应用于高能物理实验数据的基数排序算法

谌炫

核探测与核电子学国家重点实验室
中国科学院高能物理研究所

内容

- 高能物理实验数据的特点和排序需求
- 常见的基于比较的排序算法
- 非基于比较的排序算法：基数排序、计数排序
- 算法针对实验数据的测试情况

高能物理实验数据的特点

- 基于时间标记

  时间戳

- 分段有序

  多电子学通道、并发读取

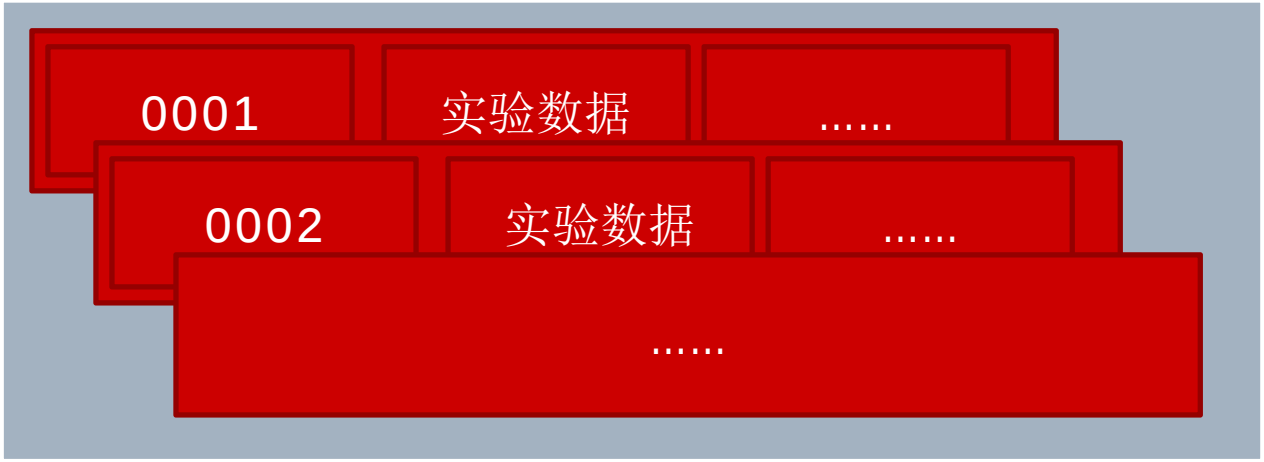
- 数据量大

  GBps级别

高能物理实验数据的特点

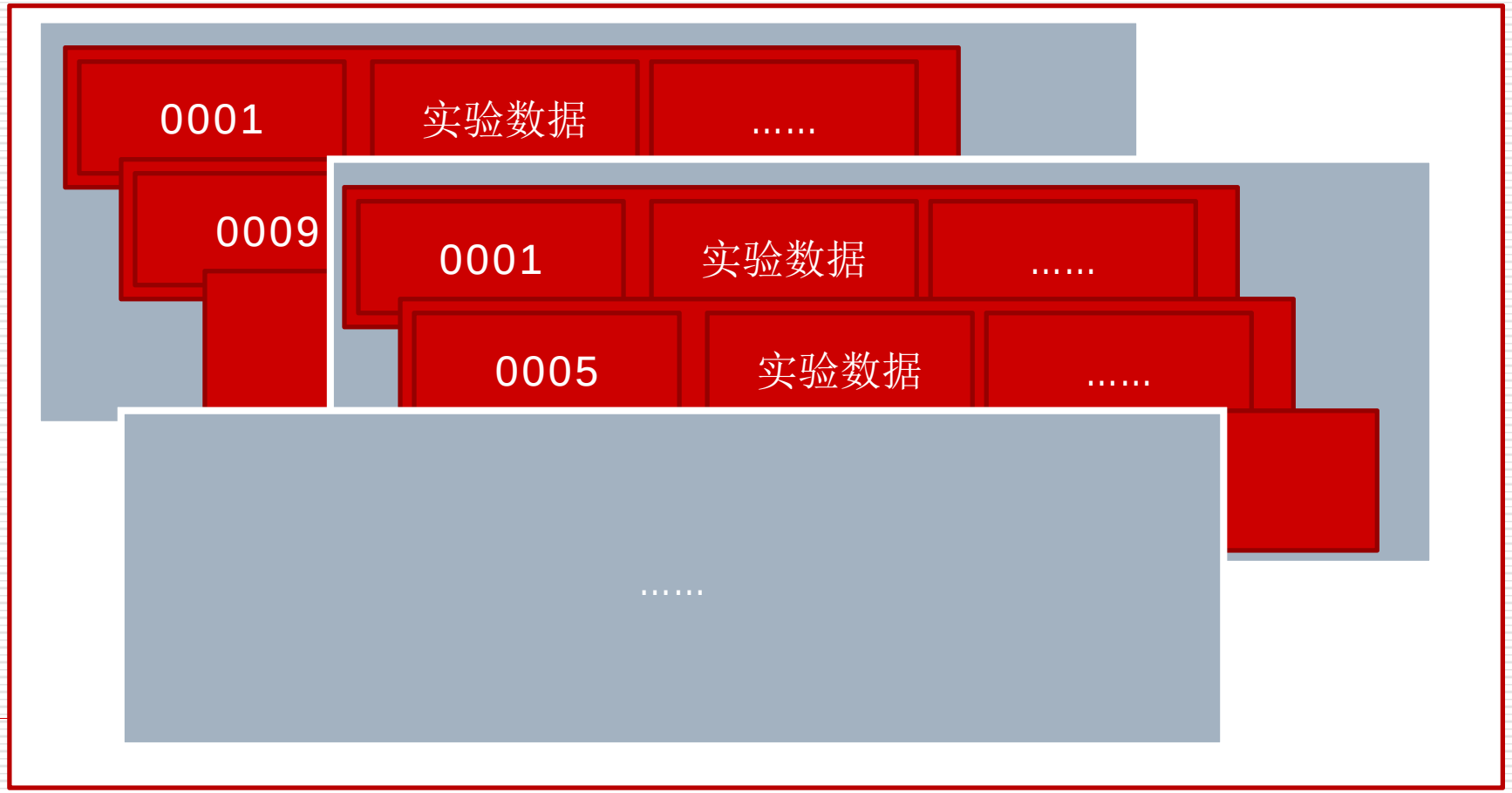
- 单位数据

- 单位通道数据集



高能物理实验数据的特点

- 总数据集



常见的排序算法

- 归并排序

  时间复杂度：$O(N\log_2 N)$

- 快速排序

  时间复杂度： $O(N\log_2 N)$

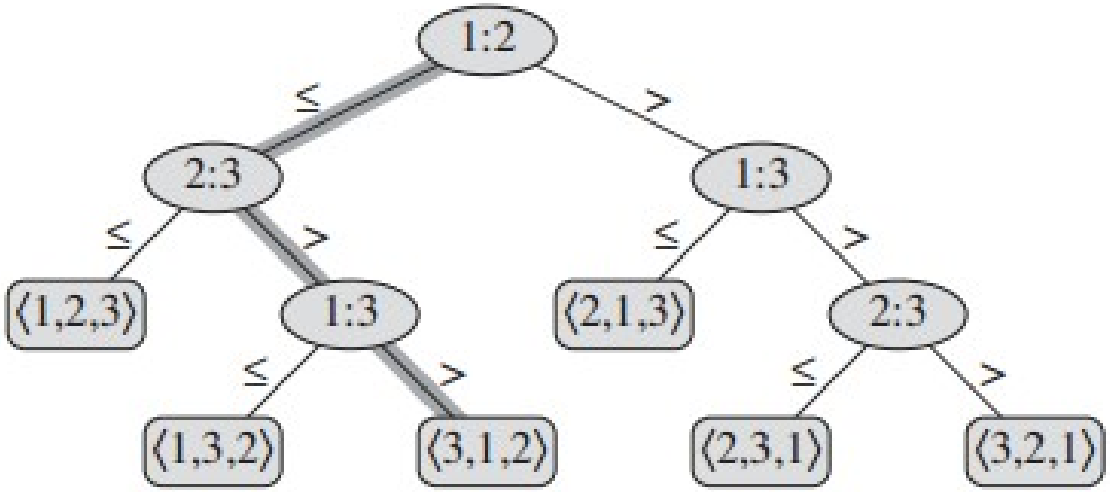
- 堆排序、插入排序等等

基于比较的排序

基于比较的排序算法

- 通过比较关键字的大小确定数据的顺序
- 时间复杂度的下界是： $\Omega(N\log_2 N)$

  通过决策树模型可以证明



基数排序

- 基数排序假设对 n 个数据元素进行排序，每个元素的排序关键字都有 d 位， 则可以对这 n 个数据元素从高位到低位或者从低位到高位依次内排序并最终实现对这 n 个整数的排序。
- 非基于比较的排序
- 时间复杂度：取决于子排序算法的时间复杂度
- 子排序必须为稳定排序
- 基数排序算法描述如下：

```
RADIX-SORT(A, d)
1 for i <- 1 to d
2    用一种稳定排序算法对 A 中数据元素的第 i 位进行排序
```

基数排序示例

- 对5个三位数的排序

原数据	对个位排序	对十位排序	对百位排序
152	250	229	152
566	152	250	229
589	566	152	250
250	589	566	566
229	229	589	589

基数排序示例

- 对5个时间数据的排序

原数据	对秒位排序	对分位排序	对时位排序
06:28:00	06:28:00	11:26:00	06:28:00
11:55:30	11:26:00	06:28:00	11:26:00
23:28:45	11:55:30	23:28:45	11:55:30
11:26:00	23:28:45	16:28:59	16:28:59
16:28:59	16:28:59	11:55:30	23:28:45

计数排序

- 计数排序假设 n 个输入数据元素的排序关键字中的每一个都是介于 0 到 k 之间的整数，此处 k 为某个整数。计数排序的基本思想是对每一个输入元素 x，确定出小于 x 的元素个数，这样就可以把 x 直接放到它在最终输出数组中的位置上。
- 非基于比较的排序算法
- 时间复杂度：O(N)；稳定

计数排序

- 计数排序算法描述如下：

```
COUNTING-SORT(A, B, k)
1 for i <- 0 to k
2    do C[i] <- 0
3 for j <- 1 to length[A]
4    do C[A[j]] <- C[A[j]] + 1
5 for i <- 1 to k
6    do C[i] <- C[i] + C[i - 1]
7 for j <- length[A] downto 1
8    do B[C[Aj]]] <- A[j]
9         C[A[j]] <- C[A[j] - 1
```

计数排序示例

- 对8个取值范围在[0, 5]之间数字的排序

  A[1..n] B[1..n] C[0..k]

序号	1	2	3	4	5	6	7	8
A	2	5	3	0	2	3	0	3

(a)

序号	0	1	2	3	4	5
C	2	0	2	3	0	1

序号	0	1	2	3	4	5
C	2	2	4	7	7	8

(b)

计数排序示例

- 对8个取值范围在[0, 5]之间数字的排序

  A[1..n] B[1..n] C[0..k]

序号	1	2	3	4	5	6	7	8
B							3	

序号	0	1	2	3	4	5
C	2	2	4	6	7	8

(c)

序号	1	2	3	4	5	6	7	8
B		0					3	

序号	0	1	2	3	4	5
C	1	2	4	6	7	8

(d)

计数排序示例

- 对8个取值范围在[0, 5]之间数字的排序

  A[1..n] B[1..n] C[0..k]

序号	1	2	3	4	5	6	7	8
B		0				3	3	

序号	0	1	2	3	4	5
C	1	2	4	5	7	8

(e)

序号	1	2	3	4	5	6	7	8
B	0	0	2	2	3	3	3	5

(f)

以计数排序作为子排序的基数排序算法

- 时间复杂度：*O*(N)；突破了基于比较的排序算法的下界
- 算法描述

```
RADIX -COUNTING-SORT(A, B, K, d)
1 for i <- 1 to d
2     COUNTING-SORT(A, B, K[i])
3     for j <- 1 to length[A]
4         A[i] = B[i]
```

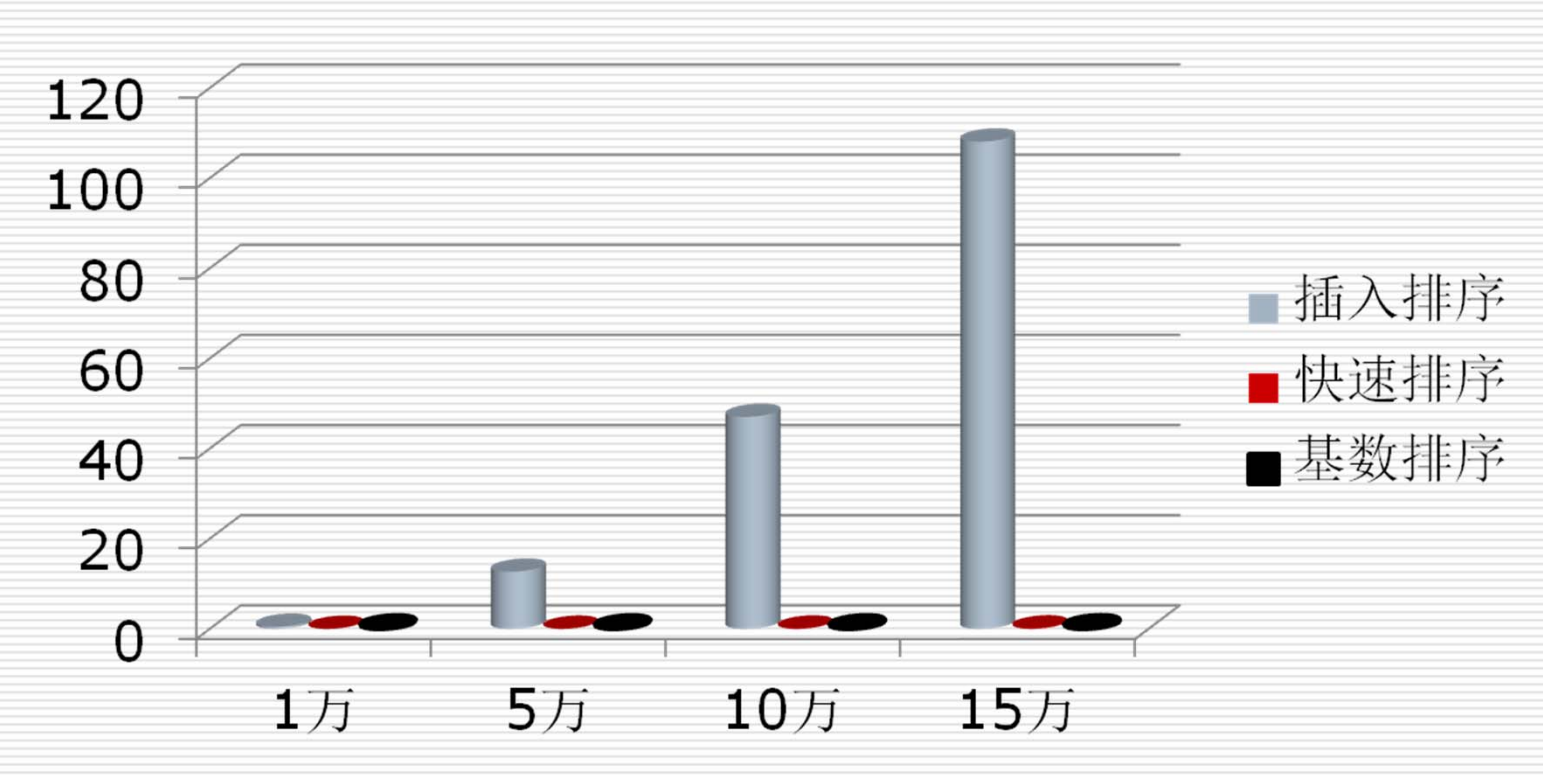
算法测试

- 随机数据

	1万	5万	10万	15万
插入排序	0.4748124	12.6790326	47.033886	108.1055356
快速排序	0.0047062	0.0284178	0.062906	0.0979182
基数排序	0.0008162	0.00475524	0.0138398	0.023450

算法测试

- 随机数据



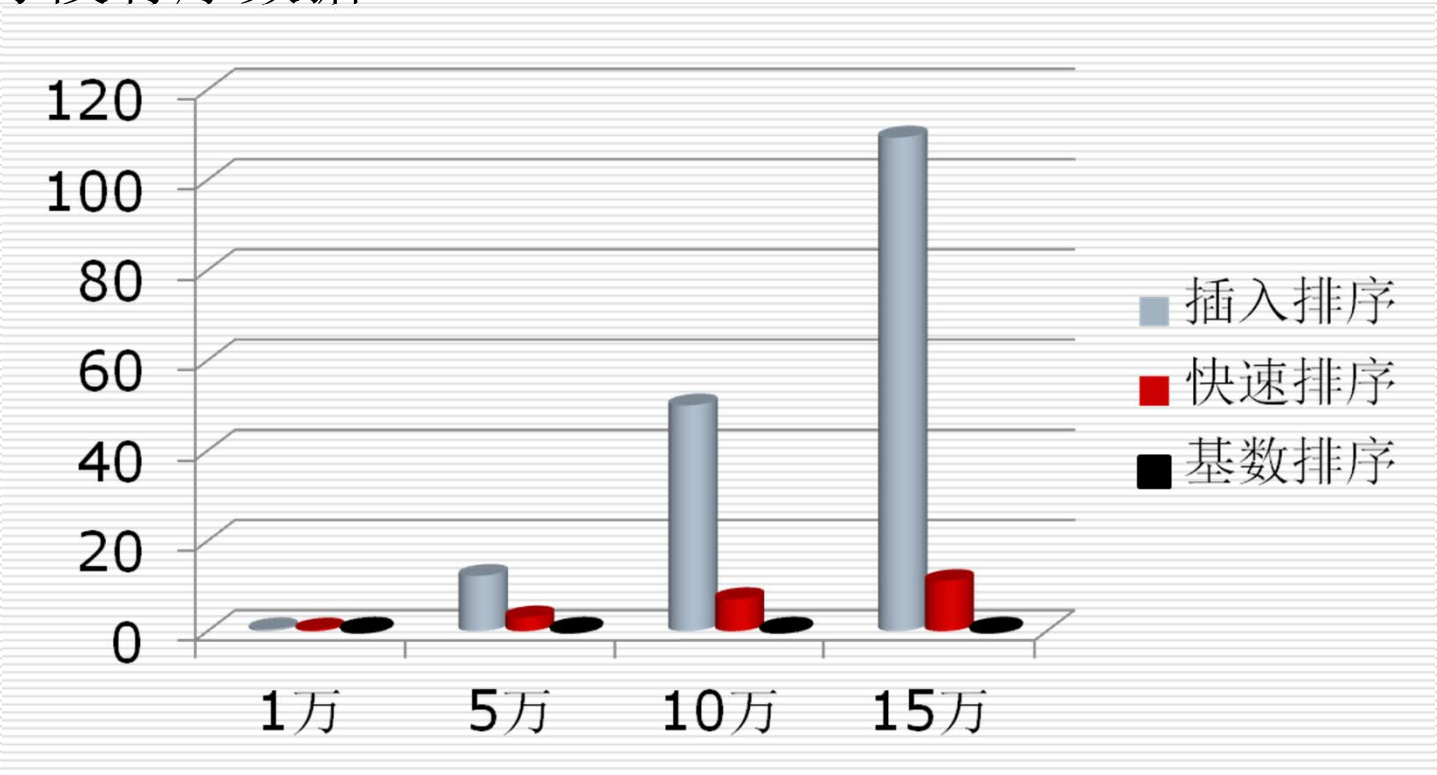
算法测试

- 分段有序数据

	1万	5万	10万	15万
插入排序	0.436084125	12.27214538	50.06480188	109.4273941
快速排序	0.37580875	3.062451	7.110748125	11.1119805
基数排序	0.00082	0.004785375	0.0132515	0.02281125

算法测试

- 分段有序数据



谢谢！