ЛЕКЦИЯ 20 - ВВЕДЕНИЕ В МЕТОДЫ ВНЕШНЕЙ СОРТИРОВКИ

Область приложения методов внешней сортировки покрывает случаи, когда число сортируемых записей превышает объем быстродействующего оперативного запоминающего устройства. Внешняя сортировка в корне отлична от внутренней (хотя в обоих случаях необходимо расположить записи данного файла в неубывающем порядке), и объясняется это тем, что время доступа к файлам на внешних носителях нас жесточайшим образом лимитирует. Структура данных должна быть такой, чтобы сравнительно медленные периферийные запоминающие устройства (ленты, диски и т. д.) могли справиться с потребностями алгоритма сортировки. Поэтому большинство изученных до сих пор методов внутренней сортировки (вставка, обмен, выбор) фактически бесполезно для внешней сортировки; необходимо рассмотреть всю проблему заново.

Предположим, например, что предназначенный для сортировки файл состоит из 5000 записей R_1, R_2, ..., R_{5000}. Длиной по 20 слов (хотя ключи K_i не обязательно такой длины). Как быть, если во внутренней памяти данной машины помещается одновременно только 1000 из этих записей?

Сразу напрашивается такое решение: начать с сортировки каждого из пяти подфайлов R_1, R_2, ..., R_{1000}; R_{1001}, R_{1002}, ..., R_{2000}; ...; R_{4001}, R_{4002}, ..., R_{5000} по отдельности и затем слить полученные подфайлы. К счастью, слияние оперирует только очень простыми структурами данных, именно линейными списками, пройти которые можно последовательным образом, как стеки или очереди. Поэтому для слияния годятся самые дешевые внешние запоминающие устройства.

Только что описанный процесс — внутренняя сортировка с последующим "внешним слиянием" — весьма популярен, и методики внешней сортировки сводятся в основном к вариациям на эту тему.

Возрастающие последовательности записей, получаемые на начальной фазе внутренней сортировки, в литературе о сортировке часто называются *цепочками;* эта терминология довольно широко распространена, но, к сожалению, она противоречит еще более распространенному использованию термина "цепочка" в других разделах вычислительной науки, где он означает *произвольную* последовательность символов. При изучении перестановок уже было дано вполне подходящее название для упорядоченных сегментов файла, которые мы договорились называть возрастающими отрезками или просто *отрезками.* В соответствии с этим будем использовать слово "отрезки" для обозначения упорядоченных частей файла. Таким образом, использование понятий "цепочки отрезков" и "отрезки цепочек" не приведет ни к каким недоразумениям.

Рассмотрим процесс внешней сортировки, использующей в качестве вспомогательной памяти *магнитные ленты* (простой аналогией магнитной ленты может служить последовательный файл). Вероятно, простейшим и наиболее привлекательным способом слияния с применением лент служит сбалансированное двухпутевое слияние, в основе которого лежит идея, использовавшаяся ранее в алгоритмах *естественного двухпутевого слияния* **N**, *простого двухпутевого слияния* **S** и *слияния списков* **L**. В процессе слияния нам потребуются четыре "рабочие ленты". На протяжении первой фазы возрастающие отрезки, получаемые при внутренней сортировке, помещаются поочередно на ленты 1 и 2 до тех пор, пока не исчерпаются исходные данные. Затем ленты 1 и 2 перематываем к началу и сливаем отрезки, находящиеся на этих лентах, получая новые отрезки, вдвое длиннее исходных. Эти новые отрезки записываются по мере их формирования попеременно на ленты 3 и 4. (Если на ленте 1 на один отрезок больше, чем на ленте 2, то предполагается, что лента 2 содержит

дополнительный "фиктивный" отрезок длины 0.) Затем все ленты перематываются к началу и содержимое лент 3 и 4 сливается в удвоенные по длине отрезки, записываемые поочередно на ленты 1 и 2. Процесс продолжается (при этом длина отрезков каждый раз удваивается) до тех пор, пока не останется один отрезок (а именно весь упорядоченный файл). Если после внутренней сортировки было получено S отрезков, причем $2^{k-1} < S \le 2^k$, то процедура сбалансированного двухпутевого слияния произведет ровно $k = \lceil \log_2 S \rceil$ проходов по всем данным.

Например, в рассмотренной выше ситуации, когда требуется упорядочить 5000 записей, а объем внутренней памяти составляет 1000 записей, мы имеем S = 5. Начальная распределительная фаза процесса сортировки поместит пять отрезков на ленты следующим образом:

Лента 1 $R_1,\ R_2, ..., R_{1000}; R_{2001},\ R_{2002}, ..., R_{3000}; R_{4001},\ R_{4002}, ..., R_{5000}$

Лента 2 $R_{1001},\ R_{1002}, ..., R_{2000}; R_{3001},\ R_{3002}, ..., R_{4000}$ (1)

Лента 3 (пустая)

Лента 4 (пустая)

После первого прохода слияния на лентах 3 и 4 получатся более длинные отрезки, чем на лентах 1 и 2:

Лента 3 $R_{1001},\ R_{1002}, ..., R_{2000}; R_{4001},\ R_{4002}..., R_{5000}$ (2)

Лента 4 $R_{2001},\ R_{2002}, ..., R_{4000}$

В конец ленты 2 неявно добавляется фиктивный отрезок, так что отрезок $R_{4001},\ R_{4002}, ..., R_{5000}$ просто копируется на ленту 3. После перемотки всех лент к началу следующий проход по данным приведет к такому результату:

Лента 1 $R_1,\ R_2, ..., R_{4000}$ (3)

Лента 2 $R_{4001},\ R_{4002}, ..., R_{5000}$

(Отрезок $R_{4001},\ R_{4002}, ..., R_{5000}$ снова копируется, но если бы мы начали с 8000 записей, то в этот момент лента 2 содержала бы $R_{4001},\ R_{4002}, ..., R_{8000}$). Наконец, после еще одной перемотки на ленте 3 окажется отрезок $R_1,\ R_2, ..., R_{4000}$, и сортировка закончится.

Сбалансированное слияние легко обобщается на случай T лент для любого $T \ge 3$. Выберем произвольное число P, такое, что $1 \le P < T$, и разделим T лент на два "банка": P лент в левом банке и $T - P$ лент в правом банке. Распределим исходные отрезки как можно равномернее по P лентам левого "банка", затем выполним P - путевое слияние слева направо, после этого — ($T - P$) - путевое слияние справа налево и т. д., пока сортировка не завершится. Обычно значение P лучше всего выбирать равным $\left\lceil \frac{T}{2} \right\rceil$.

При T = 4, P = 2 имеем частный случай — сбалансированное двухпутевое слияние. Вновь рассмотрим предыдущий пример, используя большее количество лент; положим T = 6 и P = 3. Начальное распределение теперь будет таким:

Лента 1 $R_1,\ R_2, ..., R_{1000}; R_{3001},\ R_{3002}, ..., R_{4000}$

Лента 2 $R_{1001},\ R_{1002}, ..., R_{2000}; R_{4001},\ R_{4002}, ..., R_{5000}$ (4)

Лента 3 $R_{2001},\ R_{2002}, ..., R_{3000}$

Первый проход слияния приведет к

Лента 4 $R_1,\ R_2, ..., R_{3000}$

Лента 5 R_{3001}, R_{3002}, ..., R_{5000} (5)

Лента 6 (пустая)

(Предполагается, что на ленте 3 помещен фиктивный отрезок.) На втором проходе слияния работа завершается и отрезки R_1, R_2, ..., R_{5000} помещаются на ленту 1. Этот частный случай $T = 6$ эквивалентен $T = 5$, так как шестая лента используется лишь при $S > 7$.

Трехпутевое слияние затрачивает фактически несколько больше времени центрального процессора, чем двухпутевое, но оно обычно пренебрежимо мало по сравнению с временем, необходимым для чтения, записи и перемотки ленты; мы довольно хорошо оценим время выполнения сортировки, если примем во внимание только суммарную величину перемещений лент. В предыдущем примере ((4) и (5)) требуются только два прохода по данным в сравнении с тремя проходами при $T = 4$. Таким образом, слияние при $T = 6$ займет около двух третей времени по отношению к предыдущему случаю.

Сбалансированное слияние кажется очень простым и естественным. Но если приглядеться внимательнее, то сразу видно, что это не наилучший способ в разобранных выше частных случаях. Вместо того чтобы переходить от (1) к (2) и перематывать все ленты, нам следовало остановить первое слияние, когда ленты 3 и 4 содержали соответственно R_1, R_2, ..., R_{2000} и R_{2001}, R_{2002}, ..., R_{4000}, а лента 1 была готова к считыванию R_{4001}, R_{4002}, ..., R_{5000}. Затем ленты 2, 3, 4 могли быть перемотаны к началу, и сортировка завершилась бы трехпутевым слиянием на ленту 2. Общее число записей, прочитанных с ленты в ходе этой процедуры, составило бы 4000+5000=9000 против 5000+5000+5000=15000 в сбалансированной схеме.

Имея пять отрезков и четыре ленты, можно поступить еще лучше, распределив отрезки следующим образом:

Лента 1 R_1, R_2, ..., R_{1000}; R_{3001}, R_{3002}, ..., R_{4000}

Лента 2 R_{1001}, R_{1002}, ..., R_{2000}; R_{4001}, R_{4002}, ..., R_{5000}

Лента 3 R_{2001}, R_{2002}, ..., R_{3000}

Лента 4 (пустая)

Теперь, выполнив трехпутевое слияние на ленту 4, затем перемотку лент 3 и 4 с последующим трехпутевым слиянием на ленту 3, можно было бы завершить сортировку, прочитав всего 3000+5000=8000 записей.

Наконец, если бы мы имели шесть лент, то могли бы, конечно, записать исходные отрезки на ленты 1—5 и закончить сортировку за один проход, выполнив пятипутевое слияние на ленту 6. Рассмотрение этих случаев показывает, что простое сбалансированное слияние не является наилучшим, и было бы интересно поискать улучшенные схемы слияния.

Такова основная идея подавляющего большинства методов внешней сортировки. Начальная фаза произвольного метода внешней сортировки предназначена для порождения начальных отрезков. Для этих целей используются методы внутренней сортировки. Наиболее часто применяются алгоритмы выбора с замещением, которые используют порядок, присутствующий в большинстве данных, чтобы породить длинные отрезки, значительно превосходящие емкость внутренней памяти.

Среди важнейших схем слияния, применяемых в методах внешней сортировки следует отметить: многофазное слияние; каскадное слияние; осциллирующая сортировка.

Существуют, однако, схемы внешней сортировки, не основанные на процедуре слияния. Одним из представителей таких методов является внешняя поразрядная сортировка. Этот метод иногда называют распределяющей сортировкой, поколонной сортировкой, карманной

сортировкой, цифровой сортировкой, сортировкой разделением и т. д. Он, как оказывается, по существу, *противоположен* слиянию!

Предположим, например, что в нашем распоряжении имеются четыре ленты, а ключей может быть только восемь: 0, 1, 2, 3, 4, 5, 6, 7. Если исходные данные находятся на ленте T1, то начнем с переписи всех четных ключей на T3 и всех нечетных на T4:

	T1	T2	T3	T4
Дано	{0, 1, 2, 3, 4, 5, 6, 7}	---	---	---
Проход 1			{0, 2, 4, 6}	{1, 3, 5, 7}

Теперь перематываем ленты и читаем T3, а затем T4, помещая {0, 1, 4, 5} на T1 и {2, 3, 6, 7} на T2:

Проход 2.	{0, 4} {1, 5}	{2, 6} {3, 7}	---	----

(Строка {0, 4}{1, 5} обозначает файл, содержащий записи только с ключами 0 и 4, за которыми следуют записи только с ключами 1 и 5. Заметим, что T1 теперь содержит те ключи, средний двоичный разряд которых содержит 0.) После еще одной перемотки и распределения ключей 0, 1, 2, 3 на T3 и ключей 4, 5, 6, 7 на T4 мы имеем

Проход 3	{0}{1}{2}{3}	{4}{5}{6}{7}

Теперь копирование T4 в конец T3 завершает работу. В общем случае для ключей в диапазоне от 0 до $2^k - 1$ можно отсортировать файл аналогичным образом, использовав k проходов, за которыми следует фаза окончательной "сборки", копирующая примерно половину данных с одной ленты на другую. Имея шесть лент, мы можем аналогичным образом использовать представления по основанию 3 для сортировки ключей от 0 до $3^k - 1$ за k проходов.

Оказывается верным следующее интересное обобщающее утверждение: *каждой схеме слияния соответствует схема распределения и каждой схеме распределения соответствует схема слияния.* По некотором размышлении это становится понятным. Рассмотрим сортировку слиянием, делающую все наоборот, т. е. "разливающую" окончательный выводной файл в подфайлы, которые "разливаются" в другие, и т. д.; наконец, мы разольем файл в S отрезков. Подобная схема возможна с лентами тогда и только тогда, когда возможна соответствующая схема распределения для поразрядной сортировки S ключей. Эта двойственность слияния и распределения почти точна; она не выполняется только в одном отношении: вводная лента должна сохраняться в разные моменты времени.

Таково "академическое" представление о процедурах внешней сортировки. Пока мы не вступим в единоборство с грубой действительностью настоящих лент и реальных сортируемых данных, для нас лучше, изучая характеристики этих схем, иметь весьма наивное представление о ленточной сортировке. Например, можно с легкой душой полагать (как мы делали до сих пор), что первоначальные исходные записи появляются волшебным образом в течение первой распределительной фазы; на самом деле они вероятно, будут

занимать одну из наших лент и, быть может, даже целиком заполнят несколько бобин, так как лента не бесконечна! Практические алгоритмы внешней сортировки учитывают эти и многие другие реальные ограничения, которые в свою очередь сильно влияют на выбор схемы слияния или распределения.