

题目：某个机器的配置为：CPU 8 cores, MEM 4G, HDD 4T

这个机器上有一个 1T 的无序数据文件，格式为 (key_size, key, value_size, value)

设计一个索引结构，使得并发随机地读取每一个 key-value 的代价最小

允许对数据文件做任意预处理，但是预处理的时间计入到整个读取过程的代价里

预处理：

对 1T 数据进行排序：

1T 数据，数量 $n = 1T / (\text{key_size} + \text{value_size})$;

需要对 n 条数据按照 key 的二进制顺序排列。

先将 1T 文件分为 x 份小文件，每份数据量 n/x ，使得 $n/x * \log(n/x)$ 小于 4G 并接近 4G。

对每份小文件进行快排，时间复杂度 $T = O(n/x \log n/x)$ ，由于 8 核，

时间更新为 $T = O(n/x \log n/x) / 8$

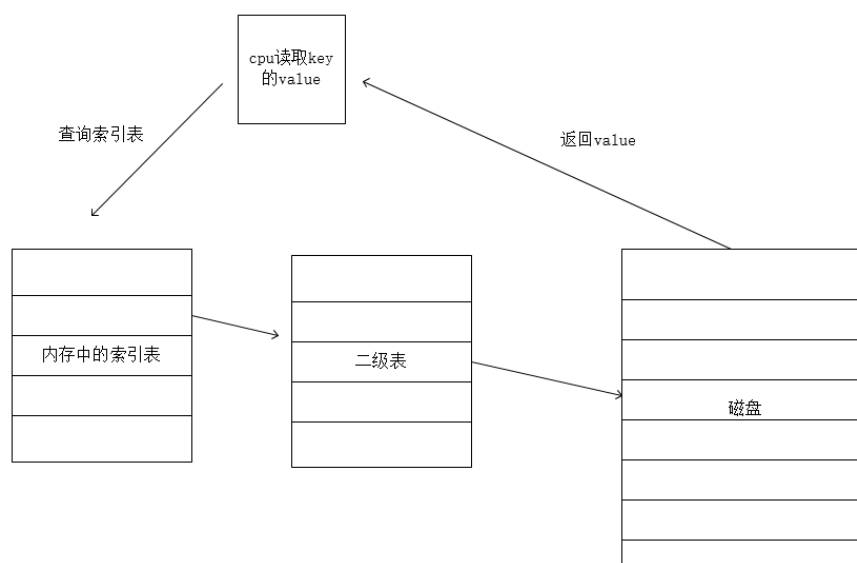
对于 IO 时间，需要做 n 次 IO 处理，

所以， $T_{\text{总}} = T + n * T_{\text{IO}}$ 。

之后所有小文件进行归并排序，时间复杂度 $T = O(n \log n)$ ，IO 时间 n 次 io 处理。

此时 $T_{\text{总}} = O(n/x \log n/x) / 8 + O(n \log n) + 2 * n * T_{\text{IO}}$

随机读取一个 key-value：



在预处理阶段，我们已经按照如上结构存储好了数据。
其中，内存中的索引表表项结构如下：

Beg	ptr
64MB	0X0040

主要有两部分，一个是 **beg**，表示某连续区域的起始地址。一个是 **ptr**，指向该地址的二级表项指针。
一个索引表表项对应的磁盘空间为 **32MB** 的连续空间。

二级表表项：

state	lock	modify	value
1	1	1	Value_ptr

State:是否被访问以及权限
Lock: 写时锁
Modify:是否被修改
Value:数据指针

时间复杂度：
一次随机读取时间：
 $T = 2 * T_IO$

综上，算上预处理时间，平均一次随机 IO 时间：

$$T=O(\frac{1}{x}\log \frac{n}{x})+O(\log n)+2*T_IO$$