解决方案文档

1. 系统设计

1.1 技术栈

后端：flask

前端：VUE、AXIOS、ELEMENTUI、jsplumb

2. 算法模型设计

2.1 第三题

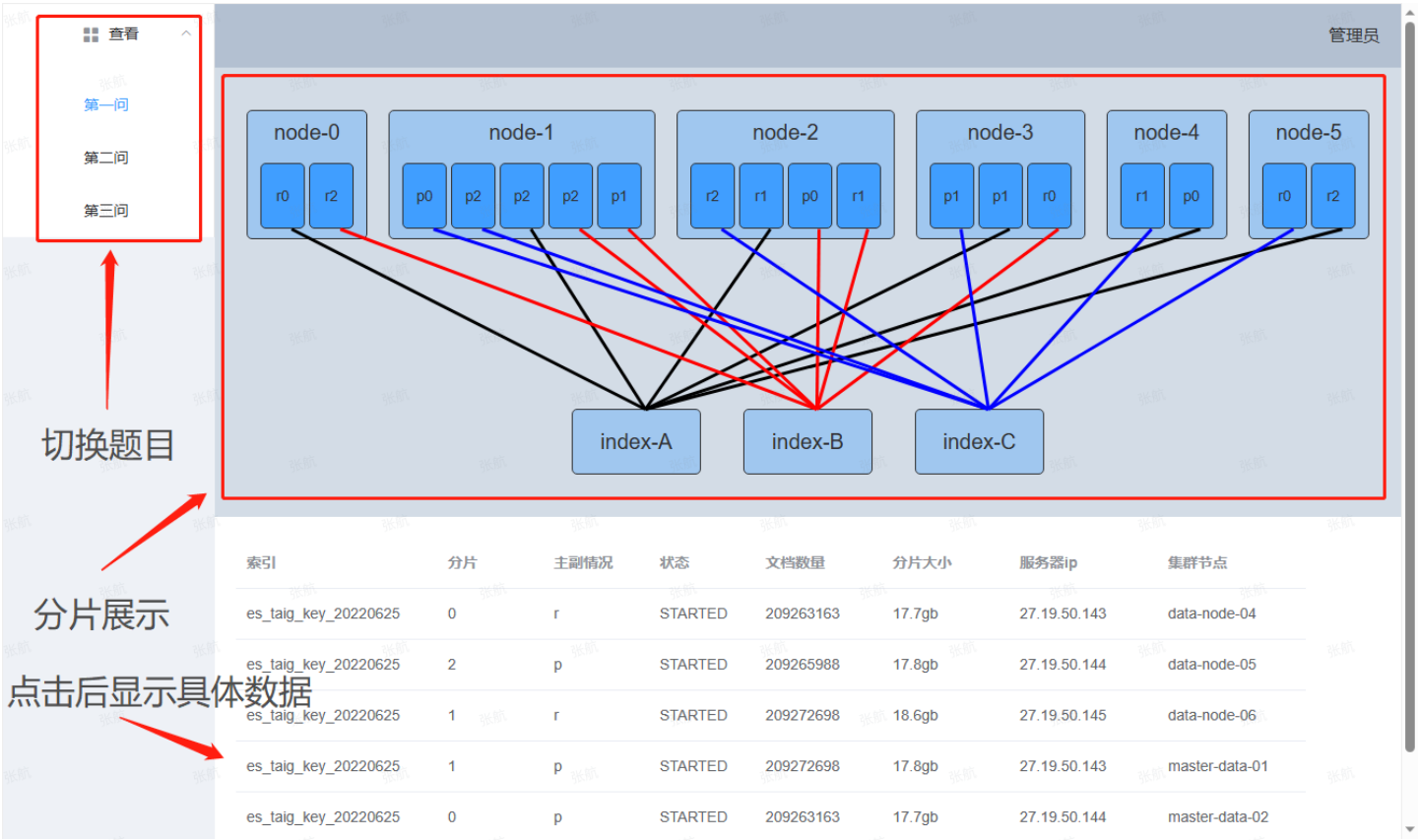
首先对于本题来说（4台物理机，7个节点，3个索引每个索引有6个分片）如果考虑在没有进行分配的情况下如何进行最优

3. 使用说明

前端地址：http://localhost:8080

后端地址：http://127.0.0.1:5000

具体请见下图：



4. 接口设计

4.1 前后端交互接口设计

```python
# 前端数据格式:
# b'', 需要先转为字符串，再json解码之后转为字典
# 具体如下例
# params: {
#             key: value
#         }
# 后端回传数据格式:
# 列表中字典
# [{key: value}, {}]


@app.route('/get_q_first_list', methods=['GET', 'POST'])
    """
    get_q_first_list: 获取1、2题的基本数据
    无参数
    return: 数据list, 以index升序排列
    """


@app.route('/get_q_first_shard_node', methods=['GET', 'POST'])
    """
```

```
    get_q_first_shard_node: 根据shard_node值, 返回对应shard_node结点数据
    name: 结点名字, 值为index&prirep&shard
    params: {
                name: index&prirep&shard
            }
    return: get_q_first_shard_node结点对象的列表(此处应只有一个节点对象)
    """


@app.route('/get_q_first_index_node', methods=['GET', 'POST'])
    """
    get_q_first_index_node: 根据index, 返回对应index_node结点数据
    name: index
    params: {
                name: index
            }
    return: get_q_first_index_node结点对象的列表
    """


@app.route('/get_q_second_connect_list', methods=['GET', 'POST'])
    """
    [
        ["es_taig_key_20220624&r&0", "data-node-05"],
        ["es_taig_key_20220624&p&0", "data-node-06"],
    ]
    get_q_second_connect_list: 返回连线, index = 0存起点index&prirep&shard, index
    return: 连线的列表
    """


@app.route('/get_q_third_list', methods=['GET', 'POST'])
    """
    get_q_third_list: index_node结点数据, 以physics升序排列
    return: 以physics升序排列结点对象的列表, 如物理机上无虚拟机, 虚拟机数据请用字符串'Nc
    """


@app.route('/get_q_physics_node', methods=['GET', 'POST'])
    """
    get_q_physics_node: 根据physics, 返回对应physics结点数据
    params: {
                name: physics
            }
    return: physics_node结点对象的列表
    """
```

4.2 数据库交互接口设计

5. 部署说明

5.1 仓库地址

https://git.code.tencent.com/zhanghang6/Elasticsearch_module

5.2 安装依赖

5.2.1 前端

1. **安装node.js(版本应为：v12.17.0，可参考下面的文章安装node版本管理调换版本)**

   参考文章：安装nodejs

2. **安装前端依赖（进入前端目录/template）**

```
npm install
```

5.2.2 后端

1. **创建一个虚拟环境（已创建请忽略此步，由于本项目虚拟环境不能通用，请自行进入后端目录/api后，删除原虚拟环境/venv后进行如下配置）**

在Windows下：

```
$ py -3 -m venv venv
```

2. **激活虚拟环境（需要在后端路径下/api）**

```
> venv\Scripts\activate
```

3. **安装后端依赖（已安装，请忽略此步）**

```
$ pip install Flask
$ pip install -U flask-cors
$ pip install openpyxl
$ pip install pandas
```

5.3 运行步骤

5.3.1 前端

a. 进入前端目录 /template

b. 安装依赖 npm install

c. 启动服务 npm run dev

5.3.2 后端

a. 进入后端目录 /api

b. 进入虚拟环境 venv\Scripts\activate

c. 启动服务 flask run

6. 当前问题以及后期优化

6.1 参考文档

jsplumb 中文基础教程

element-ui中文文档

Vue+Flask实战开发

浏览器发器POST请求

6.2 前端存在的问题

- dom之间的连线不够精准美观，需要调整样式
- 连线的生成时间复杂度为O(n3)，需要优化生成逻辑
- 处理逻辑需要改进，若索引无数据则会出错

6.3 后端存在问题