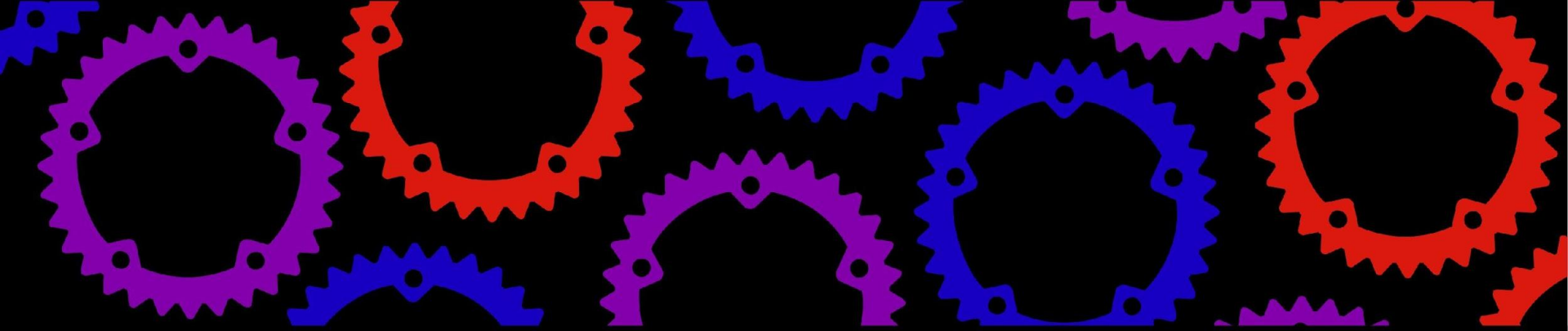


Search Engine in Production with Rust

知乎搜索 孙晓光

2019.04.21

自我介绍

孙晓光，知乎搜索后端负责人

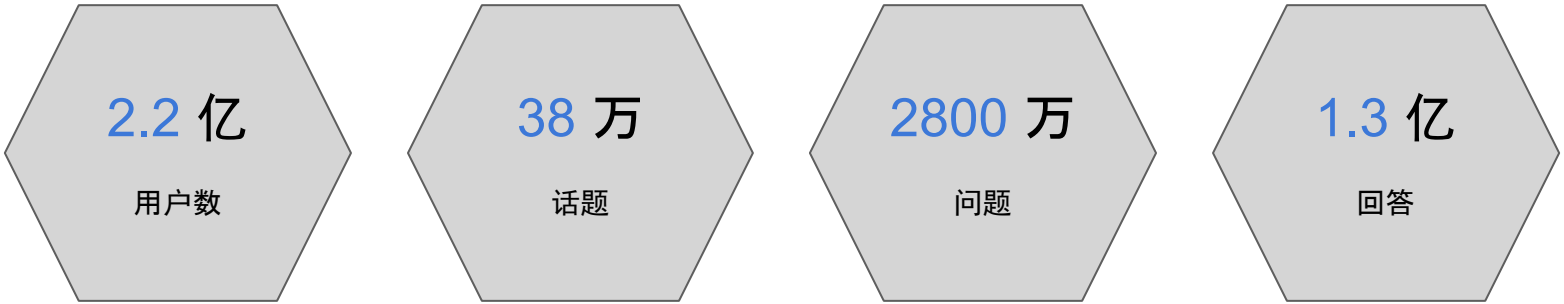
曾从事私有云相关研发，关注云原生技术

TiKV 项目 Committer

目录

- 知乎搜索
- 搜索原理
- 技术选型
- 经验教训

知乎: 可信赖的问答社区

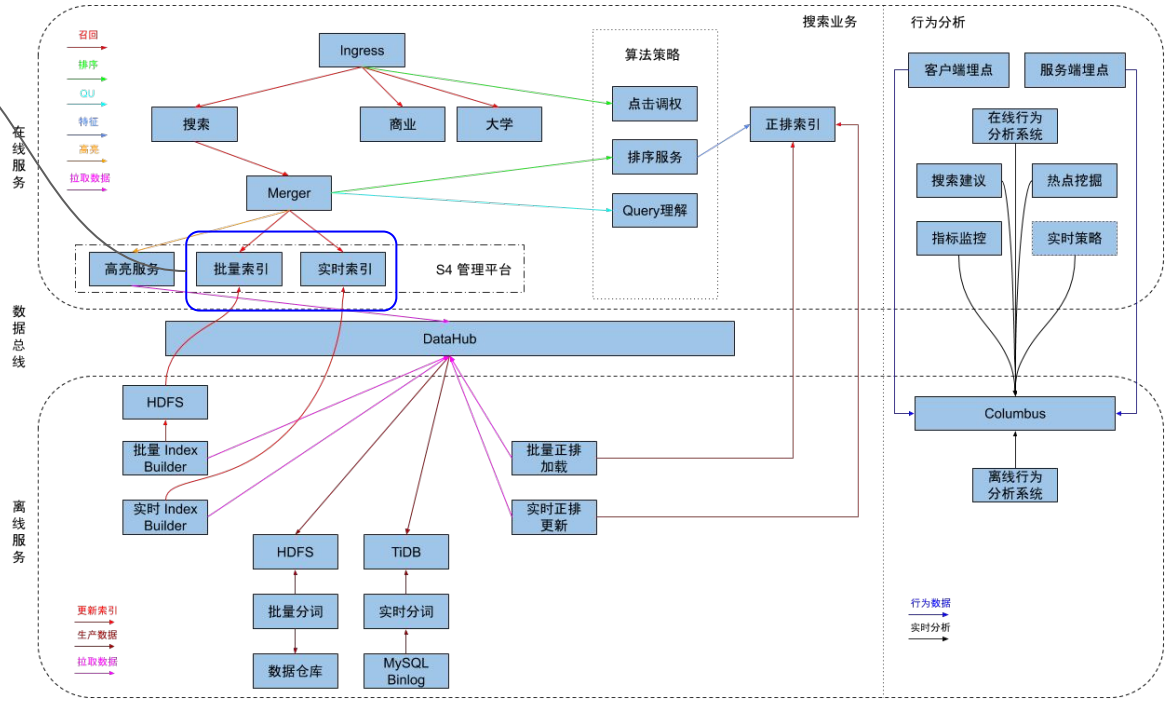


知乎搜索

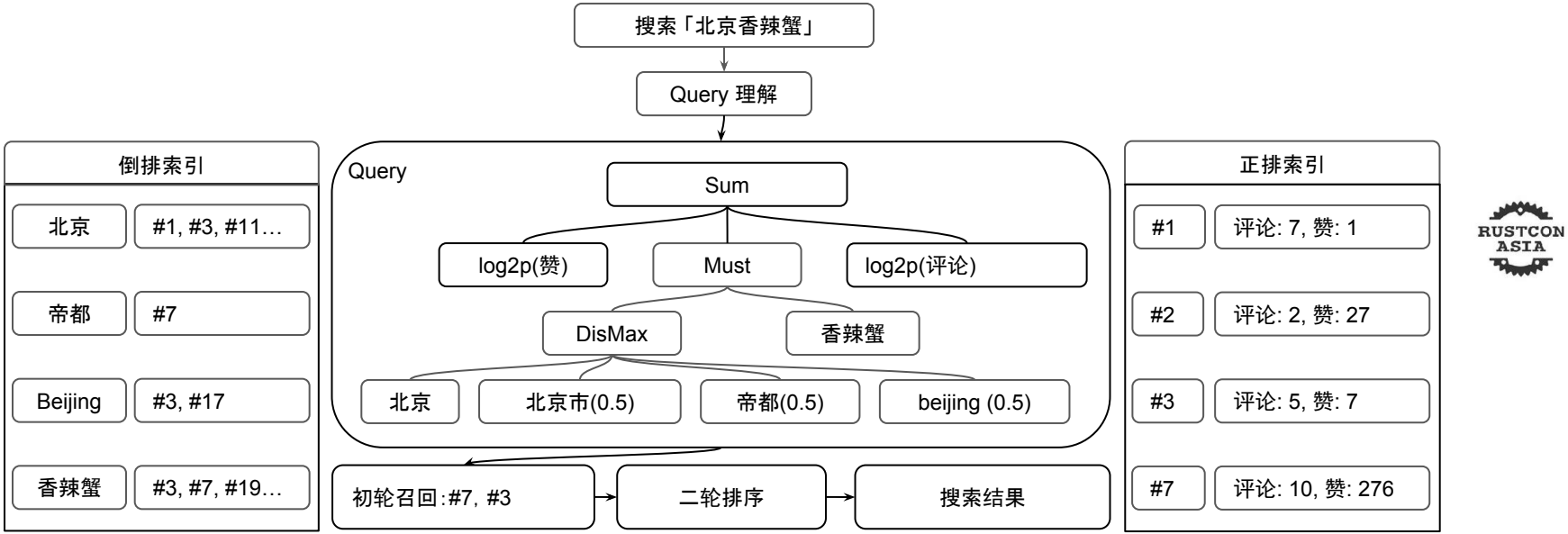
Rucene 引擎

倒排索引
- 召回
- 相关性评分

正排索引
- 文档评分
- 结果分组

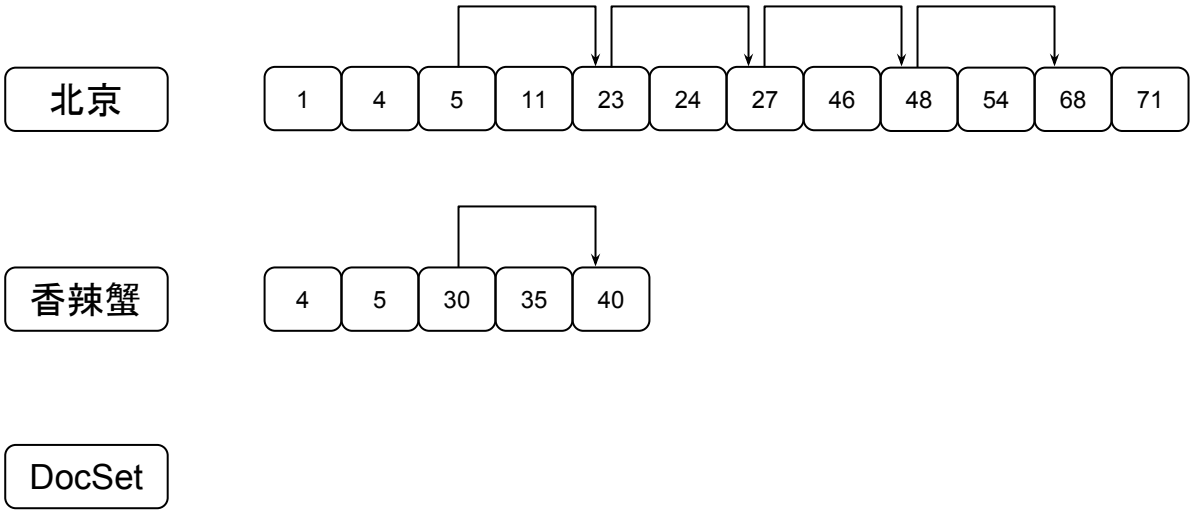


搜索流程



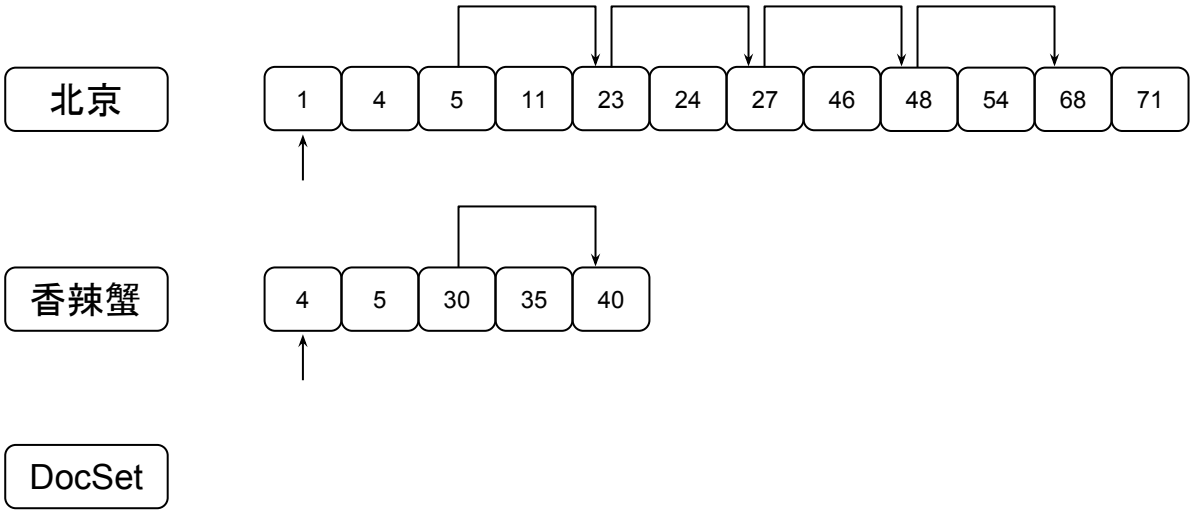
Conjunction

北京 AND 香辣蟹



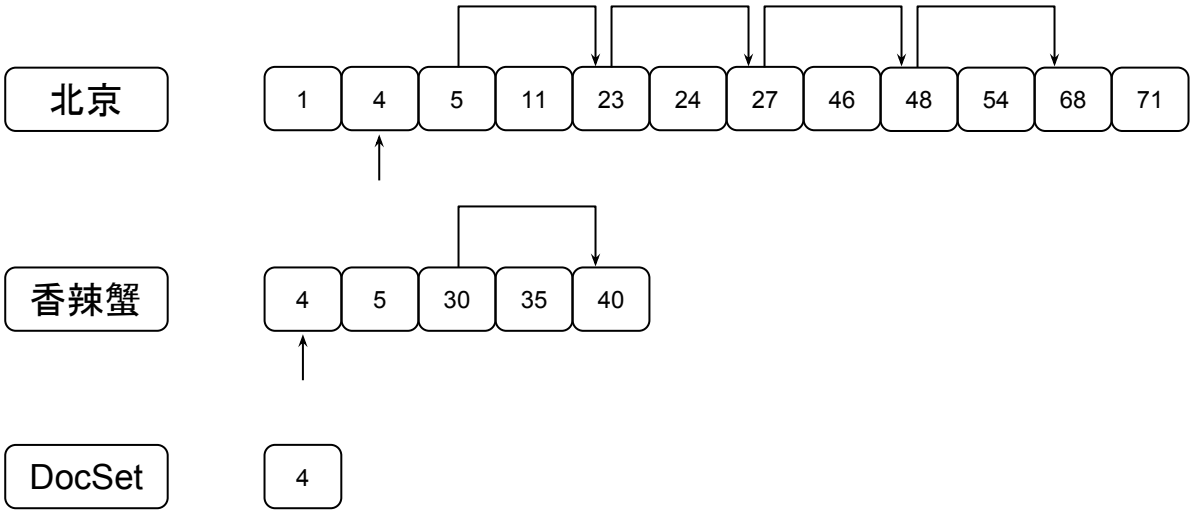
Conjunction

北京 AND 香辣蟹



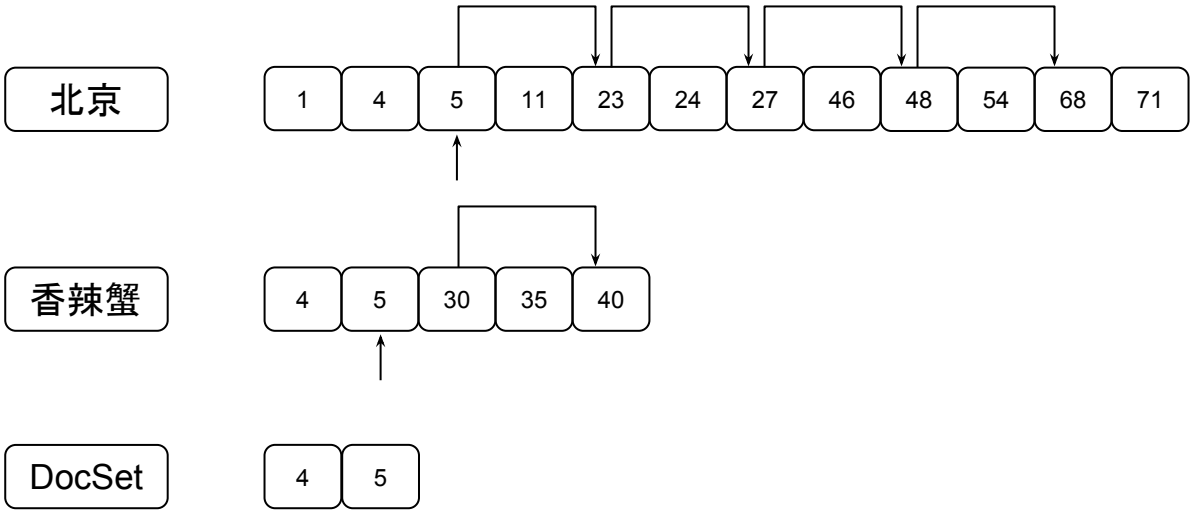
Conjunction

北京 AND 香辣蟹



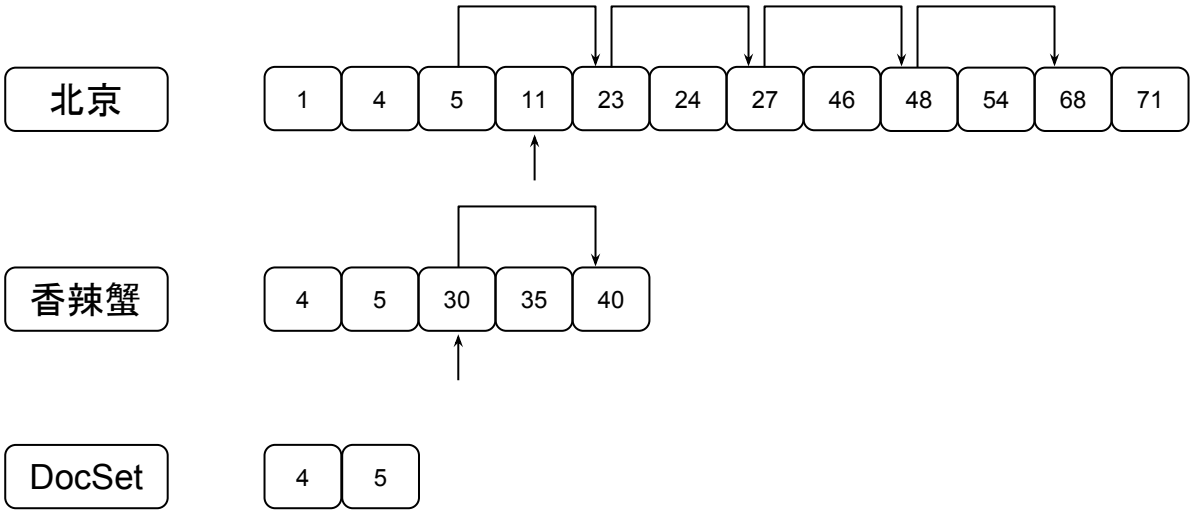
Conjunction

北京 AND 香辣蟹



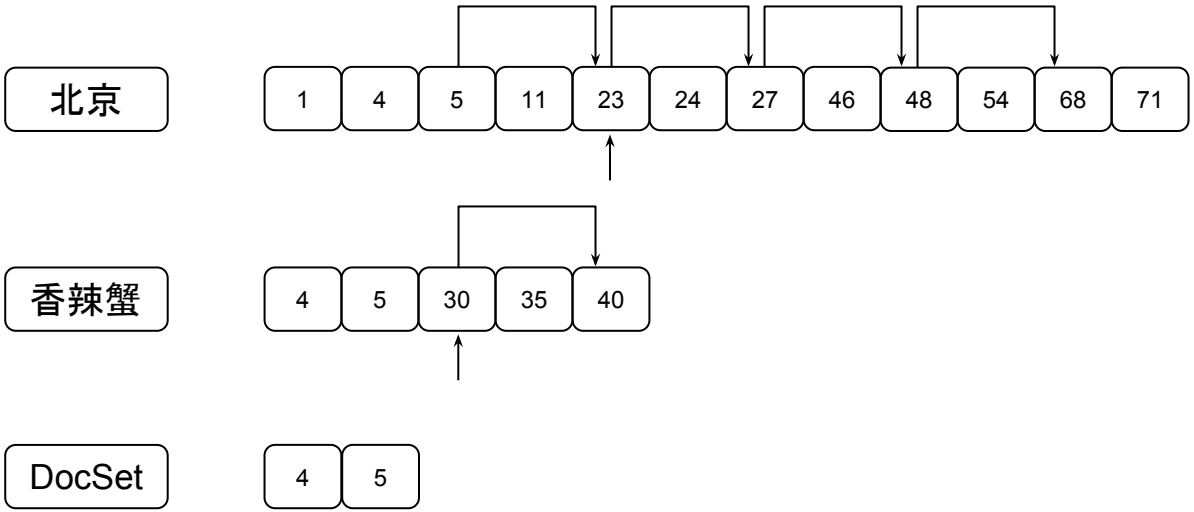
Conjunction

北京 AND 香辣蟹



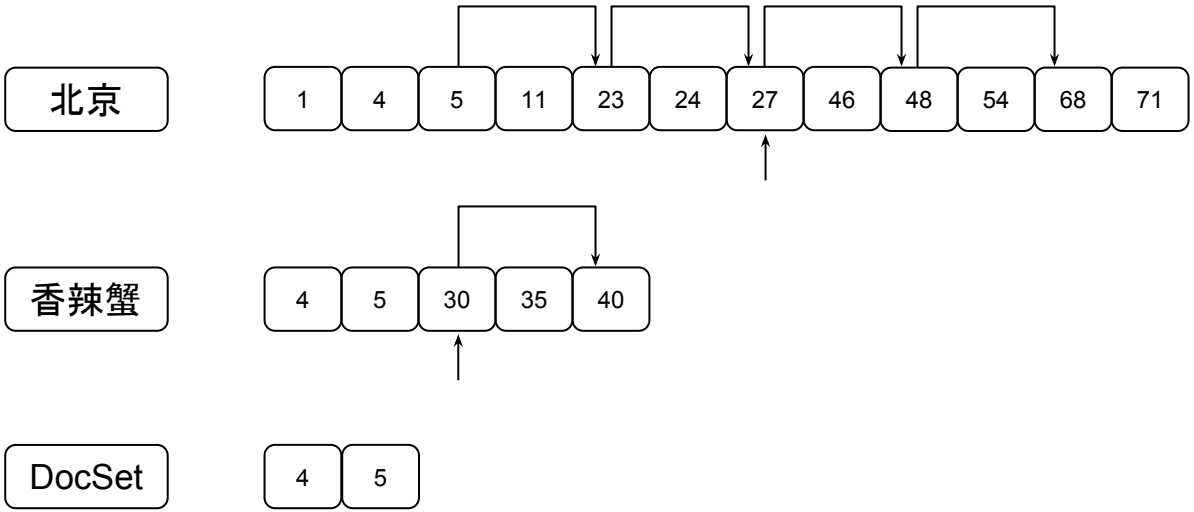
Conjunction

北京 AND 香辣蟹



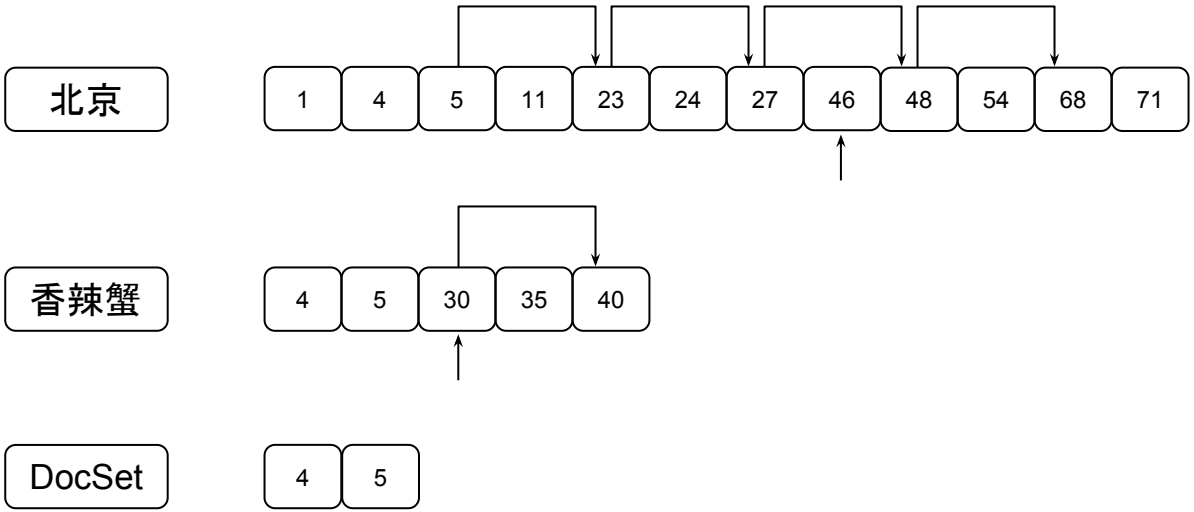
Conjunction

北京 AND 香辣蟹



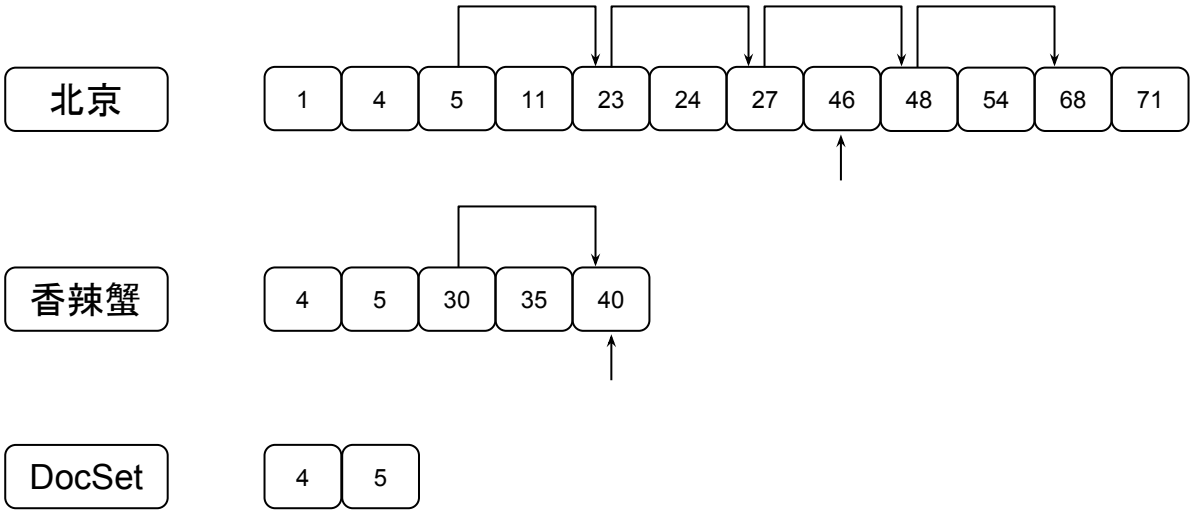
Conjunction

北京 AND 香辣蟹



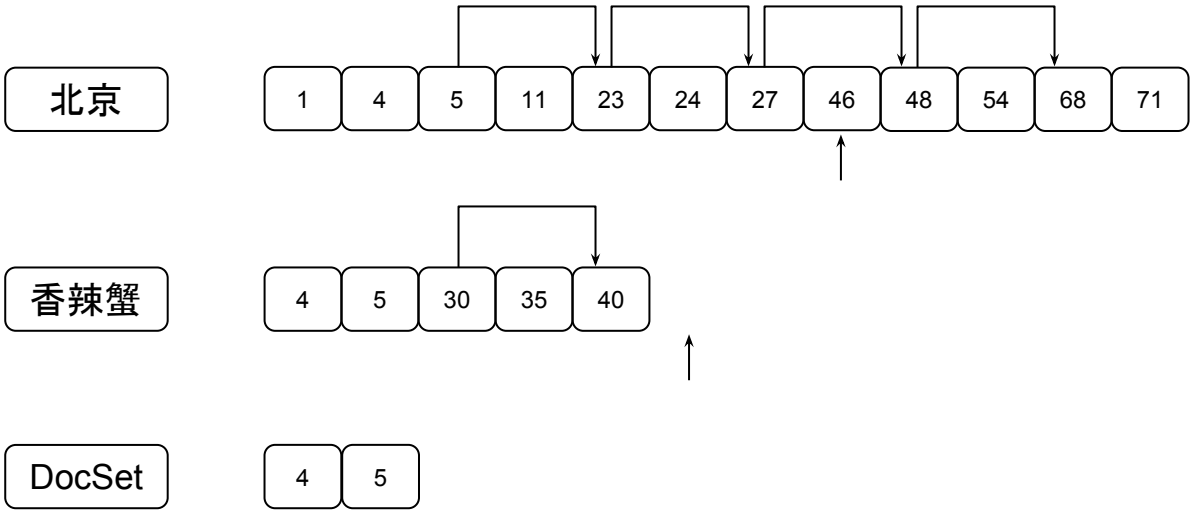
Conjunction

北京 AND 香辣蟹



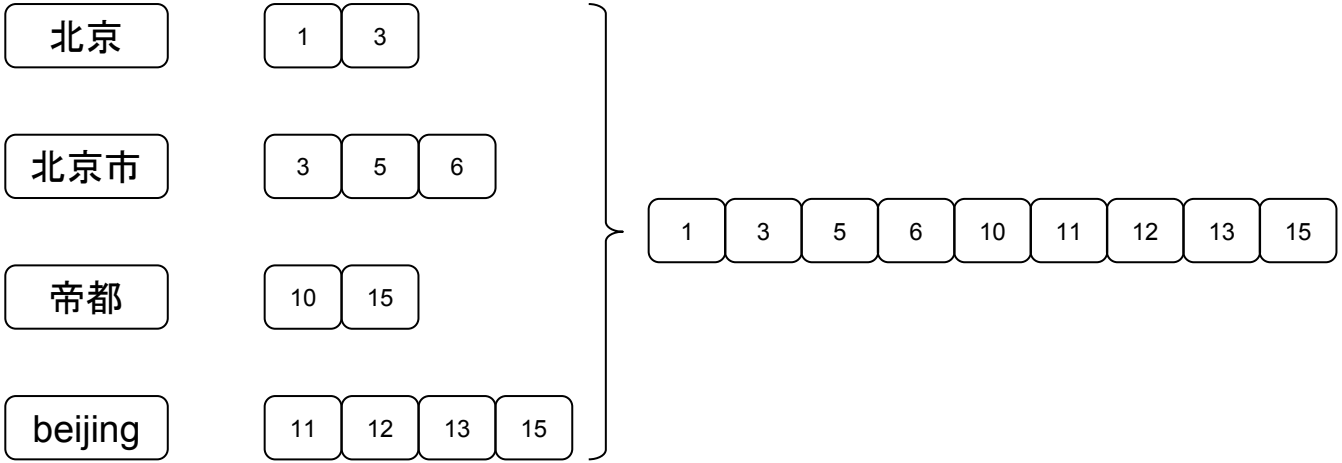
Conjunction

北京 AND 香辣蟹

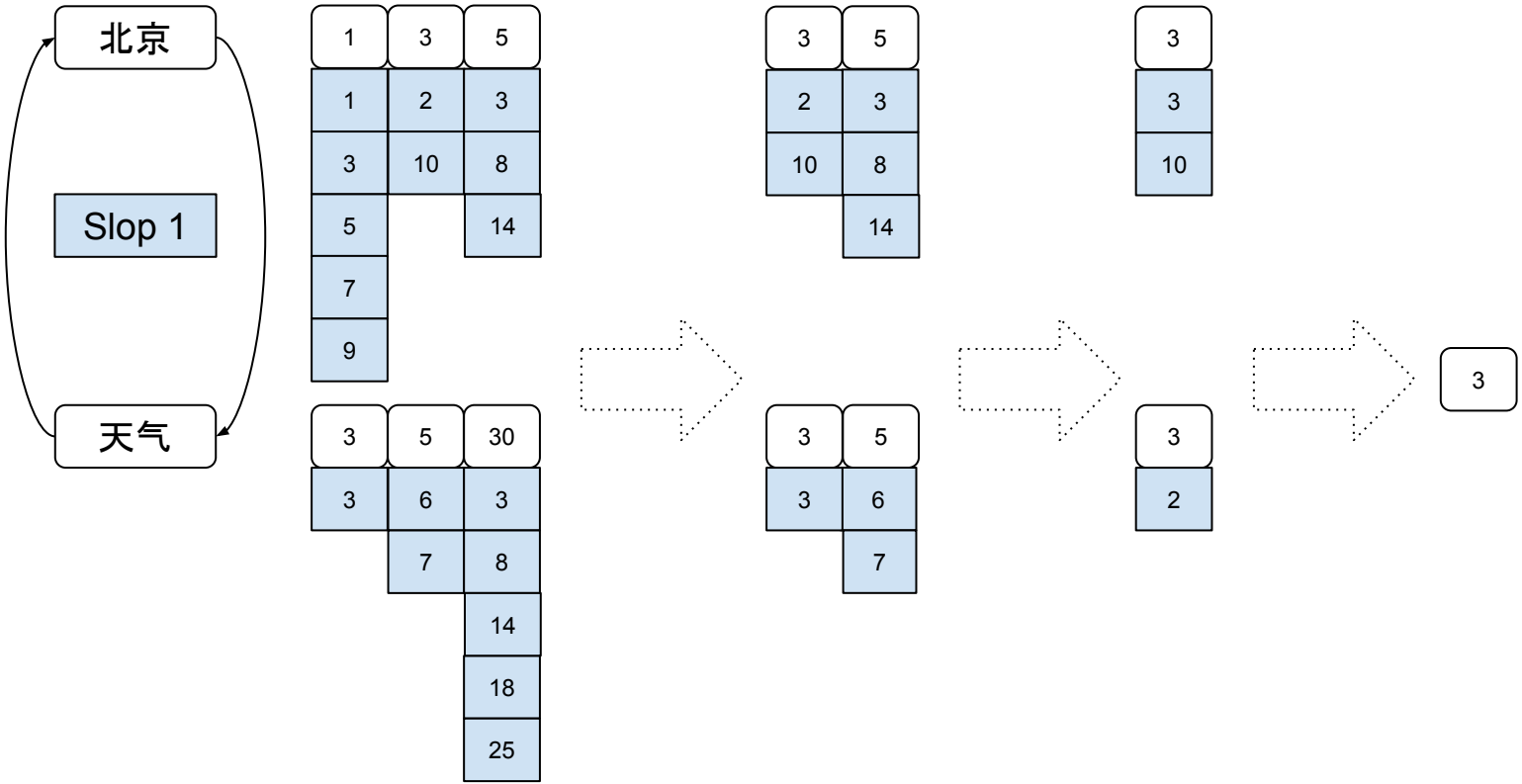


Disjunction

北京 OR 北京市 OR 帝都 OR beijing



Span



相关性打分

$TF_{(t,d)}$: *Term t* 在文档 d 中出现的次数

DF_t: 所有文档中包含 *Term t* 的文档的数量

$$Weight_t: \log_{10}\left(1 + \frac{N - DF_t + 0.5}{DF_t + 0.5}\right)$$

DL_d: 文档 d 的长度

$AVGDL$: 平均文档长度

k_1 b:调节因子

$$BM25_{(Q,d)}: \sum_{i}^{n} Weight_{Q_i} \times \frac{TF_{(t,d)} \times (k_1 + 1)}{TF_{(t,d)} + k_1 \times \left(1 - b + b \times \frac{DL_d}{AVGDL}\right)}$$

Web 搜索引擎特点

IO 密集

计算密集

平台相关性不高

需求相对固定

响应时间敏感

RUSTCON ASIA

技术栈考量

- 期望
  - 无 GC
  - 性能好
  - 表达能力强
  - 成熟的工具链
- 选项
  - C++
  - Rust

C++ vs Rust

- Pros
  - 内存安全
  - 并发安全
- Cons
  - 学习曲线过于陡峭
  - 生态相对不够成熟

经验

- 站在巨人的肩膀上，兼容 Lucene
- 目标合理拆分，读写路径拆分实现
- 努力榨干硬件的每一滴资源
- 更深入的理解，未来更多的可能

教训

- 设计先行
- 用 Rust 的方式思考
- 合理划分系统边界，使用更适合的语言和技术

总结

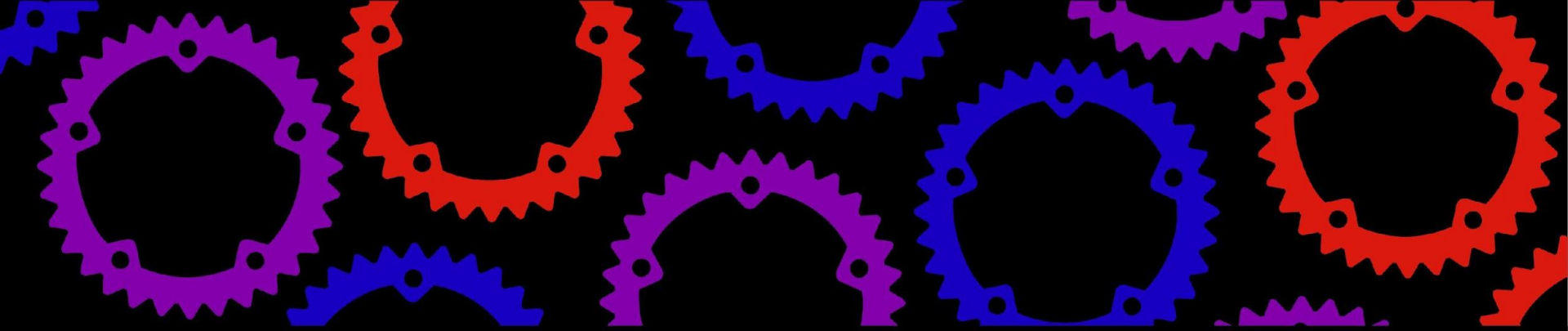
Rust

取舍

信息检索技术

利弊工程密集

知乎搜索



THANKS

github.com/sunxiaoguang zhihu.com/people/solaris sunxiaoguang@gmail.com