

Дисциплина «Проектирование баз данных»



Маркова Ирина Васильевна,
начальник управления
информатизации
markova@mit.ru



Классификация методов

Методы выполнения операторов физического плана различают:

- a) по базовой стратегии:
 - сканирование;
 - сортировка;
 - хеширование;
 - индексирование.

- b) по трудоемкости:
 - однопроходные;
 - циклические:
 - двухпроходные;
 - многопроходные;

- c) по схеме обмена между операторами физического плана:
 - итератор (не предполагает фиксации на диске);
 - материализация (с промежуточным хранением).



Допущения

- a) мера эффективности оператора - это количество операции ввода/вывода;
- b) при сопоставлении алгоритмов руководствуемся тем, что данные-аргументы любого оператора изначально располагаются на диске, но результат его выполнения сохраняется в оперативной памяти;
- c) если оператор возвращает итоговый результат, которые нужно сохранить на диске, то стоимость этой операции будет зависеть только от объема данных результата, а не от того, как они получены.



Сканирование

Сканирование – одна из наиболее употребительных функций, связанная с чтением полного содержимого некоторого отношения R .

Не имеет непосредственного отношения к реализации операций реляционной алгебры, но используется при выполнении следующих операций:

- объединение (union);
- соединение (join);
- и др.

Существует два различных способа для получения кортежей отношения R :

1. табличное сканирование, при котором, в случае компактного размещения кортежей в определенной группе блоков вторичной памяти (адреса блоков известны), система может загружать блоки последовательно и получать либо все значения либо диапазон значений;
2. индексное сканирование, при котором для некоторого атрибута отношения R имеется индекс и он используется для нахождения всех значений или диапазона.



Сканирование с сортировкой (sort-scan)

Данный метод относится к операторам физического плана и используется в следующих случаях:

- для предложения ORDER BY и др.;
- для всех многопроходных алгоритмов.

Существуют следующие способы реализации:

1. индексное сканирование,
если существует отношение R , есть атрибут A , а на него есть индекс $index(A)$ либо отношение R хранится в индексированном последовательном файле;
2. табличное или индексное сканирование с последующим упорядочением в оперативной памяти,
если R – мало и полностью помещается в оперативной памяти;
3. сортировка двухфазным многокомпонентным слиянием,
если R – большое и не помещается полностью в буферах оперативной памяти, при этом итоговый результат не сохраняется на диске, но имеется возможность последовательного получения отдельных блоков отсортированного отношения R по мере возникновения потребности использования кортежей из этих блоков.



Оценка затрат на ввод/вывод для операций сканирования

№ п/п	Способ реализации	Оценка
1.	Табличное сканирование	
	R – компактно сгруппировано	$\approx B_R$
	R – не компактно	$\approx k_R$
2.	Сканирование с сортировкой	
	R – малое отношение в ОП	$\approx B_R$
	R – малое, не компактное	$\approx k_R$
3.	Сортировка двухфазным многокомпонентным слиянием	
	R – компактно сгруппировано	$\approx 3B_R$
	R – не компактно	$\approx T + 2B_R$
4.	Индексное сканирование ($B_{index(A)} \leq B_R$)	
	R – компактно сгруппировано	$\approx B_R$
	R – не компактно	$\approx k_R$



Сортировка во вторичной памяти. Сортировка слиянием (merge-scan)

Идея: организуется файл в виде постепенно увеличивающихся серий последовательностей записей r_1, \dots, r_k , где $(1 \leq i \leq k)$, k – длина серии.

Алгоритм:

- 1) n записей делятся на два файла f_1 и f_2 (можно считать, что любой файл состоит из серий длиной 1, $k = 1$);
- 2) объединяют серии $k = 1$ и распределяют их по файлам g_1 и g_2 , организованным в виде серий длиной $k = 2$;
- 3) f_1 и f_2 – пусты, объединяем g_1 и g_2 в f_1 и f_2 , организованные в виде серий $k = 4$ и т.д.



Затраты на сортировку слиянием

После выполнения i проходов получаются два файла, состоящие из серий длиной $k = 2^i$. Если $2^i \geq n$, то один из этих файлов будет пустым, а другой будет содержать единственную серию $k = n$, т.е. будет отсортирован (при этом достаточно $\lceil \log n \rceil + 1$ проходов). Каждый проход i требует чтения и записи двух файлов, длина каждого из которых $\frac{n}{2}$.

Общее число блоков, прочитанных или записанных во время одного из проходов составит

$$\frac{2n}{b},$$

где b – число записей в блоке.

Количество операций чтения и записи блоков для всего процесса сортировки:

$$\frac{O((n \log n))}{b}.$$



Пример сортировки слиянием

Пусть есть список из 23 чисел, поделенный на два файла:

а) исходные файлы:

f_1 : 28 3 93 10 54 65 30 90 10 69 8 22

f_2 : 31 5 96 40 85 9 39 13 8 77 10

1. серии $k = 2$ ($k = 2^1$):

g_1 :	28	31	93	96	54	85	30	39	8	10	8	10
g_2 :	3	5	10	40	9	65	13	90	69	77	22	



Пример сортировки слиянием (серии $k=4$ и $k=8$)

2. серии $k=4$ ($k=2^2$):

f_1 :	3	5	28	31		9	54	65	85		8	10	69	77
f_2 :	10	40	93	96		13	30	39	90		8	10	22	

3. серии $k=8$ ($k=2^3$):

g_1 :	3	5	10	28	31	40	93	96		8	8	10	10	22	69	77
g_2 :	9	13	30	39	54	65	85	90								



Пример сортировки слиянием (серии $k=16$ и $k=32$)

4. серии $k=16$ ($k=2^4$):

f_1 : 3 5 9 10 13 28 30 31 39 40 54 65 85 90 93 96

f_2 : 8 8 10 10 22 69 77

5. серии $k=32$ ($k=2^5$):

g_1 : 3 5 8 8 9 10 10 10 13 22 28 30 31 39 40 54 65 69 77 85 90 93 96



Многоканальное слияние

Пусть в системе имеется $2m$ (m – вход, m – выход) дисководов, каждый из которых имеет собственный канал доступа к основной памяти.

Тогда можно разместить на m дисководах m файлов (f_1, f_2, \dots, f_m) , организованных в виде серий длиной k .

Таким образом, можно прочитать m серий (по одной из каждого файла) и объединить их в одну серию длиной mk . Эта серия помещается в один из m выходных файлов

(g_1, g_2, \dots, g_m) , каждый из которых по очереди получает ту или иную серию.

Процесс слияния в оперативной памяти (ОП) можно выполнить за $O(\log m)$ шагов на 1 запись.

Если имеется n записей, а длина серии после каждого прохода умножается на m , то после i проходов $k = m^i$. Если $m^i > n$, т.е. после $i = \log_m n$ проходов весь список отсортирован.



Многоканальное слияние (оценка и выводы)

Коэффициент экономии:

$$\log_2 m \quad (\text{т.к. } \log_m n = \frac{\log_2 n}{\log_2 m}).$$

Таким образом, если имеется m входных дисководов и m выходных, то можно обрабатывать в m раз быстрее, чем в предыдущем методе (1 дисковод на вход и 1 – на выход). Кроме того, можно обрабатывать данные в 2 раза быстрее, чем при наличии 1 диска на вход и выход (т.е. когда данные хранятся на одном диске). Бесконечное увеличение m не приводит к ускорению обработки, т.к. при достаточно больших значениях m время, необходимое для слияния в основной памяти (которое растёт фактически пропорционально $\log_2 m$) существенно превосходит время, требующееся для считывания и записи данных. Т.е. растёт общее время обработки данных, т.к. «узким местом» становятся вычисления в оперативной памяти.



Многофазная сортировка

Для экономии на количестве дисководов предложен алгоритм сортировки слиянием, который выполняется лишь с помощью $m + 1$ файла. При этом осуществляется ряд проходов с объединением серий из m файлов в более длинные серии в $(m + 1)$ -м файле:

1. В течение одного прохода, когда серии от каждого из m файлов объединяются в серии $(m + 1)$ -го файла, нет необходимости использовать все серии от каждого из m входных файлов.

Когда какой-либо из файлов становится выходным, он заполняется сериями определённой длины, причём количество их равно минимальному количеству серий, находящихся в объединённых файлах.

2. В результате каждого прохода получаются файлы разной длины. Длина всех серий на текущем проходе представляет собой сумму длин серий, созданных за предшествующие m проходов (если вылов полноено менее m проходов, можно считать, что гипотетические проходы, выполненные до первого, создавали серии длины 1).



Многофазная сортировка (пример)

Пусть $m = 2$, мы начинаем с двух файлов f_1 и f_2 , организованных в виде серий длиной 1 ($k = 1$). Записи из f_1 и f_2 объединяются, образуя серии длиной $k = 2$ в третьем файле f_3 ($n_1 = 13, n_2 = 21$).

Подходы	f_1	f_2	f_3
начало	13(1)	21(1)	пусто
1	пусто	8(1)	13(2)
2	8(3)	пусто	5(2)
3	3(3)	5(5)	пусто
4	пусто	2(5)	3(8)
5	2(13)	пусто	1(8)
6	1(13)	1(21)	пусто
7	пусто	пусто	1(34)л

Объединяем оставшиеся серии $k = 1$ из файла f_2 с таким же количеством серий $k = 2$ из файла f_3 . В результате получаются серии длины $k = 3$, которые помещаются в пустой файл f_1 и т.д.



Многофазная сортировка (условие сходимости)

Последовательность длин серий 1,1,2,3,5,8,13,21 представляет собой последовательность Фибоначчи. Она удовлетворяет рекуррентному соотношению:

$$F_i = F_{i-1} + F_{i-2},$$

для $i \geq 2$ с начальным значением $F_0 = F_1 = 1$.

Отношение последовательных чисел Фибоначчи F_{i-1}/F_i приближается к «золотому

соотношению» $(\sqrt{5} + 1)/2 \approx 1,618\dots$ по мере увеличения i .

Оказывается, чтобы сохранить нужный ход процесса сортировки (пока не будет отсортирован весь список), начальные количества записей в f_1 и f_2 должны представлять собой два последовательных числа Фибоначчи.

В нашем примере – $n = 4$ (число Фибоначчи F_8), распределяется $n_1 = 13(F_6)$ и $n_2 = 21(F_7)$, их соотношение $\approx 1,615$, что близко к 1,618.



Оценка временных затрат при сортировке

Дано:

 R

$$k = 10000000$$

$$L_{\text{зан}} = 160 \text{ байт}$$

$$V_b = 16384 \text{ байта} - \text{размер блок}$$

$$k_b = 100 - \text{количество записей в блоке}$$

$$B_R = 100000$$

дисковый привод – 1

оперативная память – 100 Мб (объём буферов в оперативной памяти)

количество блоков, которые можно разместить в буферах оперативной памяти –

$$100 \cdot 2^{20} / 2^{14} = 6400$$



Оценка временных затрат при многофазной сортировке

Многофазная сортировка:

1 фаза: 16 раз $((6400 \cdot 15 + 4000))$ заполняем оперативную память считываемыми блоками, сортируем, сохраняем отсортированные списки на диске: $100000 \cdot 2 = 200000$ операций дискового ввода (блоки размещаются на диске в случайном порядке).

Каждая операция считывания и записи – по 11 мксек: $200000 \cdot 11 \cdot 10^{-3} = 2200 \text{сек.} \approx 37 \text{мин.}$ (около 2 мин. на каждый подсписок).

2 фаза: $100000 \cdot 2 \cdot \log_2 16 = 100000 \cdot 8 = 800000$. $T = 800000 \cdot 11 \cdot 10^{-3} = 8800 \text{сек.} \approx 147 \text{мин.}$



Оценка временных затрат при многофазной сортировке ($m + 1$)

Многофазная сортировка ($m + 1$):

37 мин. (100000 чтений во входной буфер и т.к. каждая запись однократно помещается в выходной буфер, то количество операций записи – 100000). Общее время сортировки 74 мин.

Чем длиннее серии содержит файл перед началом внешней сортировки, тем меньше потребуется слияний и тем быстрее закончится сортировка.

До начала применения любого из методов внешней сортировки, основанных на применении серий, начальный файл частями считывается в оперативную память, к каждой части применяется один из наиболее эффективных алгоритмов внутренней сортировки (например, быстрая сортировка) и отсортированные части, образующие серии, записываются в новый файл.