CCF BDCI CCF BIG DATA & COMPUTING INTELLIGENCE CONTEST

2020 8th

CCF 大数据与计算智能大赛

赛题名：数据湖的元信息发现与分析

队伍名称：别忘了我们是学控制的

赛题简介

CCF BIG DATA & COMPUTING INTELLIGENCE CONTEST

赛题目的：通过模拟数据湖分析场景，在云平台环境上为纷繁复杂的云数据构建元信息，实现对大数据集信息的发现。

赛题内容：**数据湖**场景下，给定一批目录及**CSV文件数据**，结合**索引优化**、**模糊匹配**等相关技术，对给定的**查询条件**，**快速过滤**并**准确计算**出满足条件的记录行的总数。

赛题分析

JAVA语言

大规模文本数据

模糊查询

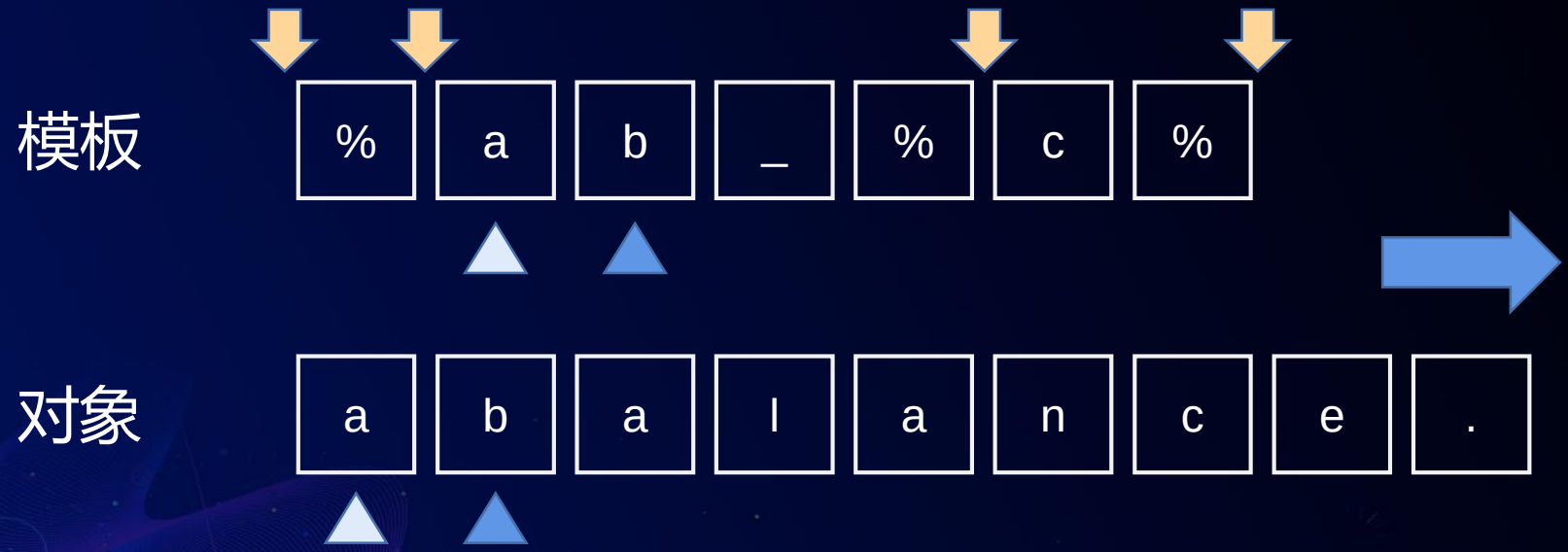
高效率

基本思路

1. 数据载入采用高IO性能与位置索引特性的Byte数组。
2. 自建快速模糊查询算法。
3. 配合高效的目录索引和列索引机制。
4. 充分利用多线程实现并行处理。
5. 充分考虑JAVA的内存回收机理。

关键技术

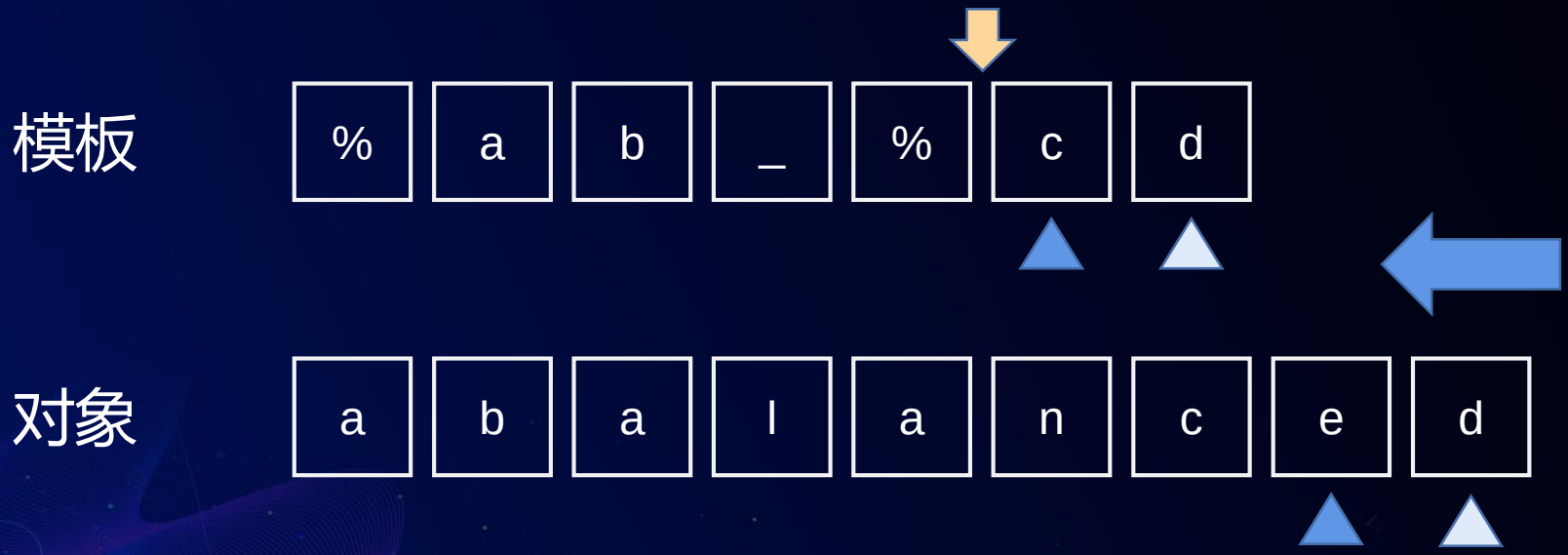
针对Byte数组的快速模糊查询引擎



- 每一个“%”表示一个新的匹配区间的开始。

关键技术

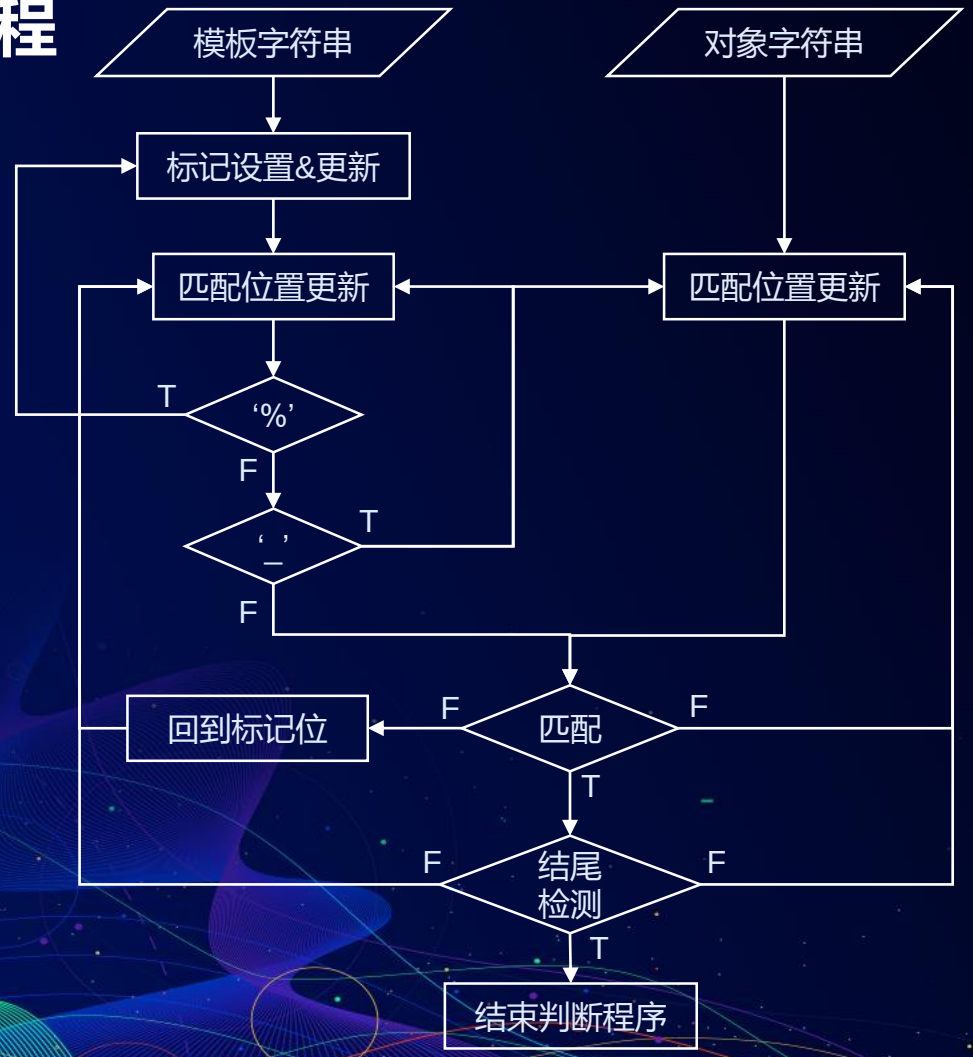
反向匹配



- 通配符标记再一次发挥作用。

关键技术

基本匹配流程

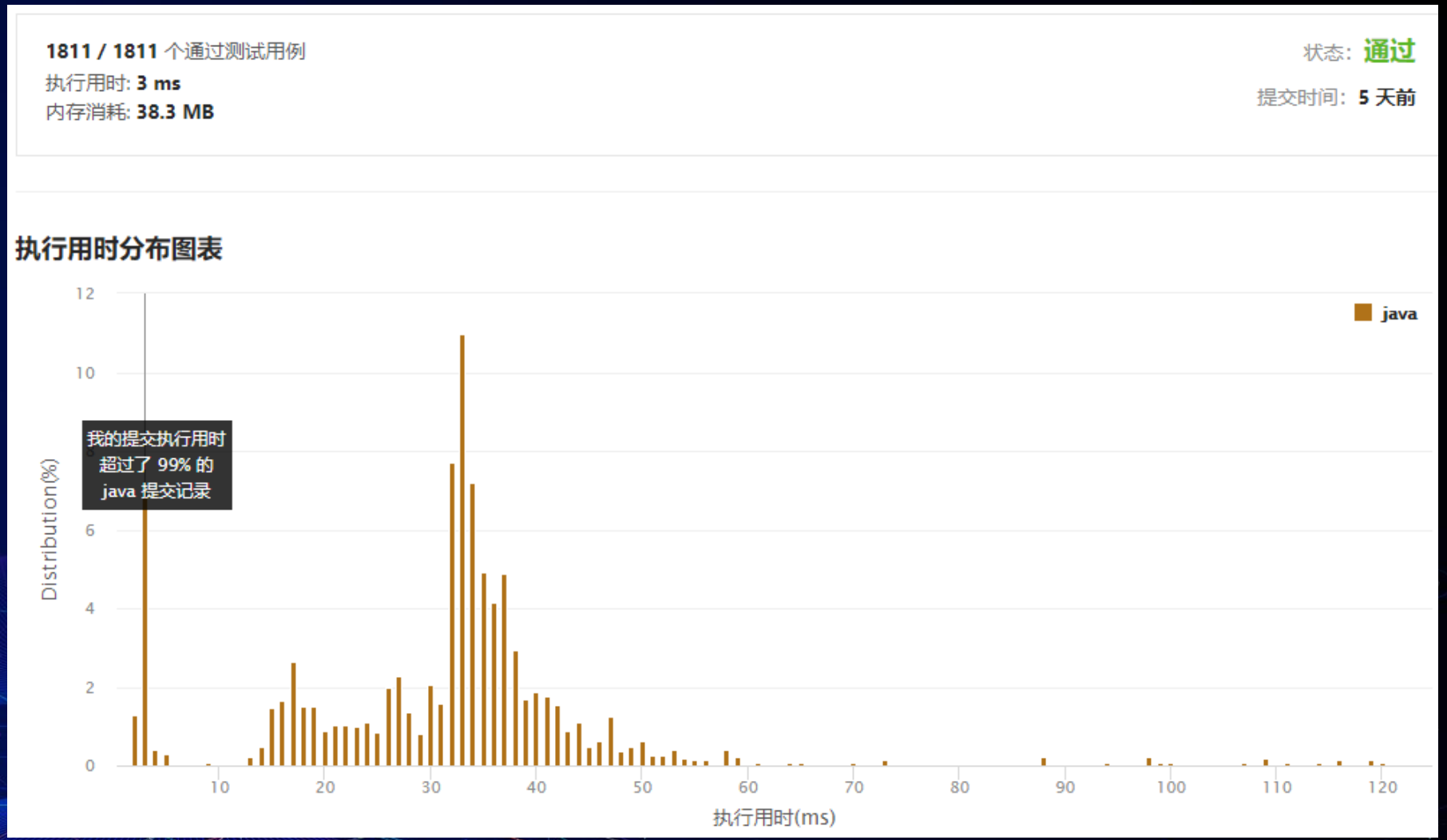


- 此匹配算法的比对周期不会长于通配符的区间长度，对于超长的匹配对象具有很高的匹配效率。
- 结合集合的逻辑判断特性，对于ANY_LIKE、ALL_LIKE、NONE_LIKE等组合查询设计提前中止条件。
- 适用于其它字符类数组结构以及数据流模式。

关键技术

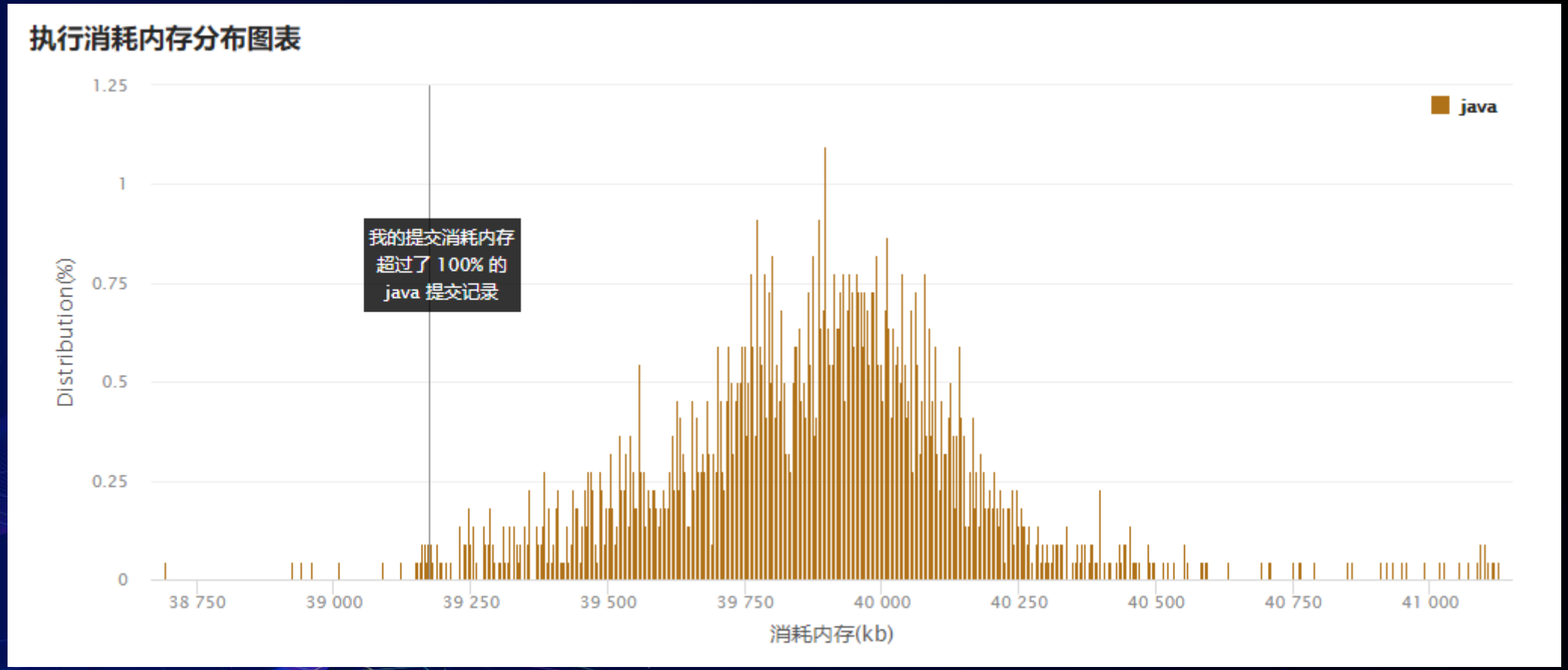
算法性能

力扣(leetcode)
44.通配符匹配



关键技术

算法性能


执行消耗内存分布图表
1.25
1
0.75
0.5
0.25
0
Distribution(%)
我的提交消耗内存
超过了 100% 的
java 提交记录
java
38 750
39 000
39 250
39 500
39 750
40 000
40 250
40 500
40 750
41 000
消耗内存(kb)

方案设计

Step1 建立目录索引

1.1 分 table 扫描数据目录和文档大小，建立**子目录索引**

1.2 建立 **HashMap <table名称，目录索引信息>**

Step2 逐条进行参数查询，分级查询 [table] [目录路径] [列数据]

2.1 建立**文档列索引**与 **HashMap<目录路径，Byte[]数据>**

2.2 利用目录索引和查询引擎进行**目录匹配**

2.3 利用列索引和查询引擎进行**列数据匹配**

Step3 提取记录信息，生成输出文件

方案优化

- **自定义高兼容性数据类**，结构化管理索引数据与查询参数信息，方便快速提取信息。
- **建立线程池**，在建立列索引和查询列数据时，均以末端子目录为单元建立多线程任务，提交线程池进行自动调度。
- **数据缓存复用**，在不切换table时，可以反复使用所载入的数据缓存。
- **内存动态管理**，建立缓存容量阈值，如达到上限，则释放部分内存，载入所需数据。

```
// 数据池的基本信息
class PoolBrief {
    String poolPath;
    HashMap<String, String> dirMap;
    int dataSize;
    int numOfCols;
    int[] sepIndex;
}
```

```
// 匹配参数要素集合
class FiltCond {
    String keyWord;
    String operator;
    String optCont;
    String optMode;
    String[] matchPatt;
}
```

总结与展望

- 合理复用已经分配的内存空间，尽量避免过多的中间变量生成。
- 根据不同的应用场景和数据特点开发相应的算法，如模糊查询与精确查询存在很大的不同，正则表达式等方法用于模糊查询并不高效。
- 对于数据湖分析这样的大数据场景，任务调度、索引优化和查询算法都将对整体效率产生重大影响。本方案在索引设计、内存管理、多线程调度等方面还有较大提升空间。

THANKS