大数据基于 spark 的杭州市空气质量发布平台

目录

一、项目背景 3
1.1 项目的提出原因 3
1.2 项目的环境背景 3
1.3 项目运作的可行性 3
1.4 项目的优势分析 3
二、项目技术栈 4
三、项目功能描述 4
3.1 用户认证与账户管理： 4
3.2 管理员控制面板： 4
3.3 数据概览主页： 4
3.4 时段查询分析页面： 5
3.5 日期查询分析页面： 5
四、系统流程图 5
4.1 数据源： 6
4.2 数据采集及预处理： 6
4.3 统一数据存储与计算 Spark HDFS： 6
4.4 连接数据库， 6
4.5 后端服务： 6
4.6 前端服务： 6
4.7 Web 网页： 6
五、数据来源 7
5.1 空气质量统计数据 7
5.2 用户信息统计数据 7
六、计算结果数据设计 8
6.1 清洗数据 8
6.1.1 清洗数据的代码 8
6.1.2 处理好的数据生成 jar 包后传至集群 9

6.2 数据设计 13
七、页面展示 13
7.1 选择登入界面 13
7.1.1 初始界面 13
7.1.2 用户注册界面 14
7.1.3 用户登录界面 15
7.1.4 管理员界面 15
7.2 统计空气质量 16
7.2.1 单日查询 16
7.2.2 时段查询 17
7.2.3 统计各污染物热力 18

一、项目背景

1.1 项目的提出原因

随着城市化进程的加速和工业活动的增加，空气质量问题已成为影响城市居民生活质量的重要因素。基于此现状，本项目决定建立一个基于 Spark 的空气质量发布平台。该平台旨在准确地统计和判断空气质量信息，为政府决策、企业运营和公众生活提供科学依据，从而提升城市环境管理水平和居民生活质量。

1.2 项目的环境背景

当前，杭州市面临着日益严峻的空气质量挑战，包括 PM2.5、PM10、SO2、CO、NO2、O3 等多种污染物的浓度监测和数据分析需求。这些数据来源于气象局、环保局等官方网站。因此，需要一个高效、可靠的数据处理和发布系统来应对这一挑战。

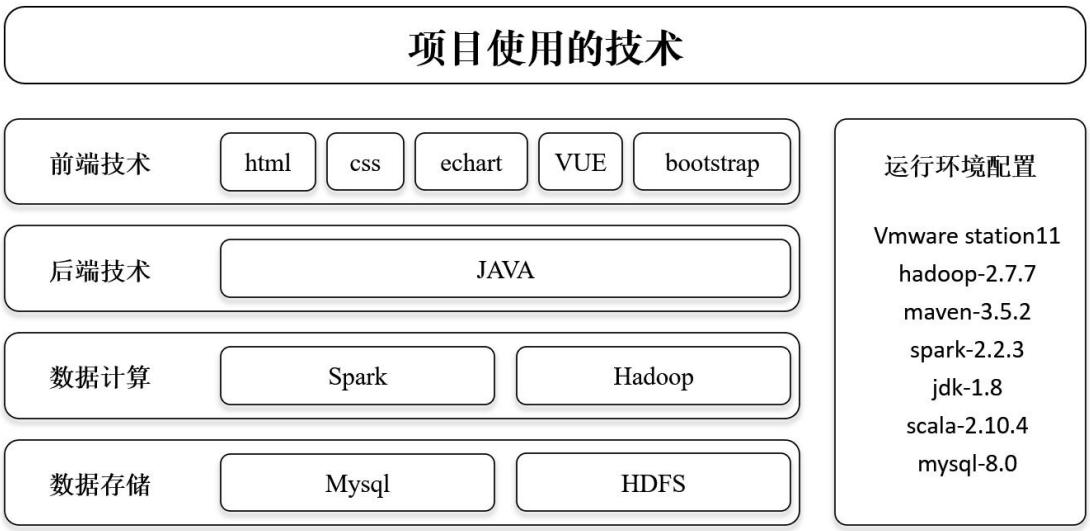
1.3 项目运作的可行性

本项目从杭州市官方网站抓取历史空气质量数据，包括 PM2.5、PM10、SO2、CO、NO2、O3 等污染物浓度。在集群环境中运行，将原始文件上传节点和 HDFS 中，利用 scala 进行数据清洗，并存储到集群节点中的 mysql 中。使用 Spark 的分布式计算框架，可以对海量数据进行清洗、转换和标准化处理，确保数据的一致性和可用性。这种架构设计保证了数据处理的高效性和系统的可扩展性。

1.4 项目的优势分析

该项目的主要优势在于其高效的数据处理能力和权威的数据来源平台。通过 Spark 的分布式计算框架，可以快速处理大量数据，确保数据的准确性。此外，系统的架构设计支持弹性扩展，能够根据数据量的增长进行相应的调整，保证系统的稳定性和可用性。这不仅有助于政府和企业做出更为科学的决策，也能让公众及时了解空气质量信息，从而采取相应的防护措施。

二、项目技术栈



三、项目功能描述

3.1 用户认证与账户管理：

新用户在注册过程中需填写必要的个人信息，并可选择上传个性化头像或使用系统提供的默认头像。已注册用户可通过验证身份信息直接登录系统，而未经验证身份的用户将无法访问平台的特定内容页面，确保平台内容的安全性和专业性。

3.2 管理员控制面板：

管理员拥有查看所有已注册用户详细信息的权限，并能监控平台上的各项数据指标，这有助于确保平台运行的透明度和数据的安全性，同时也便于管理员对平台内容和用户行为进行有效的管理和监督。

3.3 数据概览主页：

该页面提供杭州市的人口统计、地理位置信息，以及主要污染物的组成和空气质量基础数据，为用户提供一个清晰的城市空气污染概览。这种全面的数据展示有助于用户快速了解城市空气质量的整体状况，为决策提供数据支持。

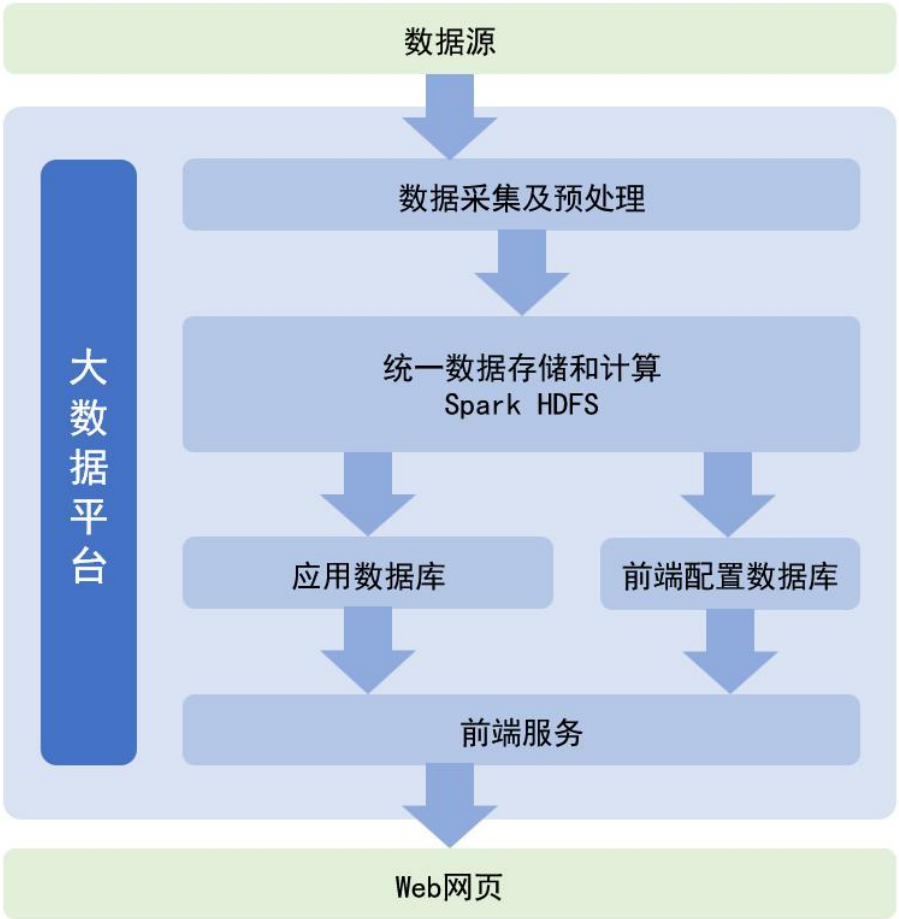
3.4 时段查询分析页面：

用户可在此页面选择特定时间段，通过折线图、玫瑰图和热力图等形式，深入分析该时段内的空气质量变化，并可针对不同污染物进行详细的数据分布查看。这种多维度的数据分析工具帮助用户更深入地理解空气质量的变化趋势和污染源的分布情况。

3.5 日期查询分析页面：

此页面允许用户选择特定日期，以柱状图、仪表盘、折线图和扇形图等多种形式，展示该日 24 小时内的污染物浓度变化，同时提供该日的 AQI 平均指数及 24 小时 AQI 趋势分析。这种详细的数据展示和分析工具使用户能够全面了解特定日期的空气质量状况，为环境管理和个人健康防护提供科学依据。

四、系统流程图



4.1 数据源：

本项目的数据来源于杭州市的官方数据开放平台，数据规模宏大，约十万条记录，来源可靠，覆盖全面，种类丰富，为后续的数据处理和分析奠定了坚实的基础。

4.2 数据采集及预处理：

通过日期格式的标准化、数据的过滤与清洗、创建 DataFrame 以及定义 Schema 等步骤，对数据进行深度整理与格式化。最终，将处理好的数据妥善保存至 MySQL 数据库，为后续的大数据处理流程做好准备。

4.3 统一数据存储与计算 Spark HDFS：

将本地采集到的数据存储于 Hadoop 集群的 HDFS 中。

4.4 连接数据库，

其中用户信息在本地数据库储存，大数据在主节点处储存，按照所需要求查询在数据库中获取相应数据。前端服务从后端数据库中读取数据，通过一系列辅助计算函数对数据进行分组、统计、平均等处理，然后将处理后的数据传递给 echarts 等可视化工具进行渲染，为用户提供直观的数据视图。

4.5 后端服务：

后端负责接收用户的查询请求，迅速从数据库中检索出相关数据，封装后返回给前端。

4.6 前端服务：

前端计算和处理后端获取到的数据，使用这些数据渲染 echarts 图呈现在网页上。

4.7 Web 网页：

最终，所有处理和分析的数据成果都以 Web 网页形式呈现给用户。用户通过浏览器即可访问，查看数据、分析结果或进行交互操作。

综上所述，本大数据系统流程实现了从数据源到 Web 网页的完整数据处理链，每个环节都经过精心设计与优化，确保了数据处理的准确性、高效性和用户体验的卓越性。

五、数据来源

5.1 空气质量统计数据

此数据来源于杭州市官方网站，为我们的平台空气质量的统计和分析提供可靠的数据支持：

站点编号	站点名称	date	timeid	PM1	PM25	PM10	CO	NO2	SO2	O3
KQ005	候车厅22B北	2021-09-23	35.0	4.802	5.606	5.694	104	151	132	6
KQ001	南2出租车候	2021-09-23	35.0	3.588	4.058	4.069	1072	27	209	115
KQ002	北2出租车候	2021-09-23	35.0	2.331	3.282	3.381	1245	265	56	97
KQ004	P4停车场	2021-09-23	35.0	0.741	3.906	0000005	3985	89	228	26
KQ003	P6停车场	2021-09-23	35.0	0.512	0.52	0.52	2337	123	185	17
KQ004	P4停车场	2021-09-23	36.0	0.959	4.774	7.699	3809	117	217	38
KQ002	北2出租车候	2021-09-23	36.0	1.963	2.307	2.318	1613	274	80	103
KQ003	P6停车场	2021-09-23	36.0	0.642	1.429	1.769	2349	121	165	8
KQ005	候车厅22B北	2021-09-23	36.0	5.374	6.12	6.211	100	150	133	12
KQ001	南2出租车候	2021-09-23	36.0	4.272	7.154	0000001	499	65	185	85
KQ004	P4停车场	2021-09-23	37.0	1.179	5.572	8.774	4396	92	231	33
KQ002	北2出租车候	2021-09-23	37.0	2.259	10.491	60.808	1262	240	26	110
KQ003	P6停车场	2021-09-23	37.0	0.654	0.696	0.696	2420	115	177	17
KQ001	南2出租车候	2021-09-23	37.0	3.945	4.378	9999999	1757	5	237	117

5.2 用户信息统计数据

便于实现管理员界面实行查询的功能需要，为维护平台秩序提供数据支持：

id	username	password	phone	header
1	zhangsn	123	15602025	(Null)
2	zhangsn	111	15602025	(Null)
12	zhangsn	22	15602025	zhangsn.jp
14	1	1	15602025	1.jpg
15	admin	666	15602025	1.jpg
17	小当家	22	15602025	小当家.jpg

六、计算结果数据设计

6.1 清洗数据

6.1.1 清洗数据的代码

先将获取的数据进行清洗，这里展示了 scala 程序的主要代码：

例如修改日期的格式为“yyyymmdd”，删除有空值字段的数据条等

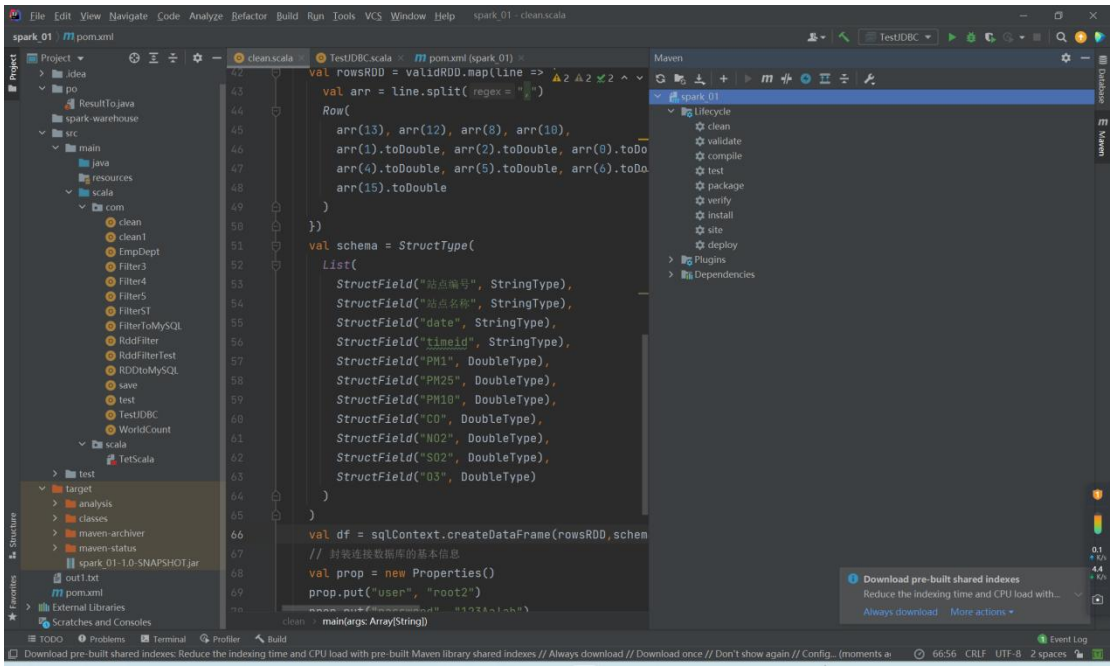
```scala
object clean {
  def main(args: Array[String]): Unit = {
    // 初始化 SparkSession
    val sc = new SparkContext(new SparkConf().setMaster("local").setAppName("CleanAndSaveToDB"))
    val sqlContext = new SQLContext(sc)
    val lineRDD = sc.textFile( path = "/yuanshi.txt")

    val outputFilePath = "/first"

    // 定义日期格式
    val inputDateFormat = "yyyy-MM-dd"
    val outputDateFormat = "yyyyMMdd"

    // 处理数据
    val filteredRDD = lineRDD.map { line =>
      val columns = line.split( regex = ",")  // 假设列是以逗号分隔
      val dateIndex = columns.indexWhere(col => col.matches( regex = "\\d{4}-\\d{2}-\\d{2}"))  // 找到日期列的位置
      if (dateIndex != -1) {
        val oldDate = java.time.LocalDate.parse(columns(dateIndex), java.time.format.DateTimeFormatter.ofPattern(inputDateForm
        val newDate = oldDate.format(java.time.format.DateTimeFormatter.ofPattern(outputDateFormat))
        columns(dateIndex) = newDate
      }
      columns.mkString(",")  // 将处理后的行重新拼接成字符串
    }

    // 将处理后的数据保存为文本文件
    filteredRDD.repartition( numPartitions = 1).saveAsTextFile(outputFilePath)
```

将数据处理为便于处理 String 或 Double 的格式

并且我们需要定义 DateFrame 的 scheme 格式，定义好要放入数据库中表的各个字段。

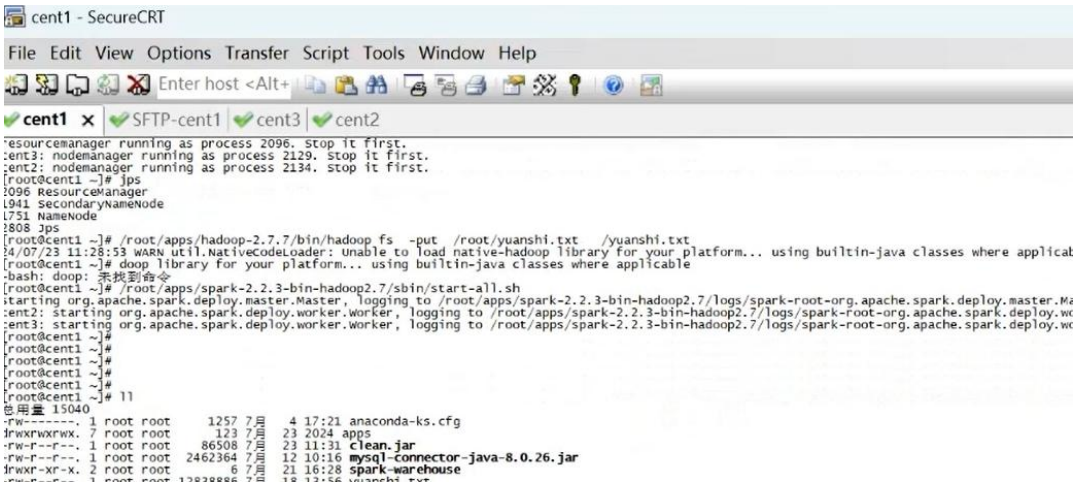
```scala
// 将处理后的数据保存为文本文件
filteredRDD.repartition( numPartitions = 1).saveAsTextFile(outputFilePath)

// 从处理后的 RDD 创建 DataFrame
val dataRDD = filteredRDD.filter(line => !line.startsWith("PM10采集值"))
val validRDD = dataRDD.filter(line => {
  val arr = line.split( regex = ",")
  arr.length >= 16
})
val rowsRDD = validRDD.map(line => {
  val arr = line.split( regex = ",")
  Row(
    arr(13), arr(12), arr(8), arr(10),
    arr(1).toDouble, arr(2).toDouble, arr(0).toDouble,
    arr(4).toDouble, arr(5).toDouble, arr(6).toDouble,
    arr(15).toDouble
  )
})
val schema = StructType(
  List(
    StructField("站点编号", StringType),
    StructField("站点名称", StringType),
    StructField("date", StringType),
    StructField("timeid", StringType),
    StructField("PM1", DoubleType),
    StructField("PM25", DoubleType),
    StructField("PM10", DoubleType),
    StructField("CO", DoubleType),
```

此外，为了在集群上运行，我们需要将代码通过 maven 打包上传到主节点。



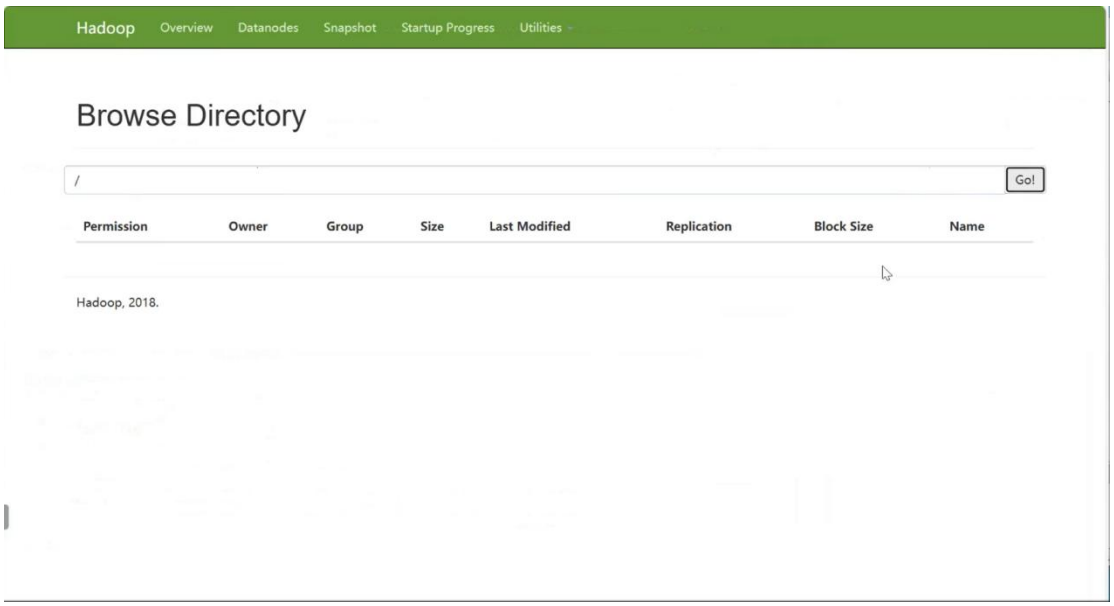
6.1.2 向节点上传原始数据并运行 jar 包

先将原始数据上传到主节点中：



再将原始数据上传到 HDFS 中：

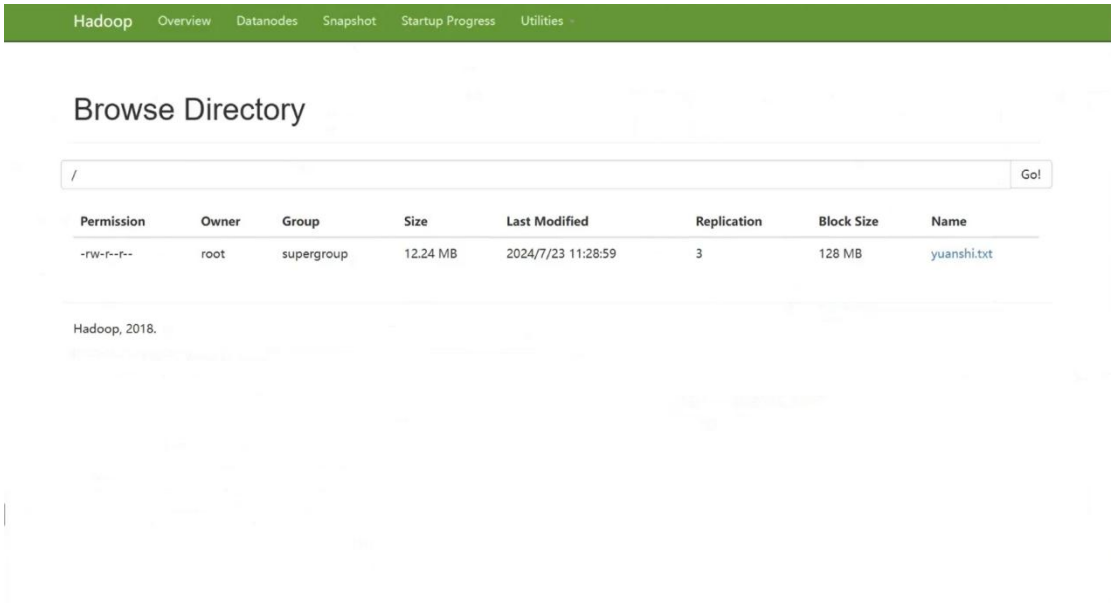
传输数据之前，hadoop 上没有相关文件



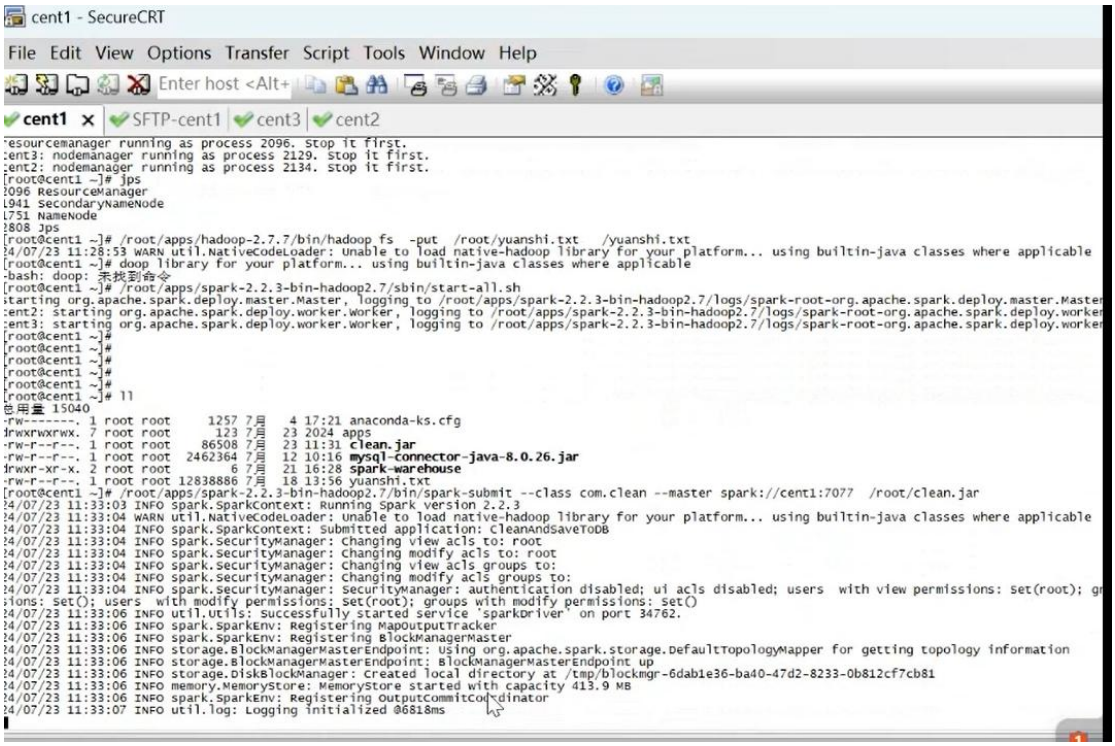
经以下指令传输：

/root/apps/hadoop-2.7.7/bin/hadoop fs -put /root/yuanshi.txt /yuanshi.txt

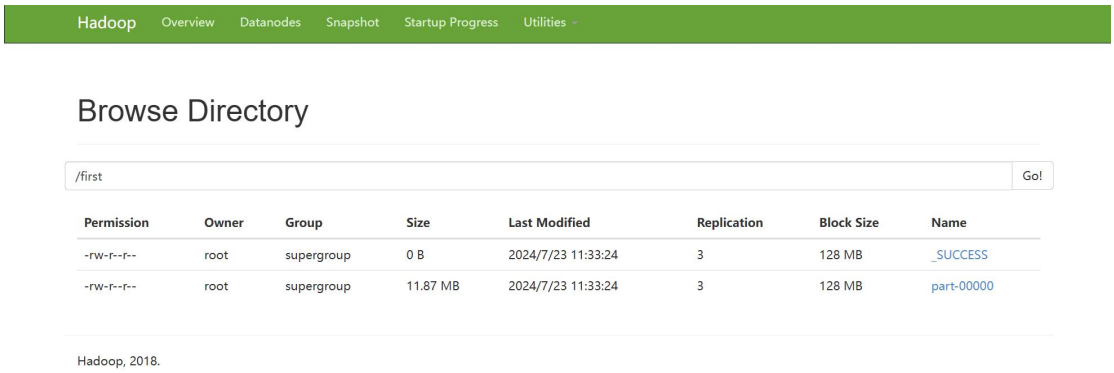
传输后



在集群上运行 jar 包对原始数据进行清晰处理并保存到数据库中：

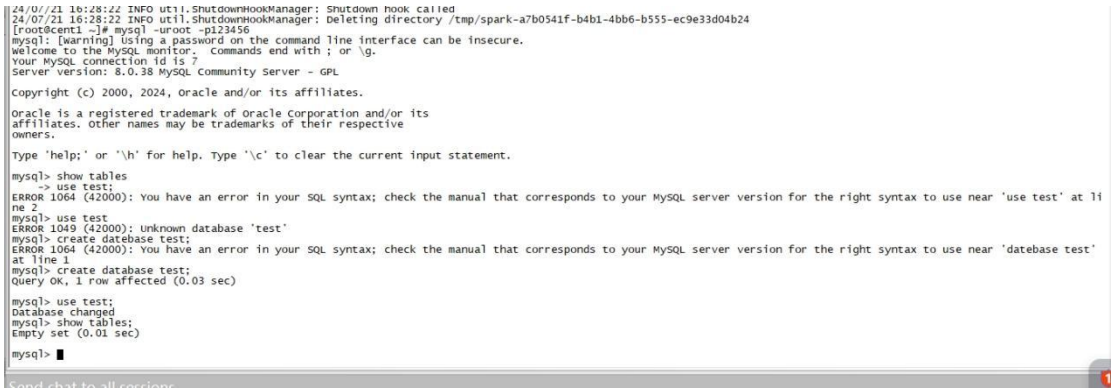


运行后 HDFS 会出现处理的结果数据：

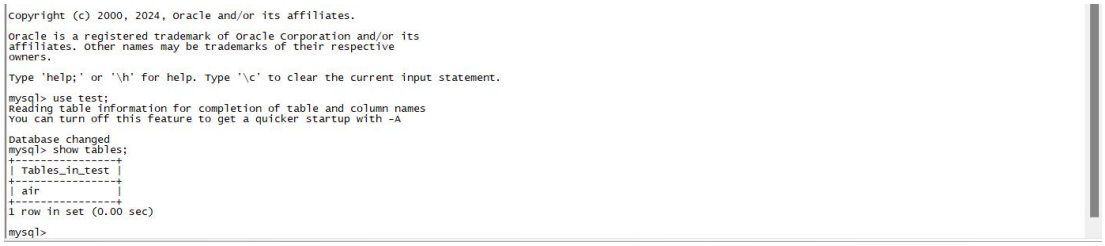


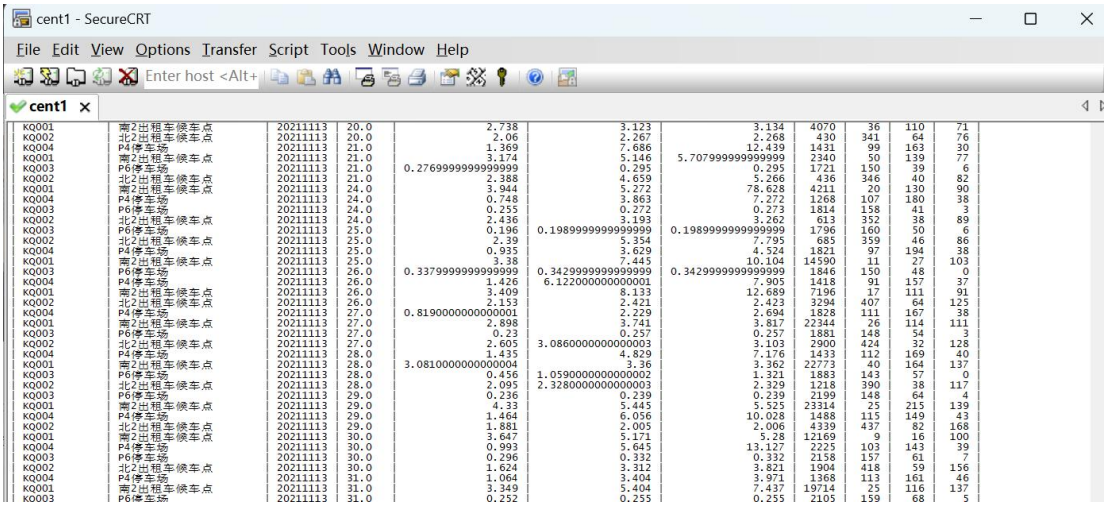
运行前后 linux 中 mysql 数据库对比：

运行前



运行后





之后修改端口即可访问集群节点 1 的数据

6.2 数据设计

一些数据需要进行实时计算，我们将处理数据的函数写在了前端，例如计算AQI 的函数如下：

```
let PM25_IQA = 0, PM10_IQA = 0, NO2_IQA = 0, CO_IQA = 0, O3_IQA = 0, SO2_IQA = 0;

if(item.air_SO2 > 0 && item.air_SO2 <= 50) {
  SO2_IQA = item.air_SO2 * 50 / 50 + 0;
} else if(item.air_SO2 > 50 && item.air_SO2 <= 150) {
  SO2_IQA = (item.air_SO2 - 50) / 2 + 50;
} else if(item.air_SO2 > 150 && item.air_SO2 <= 475) {
  SO2_IQA = (item.air_SO2 - 150) * 50 / 325 + 100;
} else if(item.air_SO2 > 475 && item.air_SO2 <= 800) {
  SO2_IQA = (item.air_SO2 - 475) * 50 / 325 + 150;
} else if(item.air_SO2 > 800 && item.air_SO2 <= 1600) {
  SO2_IQA = (item.air_SO2 - 800) * 100 / 800 + 200;
} else if(item.air_SO2 > 1600 && item.air_SO2 <= 2100) {
  SO2_IQA = (item.air_SO2 - 1600) * 100 / 500 + 300;
} else if(item.air_SO2 > 2100 && item.air_SO2 <= 2620) {
  SO2_IQA = (item.air_SO2 - 2100) * 100 / 520 + 400;
}

if(item.air_NO2 > 0 && item.air_NO2 <= 40) {
  NO2_IQA = item.air_NO2 * 50 / 40 + 0;
} else if(item.air_NO2 > 40 && item.air_NO2 <= 80) {
  NO2_IQA = (item.air_NO2 - 40) * 50 / 40 + 50;
} else if(item.air_NO2 > 80 && item.air_NO2 <= 180) {
  NO2_IQA = (item.air_NO2 - 80) * 50 / 100 + 100;
```

七、页面展示

7.1 选择登入界面

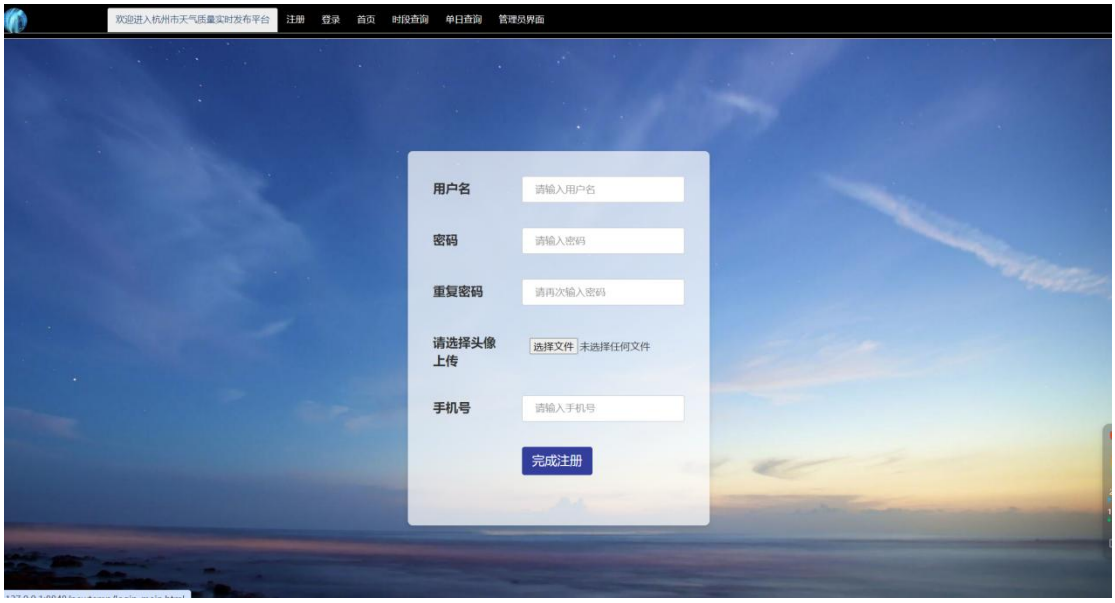
7.1.1 初始界面

初始页面，该页面分为登录、注册、管理员登录，其中登录注册访问的是本地数据库，用户登录后才可以使用本平台的一系列功能。

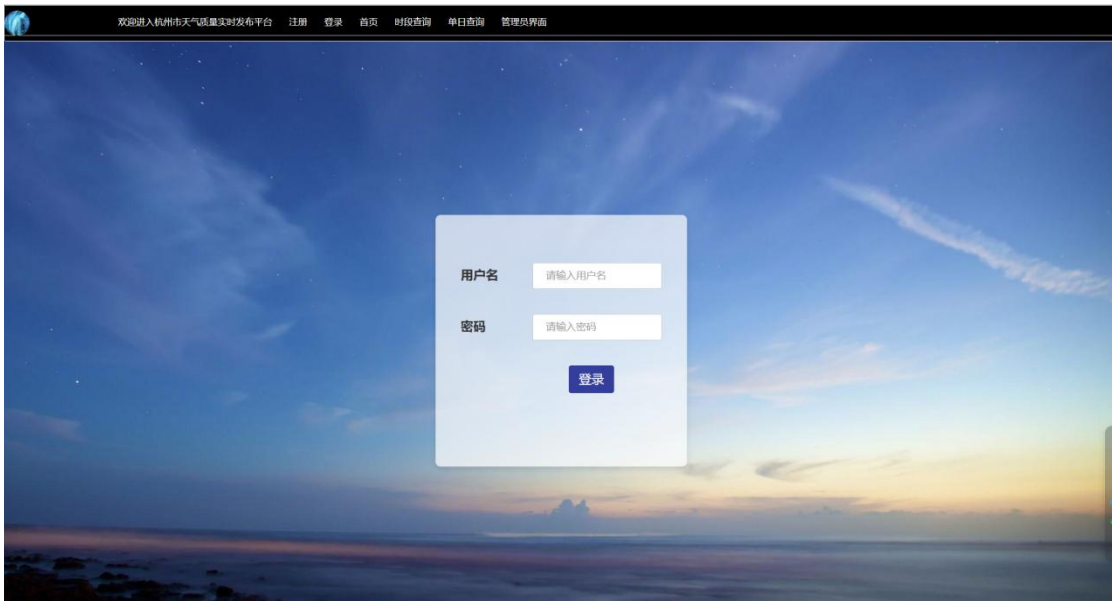
管理员账号提前预设，符合账号为‘admin’密码为‘666’后登录管理员界面。管理员可以管理所有注册本平台的用户。

7.1.2 用户注册界面

用户注册时自定义用户名，密码，可以选择上传自定义头像，或者使用默认头像，用户的注册数据将会存入后端的本地数据库中：

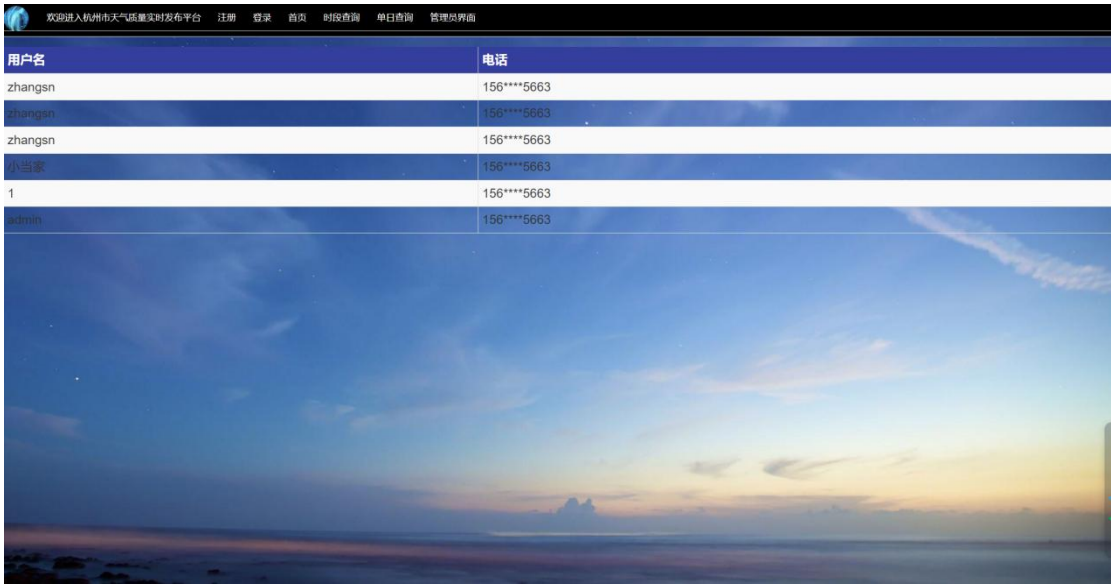


7.1.3 用户登录界面



7.1.4 管理员界面

管理员登录后会直接跳转到该页面，查询用户信息的同时也可以查看正常页面的数据：

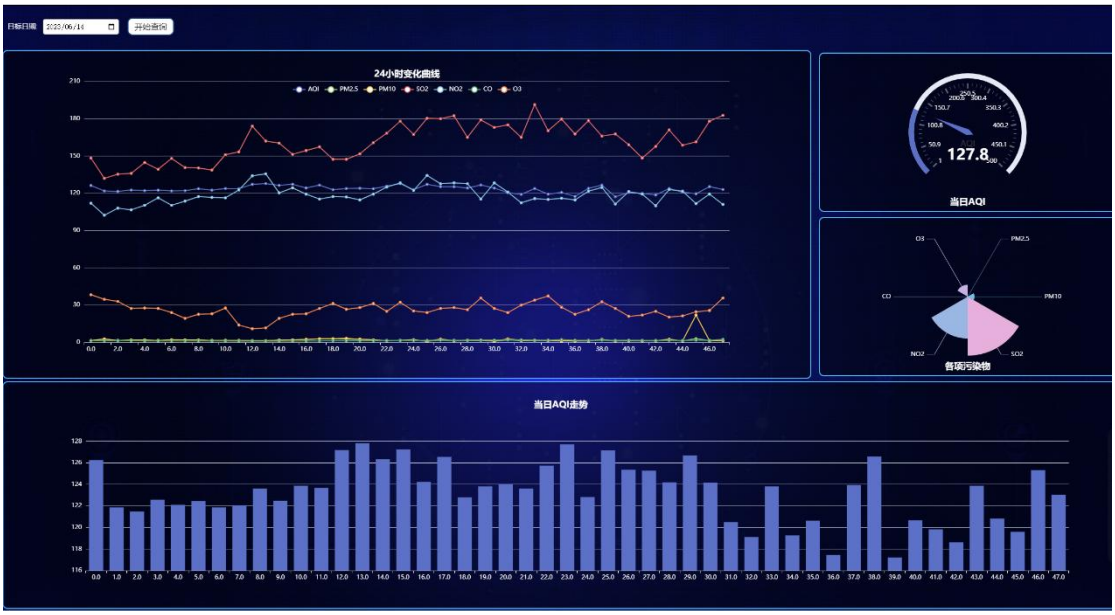


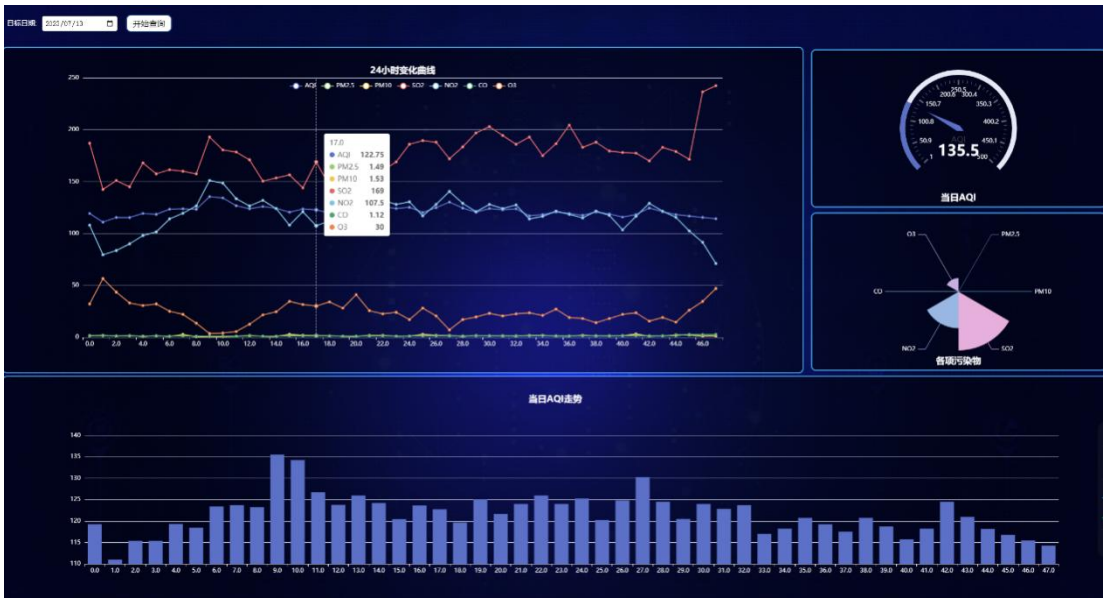
7.1.5 首页

7.2 统计空气质量

7.2.1 单日查询

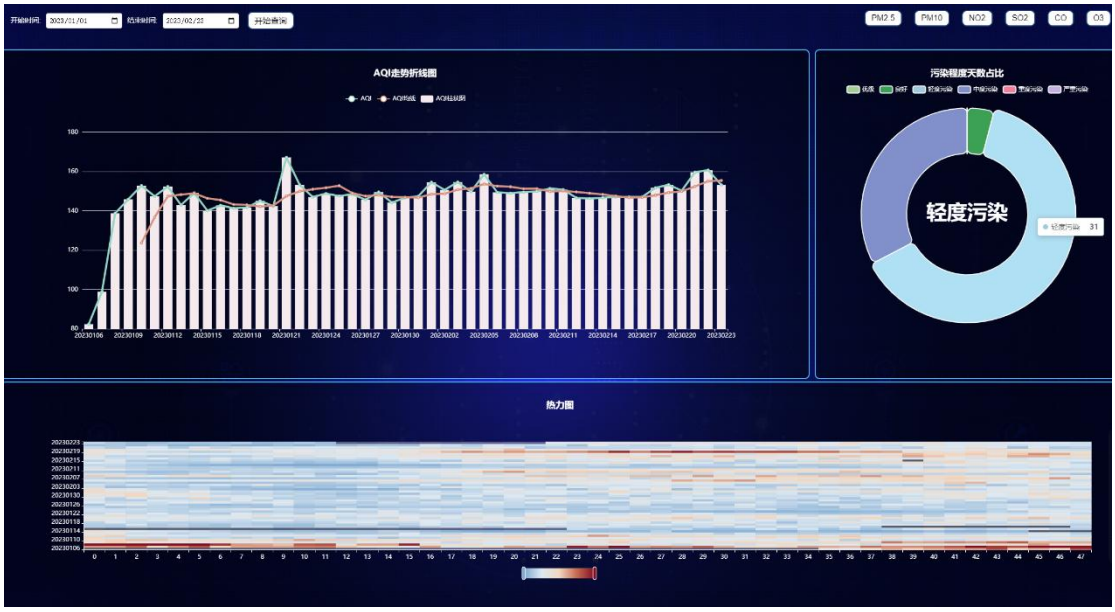
锁定具体日期，查询一天具体的空气质量数据变化，这里我们用折线图、柱状图、仪表盘、玫瑰图来呈现数据信息：





7.2.2 时段查询

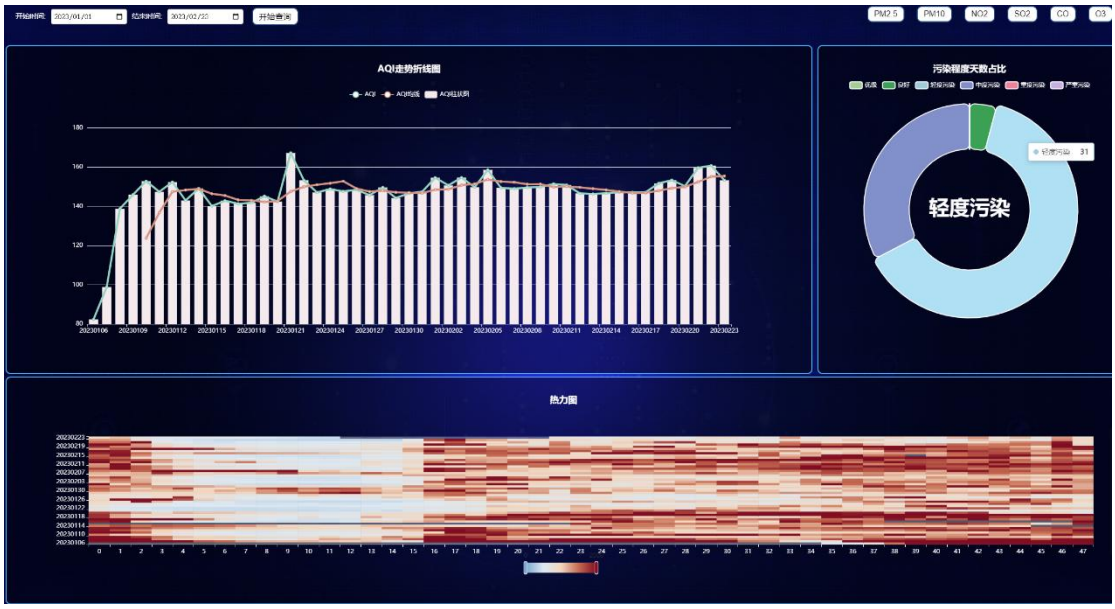
锁定起始时间、终止时间，查询一段时间内的空气质量数据，这里用柱状图、折线图、环形图、热力图来呈现数据：

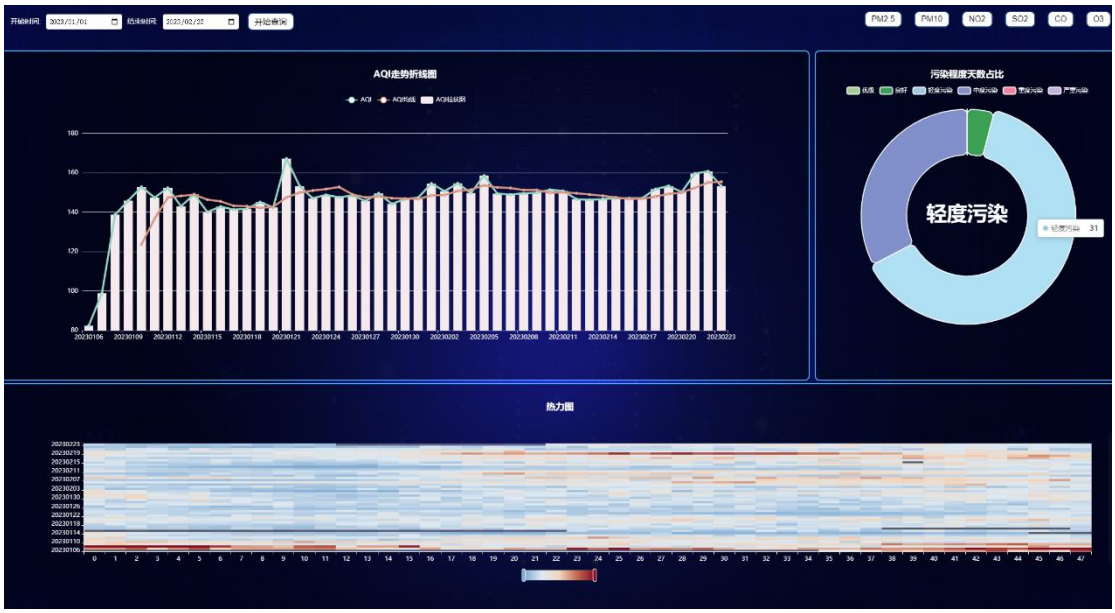


7.2.3 统计各污染物热力

在时段查询中，由于污染物有多种，我们为用户设定了可以指定查询的污染物种类来确定热力图。

锁定污染物种类（PM2.5,PM10,NO2,SO2,CO,O3），查询一天之中污染物热力图：


开始时间
2023/01/01
结束时间
2023/02/28
开始查询
PM2.5
PM10
NO2
SO2
CO
O3
AQI走势折线图
AQI
AQI均线
AQI柱状图
污染程度天数占比
优质
良好
轻度污染
中度污染
重度污染
严重污染
轻度污染 31
轻度污染
热力图


开始时间
2023/01/01
结束时间
2023/02/28
开始查询
PM2.5
PM10
NO2
SO2
CO
O3
AQI走势折线图
AQI
AQI均线
AQI柱状图
污染程度天数占比
优质
良好
轻度污染
中度污染
重度污染
严重污染
轻度污染 31
轻度污染
热力图