

友乾营·第十六期

大数据下大批量随机键值的查询

I 大批量键值查找

单条记录查找：

数据表上按键值建立索引，查找复杂度只有 $\log N$ 次，10亿行数据也只要比较30次，在现代计算机上只需要几毫秒。

大批量键值查找：

多达几千甚至几万个键值，再简单地使用数据库索引，时间延迟由此会涨到几十分钟甚至小时级别。

I 行存与列存

需要取出整条记录的场景，行存要比列存更适合

A3

Index	id	data
1	1	vf5mous8qnwc3bp24y6tz79ax0ihd1jrlkge
2	2	cvs0fpehx65wqzm3bk02dy4j9r7inl1g8ua
3	3	ieoxcb0kdwts9fqj1p8h5nmurz43gav72ly6
4	4	wcx96mpiur4sf1vage8zodhb5n02ykjtq7l3
5	5	ewma6znge4chr52uyjp1bfsq3t8lv9i7xdk0
6	6	do56m9bin8xa1c30hgy7qtusrz2w4fjlvekp
7	7	ieqna69bcthoخد108k3flpw2rjmvzy4u57s
8	8	ocvh2ek0ptfzqx14n57aid68lmyujgr3b9sw
9	9	cgsiub74nje185qv3hrao2kwmylz60px9dft
10	10	xnliom6zbesrg7k8yf39512duqjpt4cwwh0a

列式存储

A3

Index	id	data
1	1	vf5mous8qnwc3bp24y6tz79ax0ihd1jrlkge
2	2	cvs0fpehx65wqzm3bk02dy4j9r7inl1g8ua
3	3	ieoxcb0kdwts9fqj1p8h5nmurz43gav72ly6
4	4	wcx96mpiur4sf1vage8zodhb5n02ykjtq7l3
5	5	ewma6znge4chr52uyjp1bfsq3t8lv9i7xdk0
6	6	do56m9bin8xa1c30hgy7qtusrz2w4fjlvekp
7	7	ieqna69bcthoخد108k3flpw2rjmvzy4u57s
8	8	ocvh2ek0ptfzqx14n57aid68lmyujgr3b9sw
9	9	cgsiub74nje185qv3hrao2kwmylz60px9dft
10	10	xnliom6zbesrg7k8yf39512duqjpt4cwwh0a

行式存储

I 内容概要

1) 单字段键

2) 多字段键

3) 多线程查询

4) 索引冗余机制

5) 数据追加的处理

I 单字段键——数据生成

我们以下面这种数据结构为例

字段名称	类型	是否主键	说明
id	int	是	从100000000000001开始自增
data	string		需要获取的数据

按以上的数据结构，在文本文件中创建6亿条数据，可以这样写：

	A	B
1	1234567890qwertyuiopasdfghjklzxcvbnm	
2	=file("single600m.txt")	
3	for 6000	=to((A3-1)*100000+1,A3*100000).new(~+100000000000000:id,rands(A1,rand(40)+160):data)
4		=A2.export@at(B3)

I 单字段键——题目

友乾营

大批量随机键值查找

从6亿条记录中，找出由1万个随机分布键值集对应的记录

I 单字段键——索引原理

二分查找举例：找到id为82的用户信息

	id	score
	12	2374
	16	4180
	17	8515
	19	1887
第一次查到25	25	7900
	34	8398
第二次查到78	62	2277
	78	1662
第三次查到82	82	5955
	99	4495

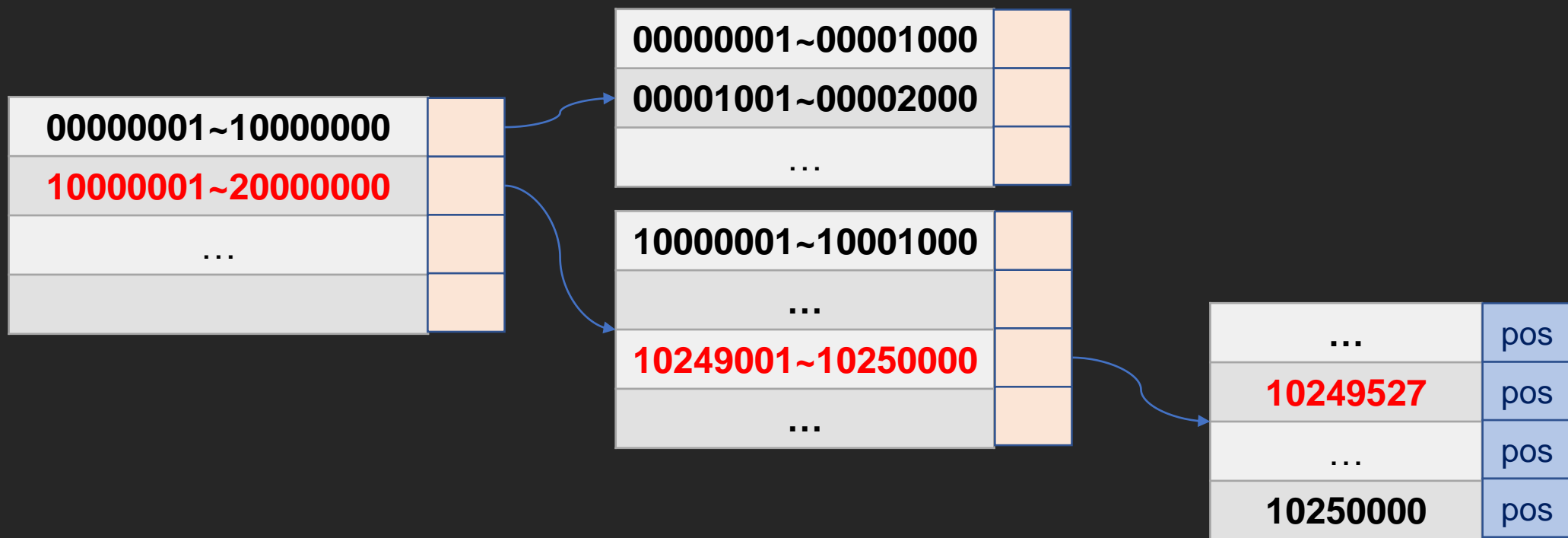
有序users表

本例中，顺序查找（遍历）需要9次比较，二分法只用了3次比较。

顺序查找的时间复杂度为 $O(n)$ ；
二分查找的时间复杂度为 $O(\log_2 n)$ 。

单字段键——分级索引

大索引内存装不下，也没必要全装下，可采用分级索引



三级索引示意图

I 单字段键——键值类型

对于非整数键值，应当先转为整数

车牌号	数据列
京A12345	...
沪B56789	...
...	...

数字化

车牌号	数据列
100112345	
110256789	

身份证	数据列
110105197608028736	...
310104199809209731	...
...	...

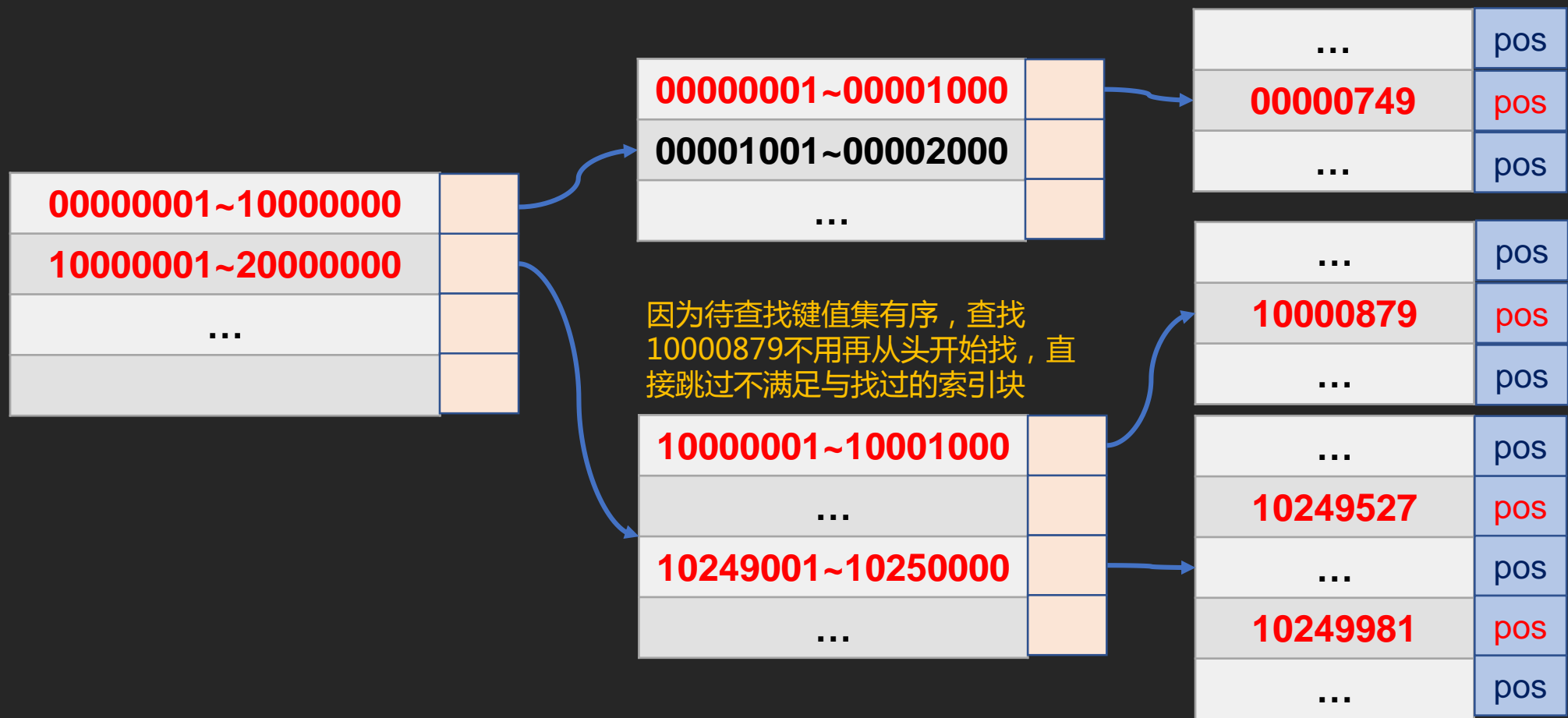
多层序号

身份证	数据列
11 01 05 19 76 08 02 87 36	...
31 01 04 19 98 09 20 97 31	...
...	...

I 单字段键——键值排序

待查找键值集有序，查找时可以少走回头路

举例：根据有序键值集（749、...、10000879、...、10249527、10249981、...）查找



因为待查找键值集有序，10249527、10249981这两个键值落在同一索引块中，不用回退比较

单字段键——索引缓存

预先加载索引缓存提升查询效率

直接使用索引进行随机键值查询：

第一次查询，耗时:80秒

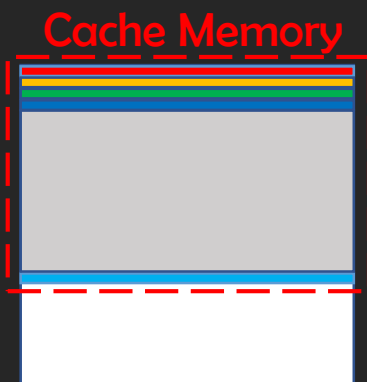
第二次查询，耗时:78秒

第三次查询，耗时:77秒

第四次查询，耗时:76秒

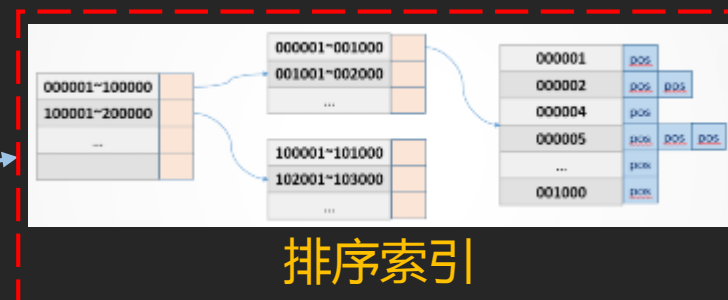
.....

第N次查询，耗时:25秒

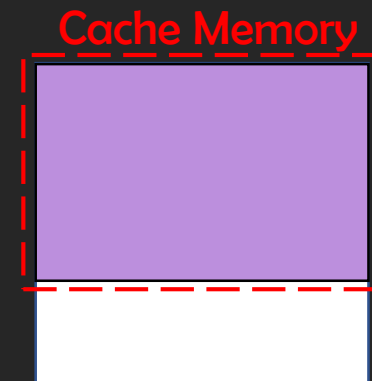


每次使用索引进行键值查询时，操作系统会产生缓存。

N次使用索引查询后，效率才会达到极限值。



预先加载索引缓存：
每次随机键值查询耗时
约:25秒



可以预先加载索引缓存，以便使每次查询均处于最高效状态。

I 单字段键——建组表与索引

利用已经创建好的文本文件，生成一份行存组表文件

	A
1	=file("single600m.txt").cursor@t()
2	=file("single600m.ctx").create @r(#id,data).append(A1)

需要注意：表中应对键值id有序，因为当键值集较为集中时，可以相对减少磁盘跳动

对该组表文件建立id键的排序索引

	A
1	=file("single600m.ctx").create().index(id_idx;id)

I 单字段键——组表查询

随机生成1万个id，利用索引查询

	A	B
1	=file("single600m.ctx").create()	/打开组表
2	=A1.index@3(id_idx)	/加载三级索引缓存
3	=10000.(10000000000000+(rand(6000000000)+1)).sort()	/随机取1万个有序键值
4	=now()	/当前时间
5	=A1.icursor(;A3.contain(id),id_idx).fetch()	/利用索引批量键值查找
6	=interval@ms(A4,now())	/查找花费时间

查询时需要注意两点：

- 1、键值排序：待查找键值集有序
- 2、索引缓存：查找前预先加载索引缓存

I 单字段键——Oracle导入数据并建索引 | 友乾营

将已经创建好的文本文件数据导入到Oracle中

Oracle建表：`create table single600m (id number(13),data varchar2(200));`

将文本文件内容，使用 Oracle 的 SqlLoader 导入表中（略）

Oracle建索引：`create unique index idx_id_600m on single600m(id);`

I 单字段键——Oracle查询

随机生成1万个id，利用索引查询

	A	B
1	<code>=10000.(10000000000000+rand(6000000000)+1).sort()</code>	/随机取1万个键值
2	<code>=A1.group((#-1)\1000)</code>	/每1000个键值为一组
3	<code>=connect("oracle")</code>	/建立数据库连接
4	<code>=now()</code>	/当前时间
5	<code>=A2.(A3.query("select * from single600m where id in (?)",~)).conj()</code>	/合并多组查询结果
6	<code>=interval@ms(A4,now())</code>	/查找花费时间
7	<code>>A3.close()</code>	/关闭数据库连接

按每1000条分为若干组是因为数据库的in中最大支持1000个数

I 多字段键

我们以下面这种数据结构为例

字段名称	类型	是否主键	说明
type	string		可枚举
id	int		每种枚举类型的 id 都从 1 开始自增
data	string		需要获取的数据

其中 type 和 id 两个字段作为联合主键确定一条记录

I 多字段键——合并主键

涉及到集合的存储和比较，相对单字段键要慢一些
为了获取高性能，更常用的办法是把多字段键拼成单字段键

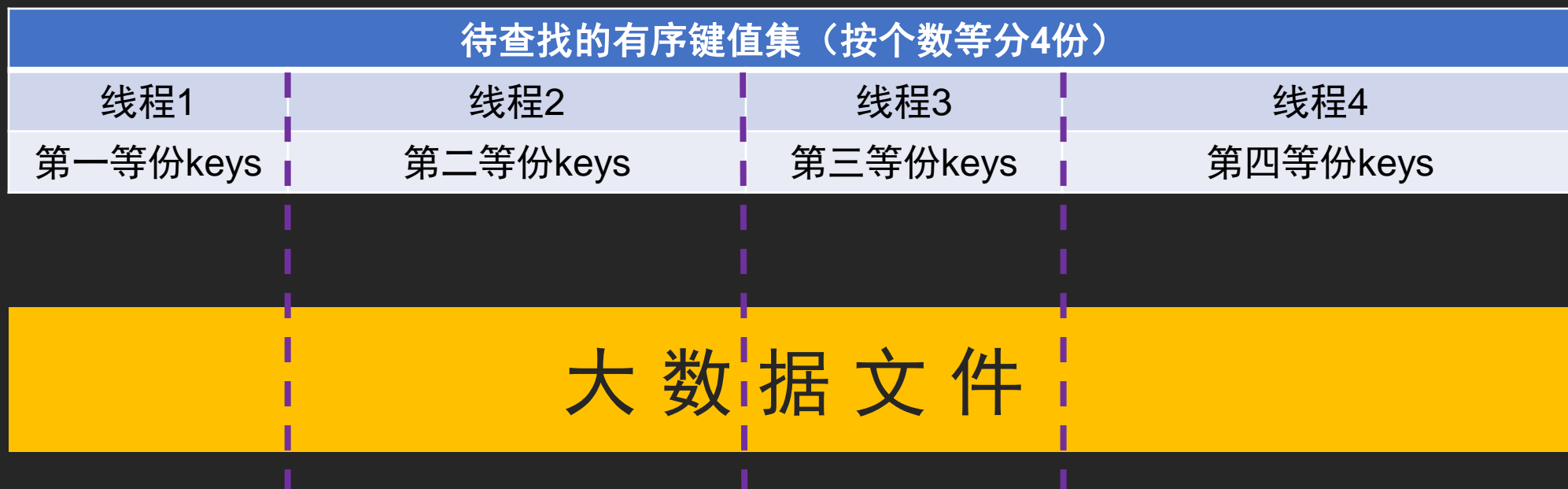
type	id		nid
type_a	934875		10010934875
type_a	934876		10010934876
...
type_b	137		10020000137
type_b	138		10020000138
...

对于合并主键后的nid，再按照单字段键的做法处理即可，注意需要确保nid有序

I 多线程查询

用多线程并行方式来进一步提高性能

举例：数据文件，4线程，有序keys按顺序分4段



I 多线程查询——组表

用多线程并行方式来进一步提高性能

	A	B
1	=file("single600m.ctx").create()	
2	=A1.index@3(id_idx)	
3	=10000.(10000000000000+(rand(6000000000)+1)).sort()	
4	=A3.group((#-1)\1000)	
5	=now()	
6	fork A4	=A1.icursor(;A6.contain(id),id_idx)
7		=B6.fetch()
8	=A6.conj()	
9	=interval@ms(A5,now())	

需要注意：

- 1、将A3中的随机键值集排序后，A4按A3的行号分割，这样可以确保每个线程的键值集集中
- 2、取数动作要在每个线程中完成，这样才能真正的并行

I 多线程查询——Oracle

用多线程并行方式来进一步提高性能

	A	B
1	=10000.(10000000000000+rand(6000000000)+1).sort()	
2	=A1.group((#-1)\1000)	
3	=now()	
4	fork A2	=connect("oracle")
5		=B4.query("select * from single600m where id in (?)",A4)
6		>B4.close()
7	=A4.conj()	
8	=interval@ms(A3,now())	

按每1000条分为若干组是因为数据库的in中最大支持1000个数

I 索引冗余机制

数据需要遍历时往往会采用列存，但不适合查找
索引冗余机制可以用于提高列存数据的随机查找性能

列存组表的建立

	A
1	=file("single600m.txt").cursor@t()
2	=file("single600m.ctx").create@r(#id,data).append(A1)

建立带值索引

	A
1	=file("single600m.ctx").create().index(id_idx;id;data)

使用带值索引文件查找，不再访问原列存组表，效率虽然比行存的无冗余排序索引还要好，但需要付出比原列存组表占用磁盘空间还要大的代价

I 测试结果对比（一）

从 6 亿条数据总量中取 10 万条批量随机键值

测试环境

处理器	Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz两颗
内存	64G
硬盘	SAS 1TB
操作系统	centos6.8(64 位)

测试结果

耗时（毫秒）					
单线程			多线程（10线程）		
Oracle	行存组表	索引冗余	Oracle	行存组表	索引冗余
117322	20745	19873	39549	10975	9561

I 测试结果对比（二）

从 6 亿条数据总量中取 10 万条批量随机键值

从 12 亿条数据总量中取 10 万条批量随机键值

测试环境

处理器	Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz两颗
内存	64G
硬盘	SSD 1TB
操作系统	centos6.8(64 位)

测试结果

耗时（毫秒）			
单线程		多线程（10线程）	
Oracle	行存组表	Oracle	行存组表
56671	23990	35184	13264
Oracle	行存组表	Oracle	行存组表
151089	24421	95987	14623

特征总结：集算器的索引性能基本上只和取出量有关，和总数关系不大，但传统数据库的索引性能和总数关系很大

数据更新

修改后文件数据的更新

date	price	...
...		
2019-04-23	50	
2019-04-24	50	
2019-04-25	50	
2019-04-26	50	
2019-04-27	50	

date	price	...
2019-04-24	51	
2019-04-26	49	

update数据(补区)

date	price	...
...		
2019-04-23	50	
2019-04-24	51	
2019-04-25	50	
2019-04-26	49	
2019-04-27	50	

当近期累积增量数据发生变化后

	A	B
1	<code>=add_file.create().update(update_data)</code>	/修改文件数据
2	<code>=add_file.reset@q()</code>	/快速重整补区数据

快速重整表示仅重整从第一次出现补区数据后的部分。之前的数据不用重写。

I 数据追加

键值有序时，直接追加新的数据即可

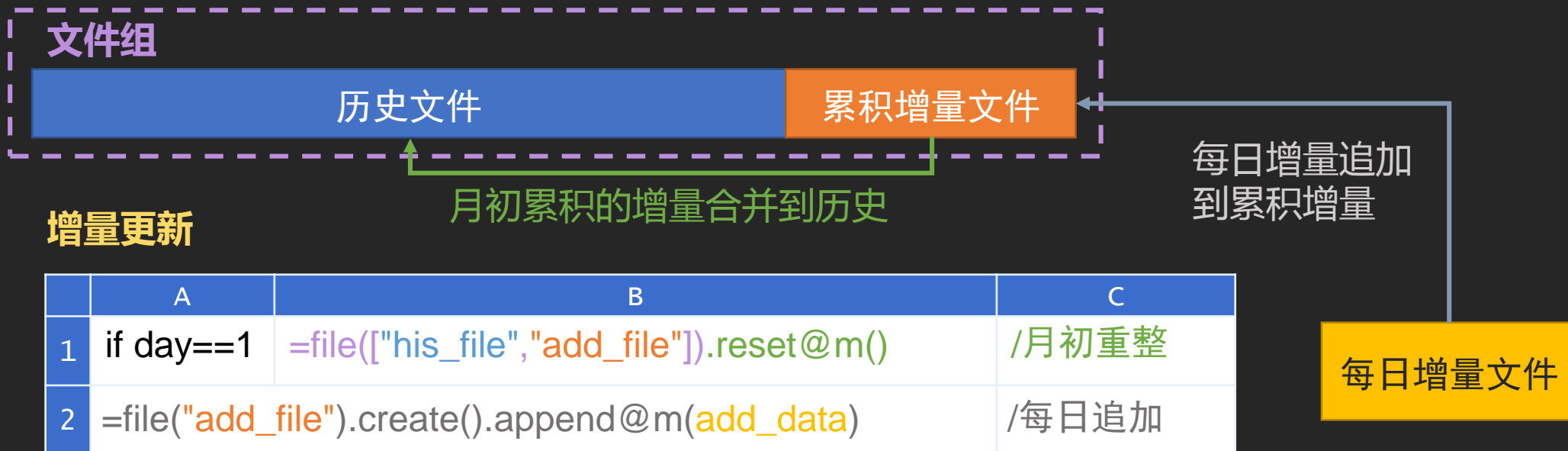
键值无序时，新、旧文件确保有序，再归并排序

	A	B
1	=file("single600m.ctx")	
2	=A1.create().cursor()	/组表游标
3	=file("singleadd.txt")	
4	=A3.cursor@t()	/新增txt游标
5	=file("single.ctx_temp").create(#id,data)	/建立新的组表
6	=A5.append([A2,A4].mergex(id))	/归并排序后结果存入新的组表

需要注意：这里的A1、A3中的组表和txt需要对id有序

I 数据追加

日增数据文件的追加



文件组查询

	A	B
1	<code>=file(["his_file","add_file"])</code>	/文件组
2	<code>=A1.create().icursor(;id=="3197608180")</code>	/查询

I 下回分解

- 集群下的大批量随机键值查找

敬请期待

好多乾

润乾线上直销系统



玩转好多乾

<http://www.raqsoft.com.cn/wx/hdq-strategy.html>