



---


# 查询操作实现




数据磁盘存放

查询操作实现

查询优化

- 
- 
- 选择  $\sigma_c(R)$
  - 连接  $(R \bowtie S)$

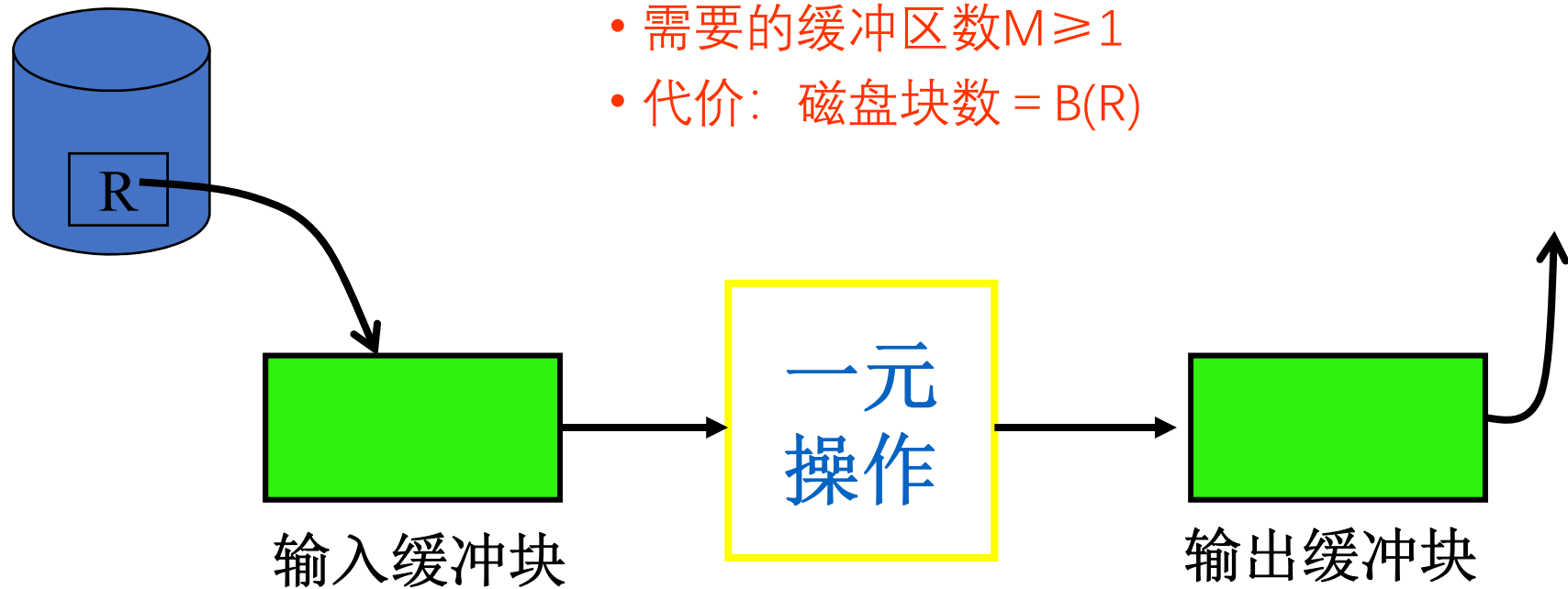
- 
- 
- 数据位于磁盘;
  - 用读写磁盘块数作为衡量每个操作代价的标准;
  - 操作结果放在内存中(不计I/O)
    - 直接送应用程序或打印;
    - (操作符之间) 流水线操作;

## 影响操作代价的参数

---

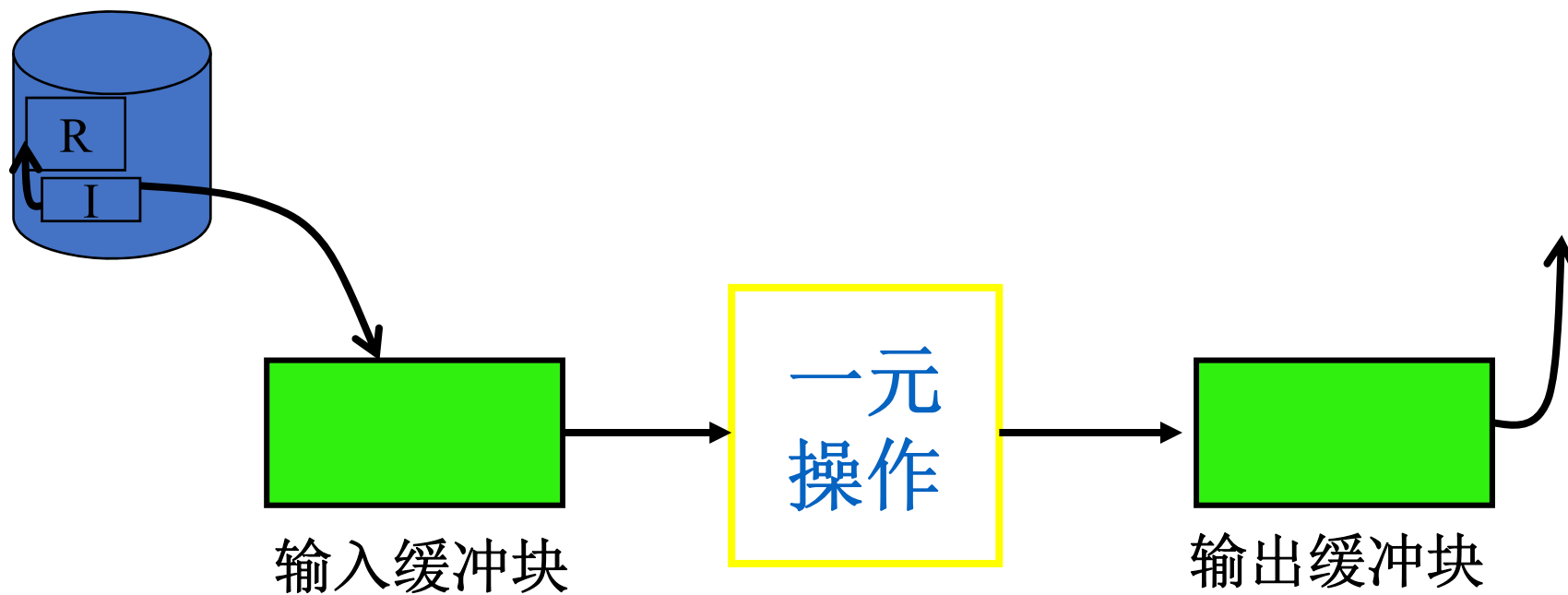
- 内存缓冲区数( $M$ )
- 关系 $R$ 的存储块数 $B(R)$
- 关系 $R$ 的元组数目 $T(R)$

## 关系的选择运算 $\sigma_C(R)$



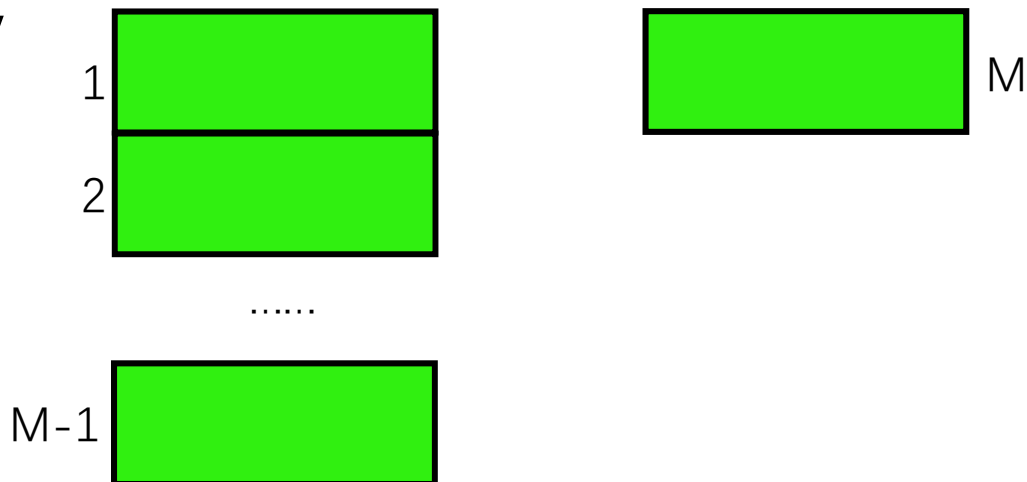
$$\sigma_{a='xyz'}(R)$$

- 假设只命中一个
  - 需要的缓冲区数  $M \geq 1$
  - 代价：磁盘块数 = 2
- 假设命中全部
  - 需要的缓冲区数  $M \geq 1$
  - 代价：  $B(R) + B(I)$



## 自然联接 $R(X,Y), S(Y,Z)$ : 对Y, R和S无序

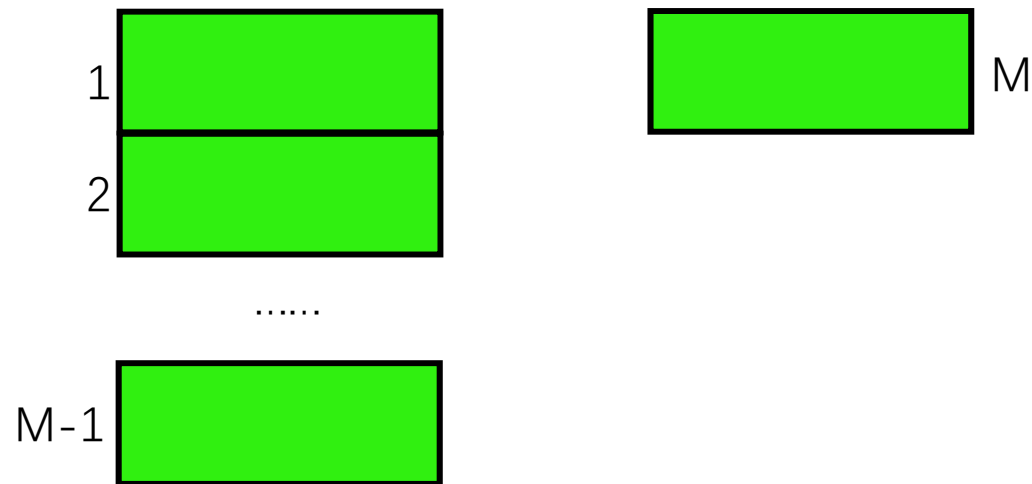
- 假设S是较小的关系, 并且  $B(S) \leq M-1$
- 将较小关系S读入内存的M-1个缓冲区中, 并在主存中用连接属性Y构造一个查找结构;
- 将R的每块读到第M个缓冲区, 对每块均进行如下的处理:
  - 对R块中的每个元组t
    - 比较t是否在S中是否有匹配的元组,
      - 如无匹配 -- 忽略t,
      - 如有匹配, 则将内存中所有匹配元组与t联接输出;

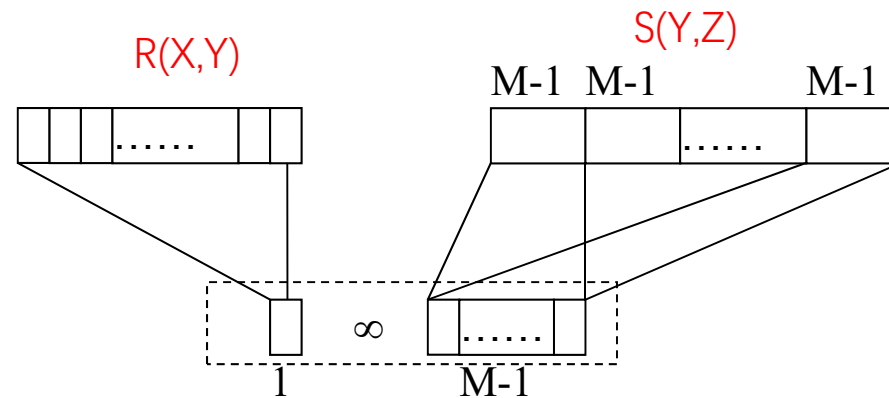




## 自然联接 $R(X,Y), S(Y,Z)$ :

- 假设S是较小的关系, 并且  $B(S) > M-1$  (小关系内存也放不下)





- M-1个块：用于读S
- 1个块：用于读R
- 在M-1个块的S元组的连接属性上构建查找结构
- 对内存中R块里的每一个元组，通过查找结构与S中的元组连接

# 块嵌套循环连接

- 算法

FOR each chunk of  $M-1$  blocks of  $S$  do

读这些 $S$ 的块到主存;

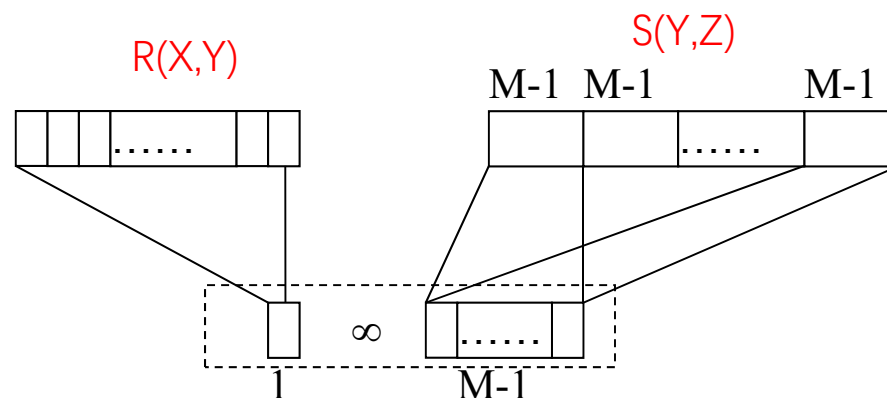
以连接属性为这些  $S$ 在内存中的元组建立查找结构

FOR 每一个块  $b \in R$  DO

读入该块( $b$ )

FOR 每个元组  $t \in b$ 块

找出当前内存中所有能与 $t$  匹配的 $S$ 元组,  
并逐个联接输出;



### ◆该算法的I/O数估计

R读入内存  
的遍数

$$\frac{B(S)}{M-1} \times B(R) + B(S)$$

S块数

R块数

$$\frac{B(S) \times B(R)}{M-1} + B(S)$$

$$\approx \frac{B(S)B(R)}{M}$$

不同分配方案有什么影响?

## 嵌套循环连接算法

---

- 一个关系读取一次，另一个反复读
- 可以用于任何大小的关系间的连接操作，  
不必受连接操作所分配的内存空间大小的  
限制

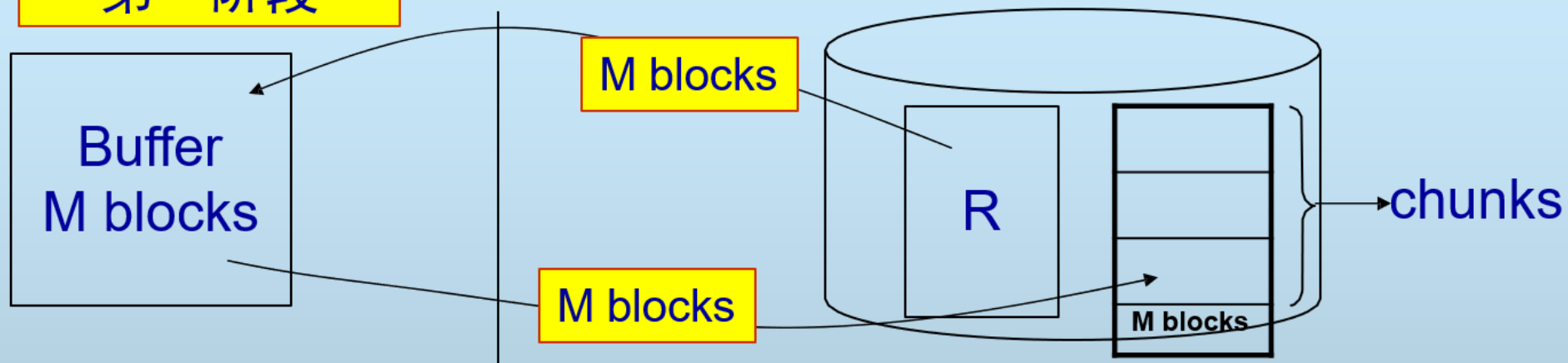


## Example

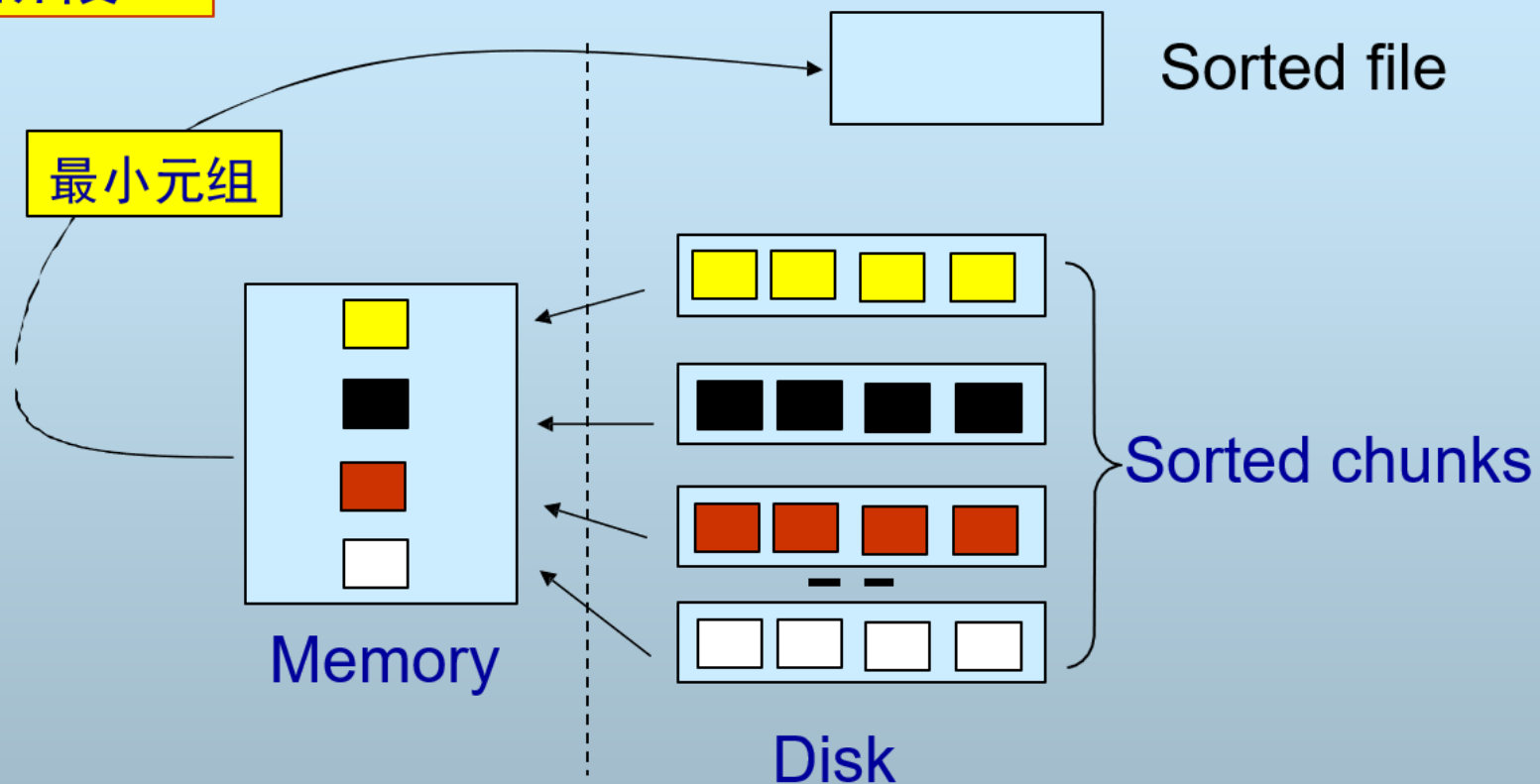
对Y, R和S有序

<b>i</b>	<b>R.Y</b>	<b>S.Y</b>	<b>j</b>
<b>1</b>	<b>10</b>	<b>5</b>	<b>1</b>
<b>2</b>	<b>20</b>	<b>20</b>	<b>2</b>
<b>3</b>	<b>25</b>	<b>20</b>	<b>3</b>
<b>4</b>	<b>30</b>	<b>30</b>	<b>4</b>
<b>5</b>	<b>40</b>	<b>30</b>	<b>5</b>
		<b>50</b>	<b>6</b>
		<b>52</b>	<b>7</b>

## 第一阶段



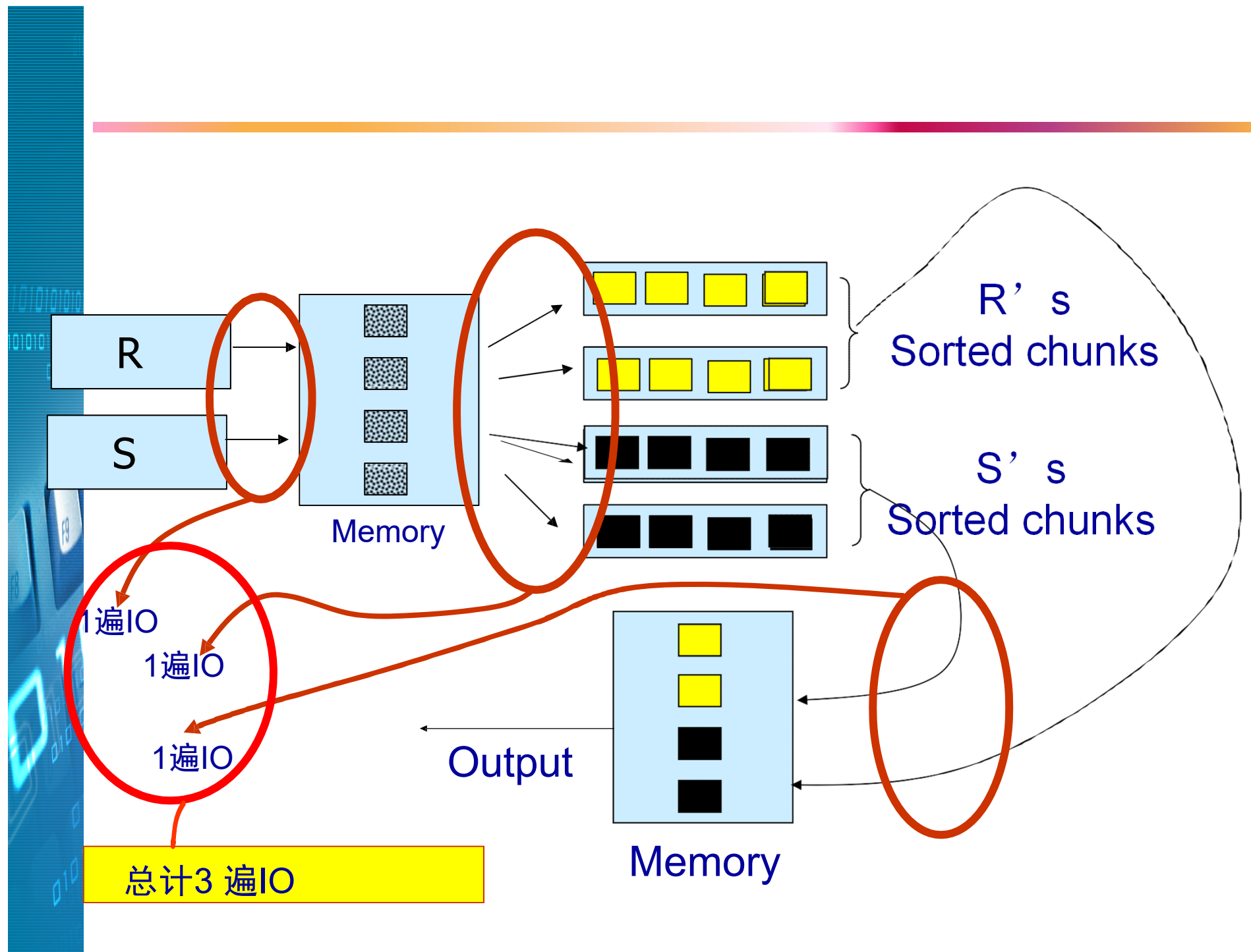
## 第二阶段





# 基于排序的联接算法

- 先对两个关系分别做两阶段多路归并排序;
  - 代价:  $4B(R) + 4B(S)$  次 I/O;
  - 主存要求:  $M \geq [\max(B(R), B(S))]^{1/2}$
- 归并排序后的  $R'$  和  $S'$ , 仅需要两个缓冲区;
  - 代价:  $B(R) + B(S)$  次 I/O;
  - 主存要求:  $M \geq 2$
- 总计需要的磁盘 I/O 次数:  $5[B(R) + B(S)]$





- 
- 
- 将排序的第二阶段与连接合并;
    - 两个关系的每个排序好的子表对应一个缓冲块
    - 主存要求:  $M \geq [B(R) + B(S)]^{1/2}$
    - 用M个缓冲区计算  $R(X,Y) \bowtie S(Y,X)$ , Y做排序关键字
    - 代价(节省1次I/O):  $3 B(R) + 3B(S)$  次I/O;

- 
- 
- 了解执行过程
  - 了解同一个查询操作，因关系的规模、可用内存的大小、数据存放特征不同，可以采用不同的执行策略，操作时间也不同。