.46	0.24	0.24	0.24	0.3

Методы матричной кластеризации (2)

Сам алгоритм BEA был предложен в [McCormick et al., 1972].

Вместо BEA можно использовать и другие алгоритмы:
[Slagle et al., 1972] [Bhat and Haupt, 1976] и другие.

Был использован в качестве первого шага в знаменитой работе
[Navathe et al, 1984].

Методы матричной кластеризации (3): дальнейшие работы

- Серия работ Chun-Hung Cheng: [Cheng, 1994] [Cheng, 1995] [Cheng and Motwani, 2009] [Cheng et al., 2011].
- В работе [Gorla and Boe, 1990] предложили свою метрику близости атрибутов и свой алгоритм кластеризации. Удалось превзойти на отдельных наборах данных стоимостной метод [Hammer and Niamir, 1979]. Кроме того: [Gorla, 2007].
- Современность: попытка использовать/исследовать наработки прошлого [Jindal et al., 2013] [Jindal et al., 2012].

Графовые методы (1)

Общая идея:

- Преобразование матрицы используемых атрибутов (Attribute Usage Matrix) в матрицу близости (Attribute Affinity Matrix, AAM);
- Поиск специальных структур в полученном графе;
- (Опционально) выбор наилучших, “разрезание” графа.

Исторически первый метод графового типа (стоимостной) был предложен в [Eisner and Severance, 1976], но графовое направление породил [Navathe and Ra, 1989] (эвристический).

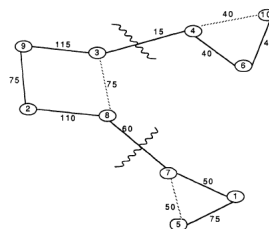
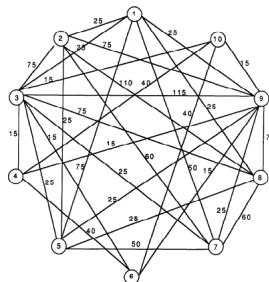
Пример графового алгоритма [Navathe and Ra, 1989]

Attribute usage matrix											Type	Number of accesses per time period
Attributes	1	2	3	4	5	6	7	8	9	10		
Transactions												
T1	1	0	0	0	1	0	1	0	0	0	R	Acc 1 = 25
T2	0	1	1	0	0	0	0	1	1	0	R	Acc 2 = 50
T3	0	0	0	1	0	1	0	0	0	1	R	Acc 3 = 25
T4	0	1	0	0	0	0	1	1	0	0	R	Acc 4 = 35
T5	1	1	1	0	1	0	1	1	1	0	U	Acc 5 = 25
T6	1	0	0	0	1	0	0	0	0	0	U	Acc 6 = 25
T7	0	0	1	0	0	1	0	0	1	0	U	Acc 7 = 25
T8	0	0	1	1	0	1	0	0	1	1	U	Acc 8 = 15

Fig.1 Attribute usage matrix

Attributes	1	2	3	4	5	6	7	8	9	10
1	75	25	25	0	75	0	50	25	25	0
2	25	110	75	0	25	0	60	110	75	0
3	25	75	115	15	25	15	25	75	115	15
4	0	0	15	40	0	40	0	0	15	40
5	75	25	25	0	75	0	50	25	25	0
6	0	0	15	40	0	40	0	0	15	40
7	50	60	25	0	50	0	85	60	25	0
8	25	110	75	0	25	0	60	110	75	0
9	25	75	115	15	25	15	25	75	115	15
10	0	0	15	40	0	40	0	0	15	40

Fig.2 Attribute affinity (AA) matrix



Графовые методы (2): дальнейшие работы

Позднее появились:

- Доказали NP-трудность формулировки Navathe, улучшение исходного алгоритма [Lin and Zhang, 1993], [Lin et al., 1993];
- Сводят задачу к TSP, решают с помощью генетического алгоритма, разрезают полученный путь [Cheng et al., 2002].
- Новая метрика близости и новый алгоритм фрагментирования были предложены в [Du et al., 2003];
- В работе [Son and Kim, 2001] авторы предложили использовать аппарат нечеткой логики для построения нечеткого графа;
- Упрощение алгоритма [Navathe and Ra, 1989] было предложено в [Marir et al., 2007];

Графовые методы (3): влияние

Идеи были восприняты сообществом, ученые с успехом применяли их для выбора других физических структур.

Использовались в:

- задаче горизонтального фрагментирования (кластеризуют предикаты) [Navathe et al., 1995], [Cheng et al., 2002];
- задаче размещения набора отношений в распределенной СУБД [Bellatreche and Benkrid, 2009];
- задаче фрагментирования в объектных СУБД [Fung03 et al., 2003], [Bellatreche et al., 2000], [Bellatreche et al., 1997].

И многих других.

Data mining и вертикальное фрагментирование (1)

Общая идея: применить поиск ассоциативных правил для атрибутов отношения. Появления искать в списке транзакций.

Ассоциативное правило: $X \longrightarrow Y$; X и Y множества атрибутов.
 Y правила есть две меры:

- confidence — вероятность появления Y если есть X (сила вывода);
- support — доля транзакций в которых есть и X и Y одновременно (частота встречи шаблонов в правиле).

Data mining и вертикальное фрагментирование (2)

Подходы:

- Работы [Gorla and Yan, 2008] [Gorla and Yan, 2009] используют Apriori алгоритм для нахождения large itemsets.
- Так же Apriori алгоритм в [Ramesh et al., 2014], нет фильтрации по confidence, нет стоимостного шага.
- В [Rodriguez and Li, 2011] решается проблема выбора порогового значения confidence и support. Предложен новый способ фрагментирования с использованием алгоритма из [Du et al., 2003].
- Сравнение алгоритмов FP-Max и Apriori [Bouakkaz et al., 2012]. Продолжили работы [Gorla and Yan, 2008] [Gorla and Yan, 2009].

Data mining и вертикальное фрагментирование (3)

Идея: ищем не frequent item sets, а closed item sets.

Closed itemset X :

- 1 Все элементы множества X встречаются в одном наборе транзакций.
- 2 Нет такого много множества X' что:
 - X' строгое надмножество X и
 - Любая транзакция содержащая X содержит X' .

Работы: [Guinepain and Gruenwald, 2006] [Guinepain and Gruenwald, 2008]
[Rahmani et al., 2011]

Data mining и вертикальное фрагментирование (4)

Индустриальное применение в MS SQL Server (вертикальное, горизонтальное, индексы): [Agrawal et al., 2004]
[Agrawal et al., 2005]

Для выбора материализованных представлений:
[Agrawal et al., 2000]

Стоимостные методы решения задачи вертикального фрагментирования

Есть явно заданная стоимостная функция и она в центре.

Как решать задачу:

- Математические методы — формулировка в виде задачи целочисленного программирования, применение некоторого математического метода для решения;
- Общие приближенные — метод восхождения к вершине, генетические алгоритмы, метод имитации отжига, методы муравьиной оптимизации, ... ;
- Специализированные приближенные — методы специально разработанные для задачи вертикального фрагментирования. Обычно специальный обход с отсечением.

Математические методы (1)

Математические методы — решение экстремальной задачи с помощью методов целочисленного программирования (integer programming).

- 1 Исторически первый способ решения задачи автоматической настройки СУБД, доминировали в 60-90 гг, особенно в задачах размещения.
- 2 Метод решения (линеаризация уравнений) составлял значительную часть предмета исследований. Ныне — редок, используются матпакеты [Papadomanolakis and Ailamaki, 2007], [Cornell and Yu, 1990].
- 3 Неэффективен. Пример [March and Rho, 1995]: задача размещения для трех отношений имеющих по три фрагмента, 4 узла, 6 запросов давала 1200 переменных и ограничений. Авторы утверждают что в то время такой размер мог решаться только на суперкомпьютерах.
- 4 Сложность линеаризации [Hammer and Niamir, 1979].

Пример постановки в виде экстремальной задачи

[March and Rho, 1995]

$$\begin{aligned} \text{Min Cost} = & \sum_j \sum_f (k,j) \sum_m (COM(k,j,m) + IO(k,j,m) \\ & + CPU(k,j,m)) + \sum_i STO(i) \end{aligned}$$

$\sum_i X_{it} \geq 1$ for all file fragments, $i = 1, 2, \dots$, number of fragments (all file fragments must be stored at one or more nodes)

$X_{it} = 1$ if file fragment i is stored at node t
0 otherwise

$Z_{kit} \leq X_{it}$ for all queries, $k = 1, 2, \dots$, number of queries for all file fragments, $i = 1, 2, \dots$, number of fragments for all nodes, $t = 1, 2, \dots$, number of nodes (a file fragment cannot be accessed from a node unless it is stored at that node)

$Z_{kit} = 1$ if query k uses file fragment i at node t
0 otherwise

$Y_{kmt} = 1$ if step m of query k is done at node t
0 otherwise

$\sum_t Y_{kmt} = 1$ for all queries, $k = 1, 2, \dots$, number of queries for all steps m , $m = 1, 2, \dots$, number of steps for query k (all query steps must be processed at some

Математические методы (2)

Формулировка задачи вертикального фрагментирования в виде экстремальной задачи довольно редка: [Hoffer, 1976] [March and Rho, 1995] [Amossen, 2010].

Еще более редко ее решение с помощью методов целочисленного программирования: [De et al., 1988] [Cornell and Yu, 1990] [Cornell and Yu, 1987].

Общие приближенные (1)

- Метод восхождения к вершине [Hammer and Niamir, 1979];
- Генетические алгоритмы [March and Rho, 1995], [Rho and March, 1994]; [Gorla and Quinn, 1991], [Song and Gorla, 2000], [Ng et al., 2003]; [Du et al., 2006];
- Муравьиная оптимизация [Cheng et al., 2011];
- Particle Swarm Optimization [Benmessahel and Touahria, 2009];
- Метод имитации отжига [Papadomanolakis and Ailamaki, 2003];
- Графовые методы: A* [Huang and Van, 1995] и Ford-Fulkerson [Eisner and Severance, 1976];
- ...

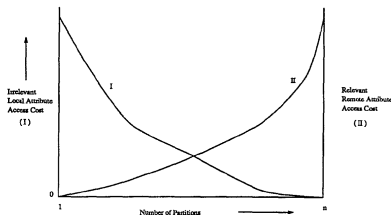
Специализированные приближенные (1): атрибутивное и транзакционное фрагментирование

Специализированные стоимостные методы — методы разработанные именно для решения задачи вертикального фрагментирования в стоимостной постановке. Учитывают специфику постановки и за счет этого обычно выигрывают у общих.

- атрибутивное: [Apers, 1988] (первая работа такого типа?) и практически все современные подходы;
- транзакционное: [Chu and leong, 1993] и [Pai-Cheng, 1992]

Специализированные приближенные (2): Partition Evaluator

Метод для сравнения эвристических алгоритмов основанных на близости: [Muthuraj et al., 1993], [Chakravarthy et al., 1994].



Проблема: различные критерии “хорошести” решения в данных работах. Результаты разные даже для одной ААМ.

Решение: функция Partition Evaluator, принимает на вход АУМ и другую информацию о транзакциях.

Компоненты:

- стоимость доступа к ненужным атрибутам (локально)
- стоимость доступа к нужным атрибутам (удаленно)

Специализированные приближенные (3): бинарное, n-арное и k-way постановка

Еще одна классификация фрагментирования:

- Бинарное: результат — два фрагмента [Navathe et al, 1984];
- N-арное: результат — сколько-то фрагментов, но максимизирует функцию.

Иногда нужно ровно k фрагментов — k -way постановка: [Son and Kim, 2004] [Rodriguez and Li, 2011]

Смешанные методы решения (1)

Методы комбинирующие эвристические и стоимостные. Были очень популярны в 80-90-е, когда вычислительные ресурсы были ограничены.

Пример [Navathe et al, 1984], в нем:

- применение процедуры, упрощение задачи;
- выбор финального решения с использованием стоимостной функции.

Attribute	5	1	7	2	8	3	9	10	4	6
5	75	75	50	25	25	25	25	0	0	0
1	75	75	50	25	25	25	25	0	0	0
7	50	50	85	60	60	25	25	0	0	0
2	25	25	60	110	110	75	75	0	0	0
8	25	25	60	110	110	75	75	0	0	0
3	25	25	25	75	75	115	115	15	15	15
9	25	25	25	75	75	115	115	15	15	15
<hr/>										
10	0	0	0	0	0	15	15	40	40	40
4	0	0	0	0	0	15	15	40	40	40
6	0	0	0	0	0	15	15	40	40	40

(c)

Смешанные методы решения (2)

Применяются и ныне, примеры:

- Фильтрация групп колонок (column group restriction) [Agrawal et al., 2004];
- Вычисляется “порядок” на атрибутах, потом “разрезается” [Jindal et al., 2012];
- Кластеризация матрицы, затем стоимостная оценка [Gorla, 2007];
- В работах [Gorla and Yan, 2008] [Gorla and Yan, 2009] [Li and Gruenwald, 2012] [Guinepain and Gruenwald, 2008] на последнем шаге применяется стоимостная оценка кандидатов;
- ...

Проблемы старых работ

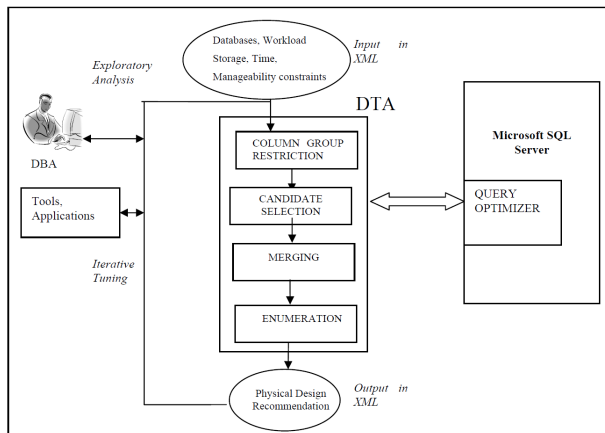
- Нет единого бенчмарка, были известные примеры на которых проверяли: матрица из [Navathe et al, 1984], примеры из [Cornell and Yu, 1990] но они неполноценны и многие работы многие работы их модифицировали;
- Нет полного порядка в работах: не все сравнивались со всеми, не все даже просто реализовывались, иногда разные метрики оценивания;
- Разделенность сообщества — про некоторые работы просто не узнать, разная терминология;
- ...

Современное состояние (1)

- Отказ от разработки своих моделей с нуля, использование “what-if” механизмов;
- Исчезновение эвристических методов, максимум — второстепенная роль;
- Вал работ от корпораций по выбору структур физического уровня для промышленных СУБД: Autoadmin Project (10 работ) [Chaudhuri and Narasayya, 2007], DB 2 (5 работ) [Rao et al., 2002], MS SQL Server и PWD (5 работ) [Nehme and Bruno, 2011], [Agrawal et al., 2004], Oracle (5 работ, не совсем выбор), PostgreSQL (более 10 работ) и т.д.
- В них вертикальное фрагментирование слабо представлено.

Современное состояние (2): общая схема и пример

Пример современной работы с вертикальным фрагментированием [Agrawal et al., 2004]:



Современные направления

Где современные работы по вертикальному фрагментированию?

- Динамизация задачи вертикального фрагментирования
[Li and Gruenwald, 2013] [Jindal et al., 2012]
[Malik et al., 2009] [Rodriguez et al., 2012];
- Современные архитектуры: фрагментирование в памяти,
колоночные СУБД.

Динамизация

Новая постановка задачи, всё что было раньше — статическая постановка.

Задача: уметь адаптировать на лету конфигурацию физического уровня (вертикальное фрагментирование) так, чтобы при изменении характера запросов, производительность не падала.

Одно из свойств самоуправляемых СУБД (self-managed database) [Chaudhuri and Weikum, 2009].

Новые архитектуры СУБД

Колоночные СУБД (column-stores).

Системы с фрагментированием в оперативной памяти.

Колоночные СУБД (1)

СУБД хранящая каждый атрибут отдельно от других, в памяти или на диске. Основные архитектурные отличия от классических СУБД с хранением по строкам (row-stores):

- Вместе с новым способом хранения нужен новый способ обработки запросов: восстановление записи (ранняя/поздняя материализация), новые планы запросов (даже в простых планах появляются циклы), новая алгебра.
- Новая реализация операторов, новые колоночные операторы.
- Сжатие данных: стало выгоднее — с 2-3 раз до 10.

Колоночные СУБД (2)

Исследовательские прототипы [Abadi et al., 2013]:

- C-Store [Lamb et al., 2013], ориентированная на работу с диском: сжатие, ранняя/поздняя материализация, специальные операторы.
- MonetDB, ориентированная на работу в оперативной памяти: упор на эффективное использования (уменьшение количества промахов в кеш-памяти, SSE инструкции и т.д.), адаптивное индексирование.
- VectorWise, ориентированная на работу в оперативной памяти: блочная обработка, кооперативные сканы.

Масса коммерческих разработок:

- Стартапы 2005–2010;
- Позже “старые” компании выпустили свои продукты или интегрировали функционал.

Колоночные СУБД (3)

Плюсы (OLAP нагрузки) и минусы (OLTP нагрузки):

- (+) Хорошо работают на read-only нагрузках;
- (+) Хорошо работают когда выбирается малая доля атрибутов;
- (-) Обновления работают хуже в колоночной системе;
- (-) Архитектура становится невыгодна когда выбирается много атрибутов;
- (-) Числа с плавающей точкой плохо сжимаются;

Хотелось бы совместить возможность делать OLAP и OLTP на одной системе.

Решение: надо сделать гибрид, возможно работающий в оперативной памяти.

Гибридные системы и системы с фрагментированием в оперативной памяти (1): ранние работы

Делают тоже самое но для страниц в оперативной памяти!

- PAX
[Ailamaki2001 et al., 2001]
министраницы;
- Группировка на страницах
[Hankins and Patel, 2003];
- Динамическая
реорганизация
[Shao et al., 2004].

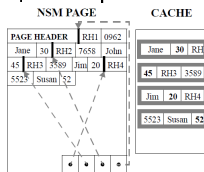


FIGURE 1: The cache behavior of NSM

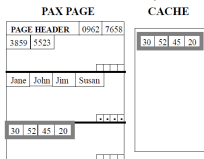


FIGURE 3: The cache behavior of PAX

Гибридные системы и системы с фрагментированием в оперативной памяти (2): настоящее время

- HYRISE [Grund et al., 2010] — фрагментирование в памяти и исполнение запросов;
- SAP HANA [Rosch et al., 2012] — фрагментирование в памяти, стоимостная модель, гибридное хранение (есть row-storage и есть column-storage);
- H2O [Grund et al., 2014] — фрагментирование и компиляция кода запроса под схему.

Ссылки I



Navathe S. B., Ra M. Vertical partitioning for database design: a graphical algorithm // Proceedings of the 1989 ACM SIGMOD international conference on Management of data. SIGMOD'89. New York, NY, USA : ACM, 1989. P. 440–450. URL: <http://doi.acm.org/10.1145/67544.66966>.



An enhanced grouping algorithm for vertical partitioning problem in DDBs / F. Marir, Y. Najjar, M.Y. AlFaress, H.I. Abdalla // Computer and information sciences, 2007. iscis 2007. 22nd international symposium on. 2007. nov. P. 1–6.



Son J. H., Kim M.-H. α -partitioning algorithm: Vertical partitioning based on the fuzzy graph // Proceedings of the 12th International Conference on Database and Expert Systems Applications. DEXA '01. London, UK, UK : Springer-Verlag, 2001. P. 537–546. URL: <http://dl.acm.org/citation.cfm?id=648314.755837>.

Ссылки II



Du J., Barker K., Alhadj R. Attraction — a global affinity measure for database vertical partitioning // ICWI. IADIS, 2003. P. 538–548.



Lin X., Zhang Y. A new graphical method of vertical partitioning in database design. // Australian Database Conference. 1993. P. 131–144. URL: <http://dblp.uni-trier.de/db/conf/adc/adc93.html#LinZ93>.



Lin X., Orlowska M., Zhang Y. A graph based cluster approach for vertical partitioning in database design // Data & Knowledge Engineering. 1993. Vol. 11, no. 2. P. 151–169. URL: <http://www.sciencedirect.com/science/article/pii/0169023X93900038>.



Navathe S. B., Karlapalem K., Ra M. A mixed fragmentation methodology for initial distributed database design // Journal of Computer and Software Engineering. 1995. Vol. 3.

Ссылки III



Bellatreche L., Karlapalem K., Simonet A. Algorithms and support for horizontal class partitioning in object-oriented databases // Distributed and Parallel Databases. 2000. Vol. 8. P. 155–179. 10.1023/A:1008745624048. URL: <http://dx.doi.org/10.1023/A:1008745624048>.



Bellatreche L., Karlapalem K., Simonet A. Horizontal class partitioning in object-oriented databases // Database and Expert Systems Applications / Ed. by Abdelkader Hameurlain, AMin Tjoa. Springer Berlin Heidelberg, 1997. Vol. 1308 of Lecture Notes in Computer Science. P. 58–67. URL: <http://dx.doi.org/10.1007/BFb0022018>.



Fung C.-W., Karlapalem K., Li Q. Cost-driven vertical class partitioning for methods in object oriented databases // The VLDB Journal. 2003. oct. Vol. 12, no. 3. P. 187–210. URL: <http://dx.doi.org/10.1007/s00778-002-0084-7>.

Ссылки IV



Bellatreche L., Benkrid S. A joint design approach of partitioning and allocation in parallel data warehouses // Data Warehousing and Knowledge Discovery / Ed. by Torben Pedersen, Mukesh Mohania, A Tjoa. Springer Berlin / Heidelberg, 2009. Vol. 5691 of Lecture Notes in Computer Science. P. 99–110. 10.1007/978-3-642-03730-6 9. URL: http://dx.doi.org/10.1007/978-3-642-03730-6_9.



Mark J. Eisner and Dennis G. Severance. 1976. Mathematical Techniques for Efficient Record Segmentation in Large Shared Databases. J. ACM 23, 4 (October 1976), 619–635. DOI=10.1145/321978.321982 <http://doi.acm.org/10.1145/321978.321982>



Cheng C.-H., Lee W.-K., Wong K.-F. A genetic algorithm-based clustering approach for database partitioning // Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on. 2002. Vol. 32, no. 3. P. 215–230.

Ссылки V



Hoffer J. A., Severance D. G. The use of cluster analysis in physical data base design // Proceedings of the 1st International Conference on Very Large Data Bases. VLDB '75. New York, NY, USA : ACM, 1975. P. 69–86. URL: <http://doi.acm.org/10.1145/1282480.1282486>.



McCormick, W.T., Schweitzer, P.J., and White, W.W. Problem decomposition and data reorganization by a clustering technique. Oper. Res. 20, 5 (Sept. 1972), 993–1009







J. R. Slagle, C. L. Chang, and S. R. Heller, A clustering and data-reorganization algorithm. IEEE Trans. Syst., Man, Cybern., vol. SMC-5, pp. 125–128, 1975.



Bhat, M. V., and Haupt, A., 1976, An efficient clustering algorithm. IEEE Transactions on Systems, Man, and Cybernetics, SMC-6, 61–64.

Ссылки VI

-  Vertical partitioning algorithms for database design / Shamkant Navathe, Stefano Ceri, Gio Wiederhold, Jingle Dou // ACM Trans. Database Syst. 1984. Vol. 9. P. 680–710. URL: <http://doi.acm.org/10.1145/1994.2209>.
-  Cheng C. Algorithms for vertical partitioning in database physical design // Omega. 1994. Vol. 22, no. 3. P. 291–303. URL: <http://www.sciencedirect.com/science/article/pii/0305048394900426>.
-  Cheng C.-H. A branch and bound clustering algorithm // Systems, Man and Cybernetics, IEEE Transactions on. 1995. Vol. 25, no. 5. P. 895–898.
-  Cheng C.-H., Motwani J. An examination of cluster identification-based algorithms for vertical partitions // Int. J. Bus. Inf. Syst. 2009. Vol. 4, no. 6. P. 622–638. URL: <http://dx.doi.org/10.1504/IJBIS.2009.026695>.

Ссылки VII



Cheng C.-H., Wong K.-F., Woo K.-H. An improved branch-and-bound clustering approach for data partitioning // International Transactions in Operational Research. 2011. Vol. 18, no. 2. P. 231–255. URL: <http://dx.doi.org/10.1111/j.1475-3995.2010.00781.x>.



Gorla N., Boe W. J. Database operating efficiency in fragmented databases in mainframe, mini, and micro system environments // Data & Knowledge Engineering. 1990. Vol. 5, no. 1. P. 1–19. URL: <http://www.sciencedirect.com/science/article/pii/0169023X9090030H>.



Gorla N. A methodology for vertically partitioning in a multi-relation database environment // Journal of Computer Science & Technology. 2007. jun. Vol. 7, no. 3. P. 217–227.

Ссылки VIII



A comparison of knives for bread slicing / Alekh Jindal, Endre Palatinus, Vladimir Pavlov, Jens Dittrich // Proc. VLDB Endow. 2013. apr. Vol. 6, no. 6. P. 361–372. URL:
<http://dl.acm.org/citation.cfm?id=2536336.2536338>.



Jindal, A., Dittrich, J.: Relax and let the database do the partitioning online. In: Castellanos, M., Dayal, U., Lehner, W. (eds.) Enabling Real-Time Business Intelligence, Lecture Notes in Business Information Processing, vol. 126, pp. 65–80. Springer Berlin Heidelberg (2012)



Guinepain S., Gruenwald L. Automatic database clustering using data mining // Proceedings of the 17th International Conference on Database and Expert Systems Applications. DEXA'06. Washington, DC, USA : IEEE Computer Society, 2006. P. 124–128. URL:
<http://dx.doi.org/10.1109/DEXA.2006.32>.

Ссылки IX



Guinepain S., Gruenwald L. Using cluster computing to support automatic and dynamic database clustering // Cluster Computing, 2008 IEEE International Conference on. 2008. P. 394–401.



Gorla N., Yan B. P. W. Vertical fragmentation in databases using datamining technique // IJDWM. 2008. Vol. 4, no. 3. P. 35–53.



Gorla N., Yan B. P. W. Vertical fragmentation in databases using datamining technique // Database Technologies: Concepts, Methodologies, Tools, and Applications / Ed. by John Erickson. IGI Global, 2009. P. 2543–2563.



Employing frequent pattern mining for finding correlations between tables in relational databases / A Rahmani, M. Nagi, Mohammad Rifaie et al. // Information Technology: New Generations (ITNG), 2011 Eighth International Conference on. 2011. April. P. 547–552.

Ссылки X



An apriori-based vertical fragmentation technique for heterogeneous distributed database transactions / Ramesh Dharavath, Vikas Kumar, Chiranjeev Kumar, Amit Kumar // Intelligent Computing, Networking, and Informatics / Ed. by Durga Prasad Mohapatra, Srikanta Patnaik. Springer India, 2014. Vol. 243 of Advances in Intelligent Systems and Computing. P. 687–695. URL: http://dx.doi.org/10.1007/978-81-322-1665-0_69.



Rodriguez L., Li X. A support-based vertical partitioning method for database design // Electrical Engineering Computing Science and Automatic Control (CCE), 2011 8th International Conference on. 2011. oct. P. 1–6.



Bouakkaz M., Ouinten Y., Ziani B. Vertical fragmentation of data warehouses using the FP-Max algorithm // Innovations in Information Technology (IIT), 2012 International Conference on. 2012. march. P. 273–276.

Ссылки XI



Agrawal S., Narasayya V., Yang B. Integrating vertical and horizontal partitioning into automated physical database design // Proceedings of the 2004 ACM SIGMOD international conference on Management of data. SIGMOD '04. New York, NY, USA : ACM, 2004. P. 359–370. URL: <http://doi.acm.org/10.1145/1007568.1007609>.



Database tuning advisor for microsoft sql server 2005: Demo / Sanjay Agrawal, Surajit Chaudhuri, Lubor Kollar et al. // Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. SIGMOD '05. New York, NY, USA : ACM, 2005. P. 930–932. URL: <http://doi.acm.org/10.1145/1066157.1066292>.



Agrawal S., Chaudhuri S., Narasayya V. R. Automated selection of materialized views and indexes in sql databases // Proceedings of the 26th International Conference on Very Large Data Bases. VLDB '00. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2000. P. 496–505. URL: <http://dl.acm.org/citation.cfm?id=645926.671701>.

Ссылки XII



Hammer M., Niamir B. A heuristic approach to attribute partitioning // Proceedings of the 1979 ACM SIGMOD international conference on Management of data. SIGMOD '79. New York, NY, USA : ACM, 1979. P. 93–101. URL: <http://doi.acm.org/10.1145/582095.582110>.



March S. T., Rho S. Allocating data and operations to nodes in distributed database design // IEEE Trans. on Knowl. and Data Eng. 1995. Vol. 7, no. 2. P. 305–317. URL: <http://dx.doi.org/10.1109/69.382299>.



Papadomanolakis S., Ailamaki A. An integer linear programming approach to database design // Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop. ICDEW '07. Washington, DC, USA : IEEE Computer Society, 2007. P. 442–449. URL: <http://dx.doi.org/10.1109/ICDEW.2007.4401027>



Hoffer, J.A. An integer programming formulation of computer database design problems. Inf. Sci. 11 (July 1976), 29–48.

Ссылки XIII



Amossen R, Vertical Partitioning of Relational OLTP Databases using Integer Programming, Data Engineering Workshops (ICDEW) of IEEE 5th International Conference on Self Managing Database Systems (SMDB), 2010



Cornell D., Yu P. An effective approach to vertical partitioning for physical design of relational databases // Software Engineering, IEEE Transactions on. 1990. Vol. 16, no. 2. P. 248–258.



Cornell D. W., Yu P. S. A vertical partitioning algorithm for relational databases // Proceedings of the Third International Conference on Data Engineering. Washington, DC, USA : IEEE Computer Society, 1987. P. 30–35. URL: <http://dl.acm.org/citation.cfm?id=645472.653385>.

Ссылки XIV



De P., Park J. S., Pirkul H. An integrated model of record segmentation and access path selection for databases // Information Systems. 1988. Vol. 13, no. 1. P. 13–30. URL: <http://www.sciencedirect.com/science/article/pii/0306437988900245>.



Chun-Hung Cheng, Angappa Gunasekaran, Kwan-Ho Woo. A bi-tour ant colony optimisation framework for vertical partitions. Int. J. of Industrial and Systems Engineering, 2011 Vol.7, No.3, pp.341–356



Efstratios Papadomanolakis and Anastassia Ailamaki. AutoPart: Automating Schema Design for Large Scientific Databases Using Data Partitioning. Technical report CMU-CS-03-159 (20 pages / 24 references version). July 2003. School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213.



Bilal Benmessahel, Mohamed Touahria. Applying A Discrete Particle Swarm Optimization Algorithm to Database Vertical Partition. CIIA, 2009

Ссылки XV



Rho S., March S. A nested genetic algorithm for distributed database design // System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on. Vol. 3. 1994. Jan. P. 33–42.



Gorla N., Quinn W. Combined optimal tuple ordering and attribute partitioning in storage schema design // Inf. Softw. Technol. 1991. jun. Vol. 33, no. 5. P. 335–339. URL:
[http://dx.doi.org/10.1016/0950-5849\(91\)90101-G](http://dx.doi.org/10.1016/0950-5849(91)90101-G).



Song S.-K., Gorla N. A genetic algorithm for vertical fragmentation and access path selection // The Computer Journal. 2000. Vol. 43, no. 1. P. 81–93. <http://comjnl.oxfordjournals.org/content/43/1/81.full.pdf+html>.

Ссылки XVI



Applying genetic algorithms in database partitioning / Vincent Ng, Dik Man Law, Narasimhaiah Gorla, Chi Kong Chan // Proceedings of the 2003 ACM symposium on Applied computing. SAC '03. New York, NY, USA : ACM, 2003. P. 544–549. URL: <http://doi.acm.org/10.1145/952532.952639>.



Du J., Alhadj R., Barker K. Genetic algorithms based approach to database vertical partition // J. Intell. Inf. Syst. 2006. Vol. 26. P. 167–183. URL: <http://dl.acm.org/citation.cfm?id=1139987.1139989>.



Huang Y.-F., Van C.-H. Vertical partitioning in database design // Information Sciences. 1995. Vol. 86, no. 1-3. P. 19–35. URL: <http://www.sciencedirect.com/science/article/pii/002002559500018K>.



Apers P. M. G. Data allocation in distributed database systems // ACM Trans. Database Syst. 1988. Vol. 13. P. 263–304. URL: <http://doi.acm.org/10.1145/44498.45063>.

Ссылки XVII



Chu W., leong I. A transaction-based approach to vertical partitioning for relational database systems // Software Engineering, IEEE Transactions on. 1993. Vol. 19, no. 8. P. 804–812.



Pai-Cheng C. A transaction-oriented approach to attribute partitioning // Information Systems. 1992. Vol. 17, no. 4. P. 329–342. URL: <http://www.sciencedirect.com/science/article/pii/030643799290022F>.



A formal approach to the vertical partitioning problem in distributed database design / J. Muthuraj, S. Chakravarthy, R. Varadarajan, S. B. Navathe // Proceedings of the second international conference on Parallel and distributed information systems. PDIS '93. Los Alamitos, CA, USA : IEEE Computer Society Press, 1993. P. 26–35. URL: <http://dl.acm.org/citation.cfm?id=382019.382410>.

Ссылки XVIII



An objective function for vertically partitioning relations in distributed databases and its analysis / Sharma Chakravarthy, Jaykumar Muthuraj, Ravi Varadarajan, Shamkant B. Navathe // Distributed and Parallel Databases. 1994. Vol. 2. P. 183–207. 10.1007/BF01267326. URL: <http://dx.doi.org/10.1007/BF01267326>.



Son J. H., Kim M.-H. An adaptable vertical partitioning method in distributed systems // Journal of Systems and Software. 2004. Vol. 73, no. 3. P. 551–561.



Rodriguez L., Li X. A vertical partitioning algorithm for distributed multimedia databases // Proceedings of the 22nd international conference on Database and expert systems applications - Volume Part II. DEXA'11. Berlin, Heidelberg : Springer-Verlag, 2011. P. 544–558. URL: <http://dl.acm.org/citation.cfm?id=2033546.2033607>.

Ссылки XIX



Chaudhuri S., Narasayya V. Self-tuning database systems: a decade of progress // Proceedings of the 33rd international conference on Very large data bases. VLDB '07. VLDB Endowment, 2007. P. 3–14. URL: <http://dl.acm.org/citation.cfm?id=1325851.1325856>.



Automating physical database design in a parallel database / Jun Rao, Chun Zhang, Nimrod Megiddo, Guy Lohman // Proceedings of the 2002 ACM SIGMOD international conference on Management of data. SIGMOD '02. New York, NY, USA : ACM, 2002. P. 558–569. URL: <http://doi.acm.org/10.1145/564691.564757>.



Nehme R., Bruno N. Automated partitioning design in parallel database systems // Proceedings of the 2011 international conference on Management of data. SIGMOD '11. New York, NY, USA : ACM, 2011. P. 1137–1148. URL: <http://doi.acm.org/10.1145/1989323.1989444>.

Ссылки XX



Li L., Gruenwald L. Autonomous database partitioning using data mining on single computers and cluster computers // Proceedings of the 16th International Database Engineering; Applications Symposium. IDEAS '12. New York, NY, USA : ACM, 2012. P. 32–41. URL: <http://doi.acm.org/10.1145/2351476.2351481>.



Li, L., Gruenwald, L.: Self-managing online partitioner for databases (SMOPD): A vertical database partitioning system with a fully automatic online approach. In: Proceedings of the 17th International Database Engineering Applications Symposium. pp. 168–173. IDEAS '13 (2013).



Malik, T., Wang, X., Dash, D., Chaudhary, A., Ailamaki, A., Burns, R.: Adaptive physical design for curated archives. In: Proceedings of the 21st International Conference on Scientific and Statistical Database Management. pp. 148–166. SSDBM 2009 (2009)

Ссылки XXI



Dymond: An active system for dynamic vertical partitioning of multimedia databases. Lisbeth Rodriguez, Xiaou Li, Jair Cervantes, Farid Garcia-Lamont. Proceedings of the 16th International Database Engineering; Applications Symposium. IDEAS '12. New York, NY, USA : ACM, 2012. P. 71–80. URL: <http://doi.acm.org/10.1145/2351476.2351485>.



Chaudhuri S., Weikum G. Self-management technology in databases // Encyclopedia of Database Systems / Ed. by Ling Liu, M.Tamer Ozsu. Springer US, 2009. P. 2550–2555. URL: http://dx.doi.org/10.1007/978-0-387-39940-9_334.



Abadi, D., Boncz, P., Harizopoulos, S.: The Design and Implementation of Modern Column-Oriented Database Systems (2013)

Ссылки XXII



Lamb, A., Fuller, M., Varadarajan, R., Tran, N., Vandiver, B., Doshi, L., Bear, C.: The vertica analytic database: C-store 7 years later. Proc. VLDB Endow. 5(12), 1790–1801 (Aug 2012)



A.Ailamaki, D.DeWitt, M.Hill, M.Skounakis. Weaving Relations for Cache Performance. Proceedings of VLDB 2001.



Richard A. Hankins and Jignesh M. Patel. 2003. Data morphing: an adaptive, cache-conscious storage technique. In Proceedings of the 29th international conference on Very large data bases - Volume 29 (VLDB '03), Johann Christoph Freytag, Peter C. Lockemann, Serge Abiteboul, Michael J. Carey, Patricia G. Selinger, and Andreas Heuer (Eds.), Vol. 29. VLDB Endowment 417–428.

Ссылки XXIII



Minglong Shao, Jiri Schindler, Steven W. Schlosser, Anastassia Ailamaki, and Gregory R. Ganger. 2004. Clotho: decoupling memory page layout from storage organization. In Proceedings of the Thirtieth international conference on Very large data bases - Volume 30 (VLDB '04), Mario A. Nascimento, M. Tamer Özsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and K. Bernhard Schiefer (Eds.), Vol. 30. VLDB Endowment 696–707.



Lightstone S. Physical database design for relational databases // Encyclopedia of Database Systems / Ed. by Ling Liu, M.Tamer 'OZSU. Springer US, 2009. P. 2108–2114. URL: http://dx.doi.org/10.1007/978-0-387-39940-9_644.



Alagiannis, I., Idreos, S., Ailamaki, A.: H2o: A hands-free adaptive store. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. pp. 1103–1114. SIGMOD '14 (2014)

Ссылки XXIV



Martin Grund, Jens Krüger, Hasso Plattner, Alexander Zeier, Philippe Cudre-Mauroux, and Samuel Madden. 2010. HYRISE: a main memory hybrid storage engine. Proc. VLDB Endow. 4, 2 (November 2010), 105–116. DOI=10.14778/1921071.1921077
<http://dx.doi.org/10.14778/1921071.1921077>



Rosch, P., Dannecker, L., Farber, F., Hackenbroich, G.: A storage advisor for hybrid-store databases. Proc. VLDB Endow. 5(12), 1748–1758 (Aug 2012)