Databend 全文索引

Rust生态的搜索服务

tantivy

Tantivy 是rust编写的全文检索 类库，对标 Java生态下的Apache Lucene。

开源的Tantivy 只有单机版本, 用户可以基于这个类库写入日志数据然后进行查询。

代码示例： https://tantivy-search.github.io/examples/basic_search.html

```
    let index_path = TempDir::new()?;
    let mut schema_builder = Schema::builder();
    schema_builder.add_text_field("title", TEXT | STORED);
    let schema = schema_builder.build();
    let index = Index::create_in_dir(&index_path, schema.clone())?;

    let title = schema.get_field("title").unwrap();
    let body = schema.get_field("body").unwrap();

    let mut old_man_doc = Document::default();
    old_man_doc.add_text(title, "The Old Man and the Sea");
    old_man_doc.add_text(
        body,
        "He was an old man who fished alone in a skiff in the Gulf Stream and \
         he had gone eighty-four days now without taking a fish.",
    );

    index_writer.add_document(old_man_doc);
    // For convenience, tantivy also comes with a macro to reduce the boilerplate above.

    index_writer.add_document(doc!(
       title => "Of Mice and Men",
       body => "A few miles south of Soledad, the Salinas River drops in close to the hillside
            bank and runs deep and green. The water is warm too, for it has slipped twinkling
            over the yellow sands in the sunlight before reaching the narrow pool. On one \
            side of the river the golden foothill slopes curve up to the strong and rocky \
            Gabilan Mountains, but on the valley side the water is lined with trees—willows \
            fresh and green with every spring, carrying in their lower leaf junctures the \
            debris of the winter's flooding; and sycamores with mottled, white, recumbent \
            limbs and branches that arch over the pool"
    ));

    index_writer.commit()?;

    // start read
```

tantivy 是一个lib，他提供了基本的 日志 Store 和 query功能，用户可以基于它构建一个基本的搜索服务。

其中store模块包含：

1. Index ，索引存储的目录，目前只支持fs 方式（利用了 MmapDirectory）
2. Segment管理
3. 后台的compaction线程 等

由于tantivy 目前Index只服务fs， 对于基于云原生的databend来说，直接集成tantivy 可行性不大。

tantivy团队也意识到了这一点，因此他们开发了基于云原生的分布式日志引擎 quickwit， 它是一个完整的产品，和databend 功能相对平行。

方案

最近思考了下Databend 中支持全文索引需求，目前有以下两种方案

构建服务日志搜索的engine

实现一个 LogEngine，类似FuseEngine，专门服务于日志场景，其中storage format 部分我们可以参考tantivy的代码，工作量类似将tantivy 代码fork出来，保留核心的索引部分， 然后集成到我们自己的segment管理，索引管理， compaction框架中。

这个方案改造量较大，但优点是相对隔离，不会入侵到fuse engine的设计中

FuseEngine 支持二级索引

在FuseEngine的基础上，我们支持二级索引的框架，用户可以通过SQL ALTER 对某个字段加入索引，类似ClickHouse的二级索引。

其中ClickHouse的全文索引功能示例如下：

`ngrambf_v1(n, size_of_bloom_filter_in_bytes, number_of_hash_functions, random_seed)`

Stores a Bloom filter that contains all ngrams from a block of data. Works only with datatypes: String, FixedString and Map. Can be used for optimization of `EQUALS`, `LIKE` and `IN` expressions.

Snowflake 也有 类似的二级索引，它暴露了更少的信息，相对clickhouse没有那么 灵活，但对于用户来说，减少了复杂度。

```
create or replace table test_table (id int, c1 int, c2 string, c3 date) as select * from v
  (1, 3, '4',  '1985-05-11'),
  (2, 4, '3',  '1996-12-20'),
  (3, 2, '1',  '1974-02-03'),
  (4, 1, '2',  '2004-03-09'),
  (5, null, null,  null);

  alter table test_table add search optimization;

  // the following query can use this optimization
  select * from test_table where id = 2;
  select * from test_table where c3 = '1985-05-11';
```

值得注意的是，snowflake支持的是全列索引，类似阿里的 AnalyticDB ， 他的作用是在表的每个列都加上索引 ，服务于 面向表某个列的点查 场景加速，并不会有日志模糊搜索的优化效果。

这个方案改造不大，parquet虽然有二级索引的支持，但databend 可以自己维护索引的结构格式 BlockMeta， 相对来说会更加灵活。

但有很明显的几个缺点：

1. 全文二级索引 相比MinMax 索引会更加占用内存大小，如果我们云平台后续是 基于小内存的机器粒度 （4c 8G) 来对用户的集群进行扩容，查询节点很可能出现OOM （若限制cache的容量，则会导致查询性能下降）。
2. Query 在查询fuse的时候，需要将filter 符合二级索引下推到fuse engine中 对 block partition进行 prune, 二级索引不一定能100%提速， 这个prune的效率取决于数据的分布情况 和 查询like条件，极端情况下二级索引性能聊胜于无。
3. 二级索引会导致数据插入性能下降。

目前看起来二级索引方案会更符合我们的演进

其他问题：

1. 我们的Goal是 云原生数仓， 全文索引是否是我们短期的目标？或是未来的目标？ 目前看来snowflake 也没有涉及这个领域
2. 二级索引功能能否让实习生或社区的学生 作为一个Topic 来做？

Cc @张雁飞 @吴炳锡 @赵博然 @张健 @白珅