

数值预报产品分布式处理与存储系统设计

王建荣, 华连生, 唐怀瓯, 王 云, 王 静

(安徽省气象信息中心, 安徽 合肥 230031)

摘 要: 气象数值预报产品数据日益增长, 传统的关系型数据库对其存储和管理能力不足, 查询规模较大的历史数据时效率较低。针对上述问题, 设计了分布式的数值预报产品处理与存储系统。通过 Quartz 任务调度定时采集数值预报产品文件; 运用 Kafka 分布式消息队列解耦数值预报产品解码与入库程序; 将解码日志文件、原始产品文件和解码得到的要素 GRIB 文件写入 HDFS 分布式文件系统, 应用 MapReduce 分布式程序将解码日志记录存入 HBase。因 HBase 对 Rowkey 的一级索引支持较好, 而对多条件查询支持不足, 需辅助 Solr 索引加以优化。HBase 接收数据时自动触发协处理器同步记录到 Solr 索引库, 实现了 HBase 的二级索引。测试结果表明, 产品文件写入 Hadoop 文件系统平均速度为 82.54 MB/s, 而 HBase 最快入库速度可达每秒 13 677 条, 数据检索结果返回时效达到毫秒级, 能够满足业务应用中对数值预报产品存储和检索时效的要求。

关键词: Quartz; 解码日志文件; Kafka; HBase; Solr; 协处理器

中图分类号: TP302

文献标识码: A

文章编号: 1673-629X(2018)02-0167-06

doi: 10.3969/j.issn.1673-629X.2018.02.036

Design of Distributed NWP Data Processing and Storage System

WANG Jian-rong, HUA Lian-sheng, TANG Huai-ou, WANG Yun, WANG Jing

(Anhui Meteorological Information Center, Hefei 230031, China)

Abstract: With the rapid growth of global and regional numerical weather prediction (NWP) products, traditional relational database has insufficient storage and management for the mass data and its query efficiency is low in long-time-series data accessing. Therefore, we design a distributed data processing and storage system. The system copies NWP files from source folders by using the Kafka Quartz scheduler and decouples NWP products decoding and storage programs by using Kafka distributed message queue. It also writes the decoding log files, source products and element GRIB files into HDFS and then inserts the decoding log file records into HBase. Because the HBase has better support for the first level index of Rowkey, but it is not enough to support the multi condition query, it is necessary to optimize the query using Solr index. HBase receives the data meanwhile it automatically triggers the coprocessor to write records synchronously to SolrCloud, which realizes the multi condition index in HBase. The test shows that the average speed of product file to Hadoop file system is 82.54 MB per second, fastest storage speed can be up to 13 677 records per second and the response time of data retrieval is up to millisecond level, thus it can meet the performance requirement of the storage and retrieval time of NWP data in business applications.

Key words: Quartz; decoding log file; Kafka; HBase; Solr; coprocessor

0 引言

数值预报产品是 14 大类气象资料之一, 是天气预报、分析和气候预测的重要资料来源, 在科研和业务中发挥了重要作用。

中国气象局 CMISS^[1-2] (全国综合气象信息共享平台) 数据库中存储了多种数值预报产品信息, 包含起报时间、预报时效、层次、预报要素代码、区域代码、单要素 GRIB 文件路径等字段, 而具体的 GRIB 文件存储在 GPFS 文件系统中。为确保 Oracle 数据库的稳定

运行, 数值预报产品记录保存 3~6 个月, 并定时清除表空间。在科研和业务工作中, 往往需要长时间序列的数值预报产品数据, 并且要求实时检索性能, 因此考虑利用分布式架构来解决海量气象数据存储检索所面临的问题。

在分布式存储和计算技术中, Hadoop 框架具有高吞吐量、高并发、高容错性、高可靠性、低成本等优势。目前基于 Hadoop 生态系统的气象数据存储方案成为国内外的研究热点。李永生等^[3] 选用 Hadoop 与

收稿日期: 2017-03-22

修回日期: 2017-07-27

网络出版时间: 2017-11-15

基金项目: 中国气象局关键技术集成项目 (CMAGJ2015M29); 安徽省气象局科技发展基金项目 (KM201604)

作者简介: 王建荣 (1981-) 男, 工程师, 研究方向为分布式计算、数据库系统设计; 华连生, 高工, 研究方向为气象信息系统设计。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20171115.1438.068.html>

HBase 相结合的方式设计数值预报产品服务平台;陈东辉等^[4]详细介绍了基于 HBase 的气象地面分钟数据分布式存储系统。文中选取 HBase 数据库实现气象数据文件的分布式存储管理;使用 Quartz 定时采集数值预报产品文件;利用 Kafka 消息队列将文件采集、产品解码、存储入库功能解耦;进行前端 GRIB 解码入库性能优化和后端数据检索性能优化。实验测试验证了

数值预报产品分布式处理与存储系统设计的可行性,为海量气象数据的处理、存储和检索服务提供一种解决方法。

1 系统整体设计

1.1 系统功能模块

系统功能模块如图 1 所示。

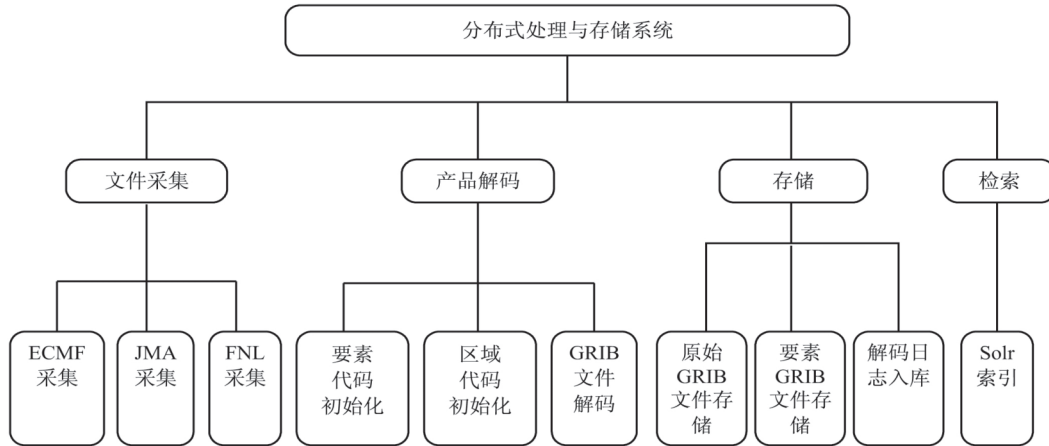


图 1 系统功能模块

(1) 文件采集模块。

通过 Quartz scheduler 定时从数值预报产品目录复制 GRIB 文件到解码程序入口目录。

(2) 产品解码模块。

调用 GRIB API^[5-6]实现 GRIB1、GRIB2 文件的解码,并且生成解码日志文件和要素 GRIB 文件(GRIB2 格式)。

(3) 数据存储模块。

调用 HDFS^[7-8] API 将产品文件、要素 GRIB 文件和解码日志文件上传至 HDFS 分布式文件系统。另一方面,使用 MapReduce 并程序将解码日志文件存入 HBase。

(4) 数据检索模块。

利用 Solr 实现 HBase 的辅助索引,提高数值预报产品数据的检索效率。

1.2 系统总体流程

系统一次完整的执行流程如图 2 所示。

执行步骤如下:

- (1) Quartz 周期性调度完成数值预报产品文件采集和消息入队;
- (2) 解码程序读消息,并根据包含的文件名解码产品;
- (3) 将产品文件、要素 GRIB 文件全部上传至 HDFS;
- (4) 生成解码日志文件如消息队列;
- (5) 入库程序读消息,将日志文件入 HBase;
- (6) HBase 协处理器同步记录至 Solr 索引库。

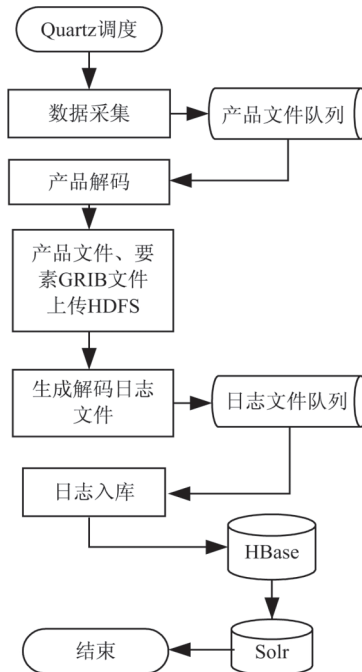


图 2 总体流程

2 Quartz 基本概念及应用

2.1 基本概念

Quartz 是 OpenSymphony 开源组织在任务调度领域的一个开源项目,基于 Java 实现。

2.2 Quartz 定时执行文件采集程序

主要代码如下:

```
Scheduler scheduler = StdSchedulerFactory.getDefaultScheduler();
```

```
scheduler.start();
JobDetail job = JobBuilder.newJob( GribProcessJob.class ).
withIdentity( "job", "group" ).build();
.....
scheduler.scheduleJob( job, trigger );
```

GribProcessJob 类实现 Job 接口,重载 execute 函数,完成 GRIB 文件采集转储和发送消息到产品文件队列的过程。

3 系统消息队列设计

3.1 Kafka 消息队列简介

Apache Kafka 是用 Scala 语言实现的分布式消息队列系统,使用 Zookeeper 进行集群的管理。Kafka 有以下特性:可扩展性、数据分区、低延迟、持久存储、处理大量不同消费者的能力。

Kafka 由 Producer、Broker(消息服务器)和 Consumer 三部分组成,Producer 和 Consumer 均属于客户端。应用程序通过 Producer API 发送消息到 Broker 集群 Leader(主节点),再通过 Consumer API 从 Broker 服务器消费消息。Kafka 消息的两个重要概念为 Topic(主题)和 Partition(分区)。

3.2 消息队列设计

分布式处理与存储系统创建了两个消息队列:产品文件队列和日志文件队列。为产品文件队列创建名为“gribfilelist”的 topic,每个 topic 包含 3 个 partition;为日志文件队列创建名为“logfilelist”的 topic,每个 topic 也包含 3 个 partition。key 相同的消息都被发送到同一个分区(partition),如所有的 ecmf 文件名被发送到相同的分区,而 jma 文件名被发送到另一个分区。

客户端解码程序完成 GRIB 文件解码后将解码日志文件发送至日志文件队列。

客户端入库程序循环请求消息队列,检查并获取最新的消息后按顺序完成:数值预报产品文件、要素 GRIB 文件和解码日志文件写入 HDFS;解码日志 MapReduce 方式存入 HBase 数据库。

3.3 异步发送模式

Kafka Producer 的异步发送模式允许进行批量发送:客户端先将消息缓存在内存中,然后一次请求批量发送出去。

配置策略,比如可以指定缓存的消息达到某个量的时候就发出去,或者缓存了固定的时间后就发送出去,可提高消息发送效率、减少服务端的 I/O 次数。

4 数值预报产品解码

4.1 GRIB 概述

GRIB 码即二进制格点加工数据,是 WMO(世界

气象组织)推荐使用的表格驱动代码之一,主要用来表示天气分析和预报的产品资料。现行的 GRIB 码有两个版本(Edition),即 GRIB1 和 GRIB2。GRIB2 对数据的描述基于模板和码表,而模板引用码表。

4.2 GRIB API 概述

GRIB API 是由 ECMWF(欧洲中期天气预报中心)设计研发的,为用户提供了 C/C++、Fortran 等语言的编程接口。用户程序使用 key/value(键/值)方法存取 GRIB 数据。GRIB 文件中所有信息(Message)都通过 key 来检索。每个 key 都有固定的类型,如实型、整型、字符串等。

4.3 使用 GRIB API 解码

系统采用 GRIB API 的 C/C++ 接口实现数值预报产品解码。以 ECMF 产品为例,Quartz 每 5 min 执行文件采集任务,从 ECMF 产品目录复制文件至解码程序临时目录 temp 下,例如产品文件名为:

W_NAFP_C_ECMF_20160511055659_P_C1D05110000051100011.bin

解码程序调用 GRIB API 对其进行解码后输出解码日志文件和要素 GRIB 文件:

W_NAFP_C_ECMF_20160511055659_P_C1D05110000051100011.bin.log

该文件由多条解码记录组成,单条记录的格式如下:

1|20160511|0|0|98|0|100|850|WIV|ANEA|250|250|NAFP_ECMF_0_FTM-98-ANEA-WIV-250X250-100-850-999998-999998-999998-2016051100-0.GRB

各字段用“|”分割,字段定义与表 1 相对应,而 NAFP_ECMF_0_FTM-98-ANEA-WIV-250X250-100-850-999998-999998-999998-2016051100-0.GRB 即是要素 GRIB 文件。文件名含义:加工中心代码为欧洲中期天气预报中心、预报分辨率为 0.25°×0.25°、850 hPa 等压面层格点经纬度范围(60°,-10°,-60°,150°)的纬向风资料,其存储于 HDFS 分布式文件系统 ECMWF 相关存储路径下。

5 数据存储模型设计

5.1 HBase 简介

HBase(Hadoop database)^[9-10]运行在 HDFS 分布式文件系统上,使用 Zookeeper 管理集群,提供高可靠性、高性能、列存储、可伸缩、实时读写特性,主要用来存储非结构化和半结构化的松散数据。

5.2 数据存储模型

系统将数值预报产品通过 GRIB API 解码后存储在 HBase 中,不同的数值预报产品分开存储在不同的

实体数据表中,目前存储了 3 大类数值预报产品,分别为 ECMWF(欧洲中期数值预报中心)发布的细网格($0.25^{\circ} \times 0.25^{\circ}$ 水平分辨率)和粗网格($2.5^{\circ} \times 2.5^{\circ}$ 水平分辨率)的数值预报产品,JMA(日本气象厅)发布的 $0.5^{\circ} \times 0.5^{\circ}$ 水平分辨率和 $1.25^{\circ} \times 1.25^{\circ}$ 水平分辨率的数值预报产品,NCEP/FNL 再分析资料。数据表以行键、列族、数据的方式存储数值产品的实体数据。数据表存储内容见表 1。

表 1 数据表存储内容说明

存储列名	含义说明
data: d_iymdhm	入库时间
data: date	资料时间
data: year	年
data: month	月
data: day	日
data: hour	时
data: validtime	预报时效
data: fieldtype	场类型(产品类型)
data: centre	加工中心代码
data: subcentre	子中心标识
data: elementcode	预报要素代码
data: areacode	预报区域代码
data: iIncrement	I 方向增量
data: jIncrement	J 方向增量
data: gribpath	要素文件存储路径
data: srcfilename	源文件名

data: gribpath 是解码所得要素 GRIB 文件在 HDFS 中的存储路径。

选取表 1 中 data: date、data: validtime 和 data: centre 三列做数据模型展示,见表 2。

表 2 数据模型示例表

行键	时间戳	data		
		data: date	data: validtime	data: centre
rk ₁	T ₁	20160511	003	98
rk ₂	T ₂	20160511	006	98
rk _N	T _N	20160511	012	98

Rowkey(行键): HBase 中的 Rowkey 唯一标识一行记录。根据 HBase 的优化原则^[7],Rowkey 的长度易固定且不超过 200 Bytes,设计如下: AAAAATTT: yyyyMMdd: nnnmmmm: IIIIJJJJ

其中 AAAA 为 5 字母长度的英文缩写,不足 5 位则在其后补“9”,代表数值预报产品的预报要素名称;TTT 为预报时效;nnn 表示高度层类型,mmmm 表示层次;IIII 表示 4 位 I 方向增量,不足 4 位则前导

置“0”;JJJJ 表示 4 位 J 方向增量,不足 4 位则前导置 0。

以 ECMF 数据表的行键为例:

TEMP9006: 20160511: 1000010: 0250X0250

其含义是:对于温度要素(temp),在 2016 年 5 月 11 日 00:00 起报,预报时效为未来 6 h 的预报场,预报层次为 10 hPa,I 方向增量为 0.25° ,J 方向增量为 0.25° 。

时间戳(timestamp):每条数据更新的历史记录,同一行键数据再次入库会记录不同的时间戳。

列族(column family):每种数值预报产品的表结构基本相同,每张表只设一个列族 data,包含的列(column qualifier)有 data: date、data: validtime、data: centre、data: gribpath 等。HBase 存储的都是 Byte 数组。

6 基于 Solr 的二级索引设计

6.1 Solr 简介

Apache Solr 是一种开源的、基于 Lucene 的全文检索引擎,支持 XML、JSON 和 python 等常用的输出格式。而 SolrCloud^[11-12] 是基于 Solr 和 Zookeeper 的分布式搜索方案,使用 Zookeeper 作为集群的配置信息中心。

6.2 二级索引设计

HBase 在存储时默认按照 Rowkey 进行排序(字典序)并通过 Rowkey 及其 range 来检索数据,在 HBase 查询时,有以下几种方式:

(1) 通过 get 方式,指定 Rowkey 获取唯一一条记录;

(2) 通过 scan 方式,设置 startRow 和 stopRow 参数进行范围匹配;

(3) 全表扫描,即直接扫描整张表中所有行记录。

HBase 对 Rowkey 的一级索引支持较好,按 Rowkey 查询的响应时间达到毫秒级。HBase 内置 Filter(过滤器)特性以支持多条件查询的二级索引。但 HBase 的 Filter 是直接扫描记录的,如果数据范围很大,会导致查询速度很慢。因此基于 Solr 来实现二级索引,满足 Rowkey 之外的多要素数据检索需求。

设计 Solr 索引的关键问题是合理地配置索引字段。Zookeeper 统一管理 XML 格式的 Solr 索引字段描述文件: managed-schema,SolrCloud 各实例共享同一个 managed-schema。

主要配置如下:

```
<field name="id" type="string" indexed="true" stored="true" required="true" multiValued="false" />
```

```
<field name="edition" type="string" indexed="true" stored="true" required="false" />
```

.....

<uniqueKey>id</uniqueKey>

设置 HBase 表的 id 字段为 Solr 索引的 unique-Key, 存储 HBase 记录的 Rowkey 值。

7 数值预报产品数据入库性能优化

上文所介绍的 Solr 索引设计是入库性能优化的前提。

7.1 解码日志入库流程

入库程序采用了 MapReduce 编程模型^[13-14]。MapReduce 作业读取解码日志文件插入到 HBase 数据库中。解码程序省略了 reduce 步骤, 因 mapper 输出中间数据到 reducer 需要通过网络, 受限于 Hadoop 集群带宽。

7.2 HBase 协处理器

HBase 的协处理器^[15] (Coprocessor) 分为两类, Observer 和 EndPoint, 其中 Observer 的代码部署在服务端, 相当于对 API 调用的代理。系统选用 RegionObserver 接口。

7.3 HBase 协处理器向 Solr 同步记录

HBase 协处理器需要获取 HBase 的插入和更新操作: 拦截 put 操作, 获取其内容, 同步写入 Solr。HBase 协处理器定义以及同步数据到 Solr 的主要代码如下:

```
public class SolrIndexCoprocessorObserver extends BaseRegionObserver {
    @Override
    public void postPut( ObserverContext<RegionCoprocessorEnvironment> e, Put put, WALEdit edit, Durability durability) throws IOException {
        String rowKey = Bytes.toString( put.getRow() );
        try {
            Cell cellEdit = put.get( Bytes.toBytes( " data" ), Bytes.toBytes( " edition" ) ).get( 0 );
            String strEdit = new String( CellUtil.cloneValue( cellEdit ) );
            .....
            SolrInputDocument doc = new SolrInputDocument();
            doc.addField( "id", rowKey );
            doc.addField( "edition", strEdit );
            .....
            //写入缓冲
            SolrWriter.addDocToCache( doc );
        }
    }
}
```

8 性能测试

8.1 测试环境

(1) 软件及版本: Quartz-2.2.3; hadoop-2.6.0; zookeeper-3.4.6; solr 5.5.4; hbase-1.2.2; GRIB API 1.13.

1。

(2) 硬件配置。

测试环境由 6 台 X86 架构的服务器组成, 操作系统均为 64 位 Ubuntu 14.04。其中 5 台服务器构建 Hadoop、Zookeeper、HBase、Solr 集群, 1 台部署数值预报产品解码入库程序。

处理器: Intel Core i5-3470 3.20 GHz;

磁盘: 1TB, 7200 rpm, SATA III 接口;

内存: 16 GB;

网络环境为千兆局域网。

8.2 测试对象和方法

选取 ECMWF 高分辨率数值预报产品及其解码产生的要素 GRIB 文件为测试对象, 其常见的文件大小分布为: 约 2 MB、约 10 MB、约 105 MB 和约 160 MB, 而解码得到的要素 GRIB 文件数也随之不同。

(1) HDFS 写入性能。

数值预报产品有 846 个文件, 共 96 GB, 平均大小 116 M。客户端程序调用 HDFS API 的文件复制操作将数值预报产品文件写入 HDFS 文件系统需要的时间为 1 190.986 s, 平均写文件速度为 82.54 MB/s; 要素 GRIB 文件上传至 HDFS 集群的速度近似。

(2) HBase 入库性能。

采用统计学方法: 总体有 96 360 个解码日志文件, 共 57 816 000 条记录, 耗时 4 576.9 s, 平均写入速度 12 632 条/s; 随机抽取 1 000, 2 000, ..., 10 000 条记录入库, 见图 3。测试结果表明, 随着入库记录数的增加, 数据入库性能总体平稳, 最快写入速度为 13 677 条/s。

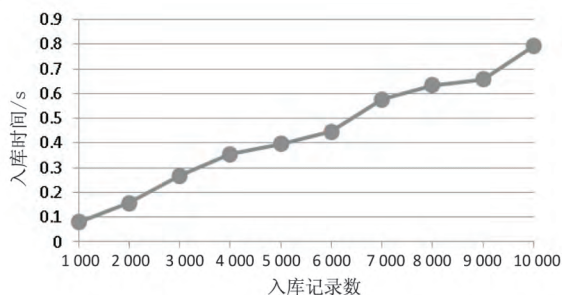


图3 入库时间和入库记录数的关系

(3) 索引完整性验证。

测试用例如下:

用例编号 UC1: 按起报时间、预报层次、预报时效、单预报要素检索预报要素场;

用例编号 UC2: 按起报时间范围、预报层次、预报时效、单预报要素检索预报要素场;

用例编号 UC3: 按起报时间、预报层次、预报时效、多预报要素检索预报要素场。

基于 HBase Filter^[16] 的条件过滤查询和辅助索引

查询返回的记录数对比如表 3 所示。

表 3 HBase Filter 与 SolrCloud 查询记录数对比

测试用例	Filter 检索			Solr 检索		
	No.1	No.2	No.3	No.1	No.2	No.3
UC1	1	1	1	1	1	1
UC2	1 359	63 208	478 683	1 359	63 208	478 683
UC3	17	20	22	17	20	22

表 3 中每个测试用例均做了 3 组对比,基于 SolrCloud 索引的查询记录数均和 HBase Filter 查询的记录数一致,说明索引完整可用。

(4) HBase 检索性能。

表 3 中各测试用例最大查询记录数所需时间对比如表 4 所示。

表 4 HBase Filter 与 Solr 查询效率对比

测试用例	Filter 检索 时间/s	Solr 检索 时间/s
UC1	1.548	0.262
UC2	67.832	2.749
UC3	3.517	0.736

由表 4 可知,基于 SolrCloud 的查询效率远远高于 HBase Filter 查询,按时间点的查询基本都在毫秒级返回结果;对于 UC2 中,按时间范围检索方面,HBase Filter 效率较低,不适合时间序列的查询,在实际的气象业务应用中,需要结合 Solr 对 HBase 进行索引优化,来满足检索时效的要求。

9 结束语

针对关系型数据库对数值预报产品数据的存储及检索效率低等问题,设计了分布式处理与存储系统。利用 Quartz 任务调度采集数值预报产品文件,Kafka 消息队列解耦数值产品解码与入库程序,研究 HBase 分布式数据库结合 SolrCloud 索引服务的数据存储与检索优化方案,设计了适合气象业务应用的数值预报产品数据存储模型,并建立 Solr 索引。关键技术是前端 MapReduce 并行程序入库、HBase 协处理器同步记录至 SolrCloud。实验测试表明,该方案提高了存储效

率和检索速度,能够满足业务中的时效性要求。

参考文献:

- [1] 熊安元,赵芳,王颖,等.全国综合气象信息共享系统的设计与实现[J].应用气象学报,2015,26(4):500-512.
- [2] 杨润芝,马强,李德泉,等.内存转发模型在 CIMISS 数据收发系统中的应用[J].应用气象学报,2012,23(3):377-384.
- [3] 李永生,曾沁,徐美红,等.基于 Hadoop 的数值预报产品服务平台设计与实现[J].应用气象学报,2015,26(1):122-128.
- [4] 陈东辉,曾乐,梁中军,等.基于 HBase 的气象地面分钟数据分布式存储系统[J].计算机应用,2014,34(9):2617-2621.
- [5] 张芳,周峥嵘,刘媛媛.ECMWF GRIB API 及其应用[C]//中国气象学会气象通信与信息技术委员会暨国家气象信息中心科技年会.北京:国家气象信息中心,2011.
- [6] 李蔚.NECP FNL 资料解码及数据格式转换[J].气象与减灾研究,2011,34(1):63-68.
- [7] WHITE T.Hadoop: the definitive guide,3E[M].[s.l.]: O'Reilly Media,2012.
- [8] DUTTA H,KAMIL A,POOLERY M,et al.Distributed storage of large-scale multidimensional electroencephalogram data using Hadoop and HBase[M]//Grid and cloud database management.Berlin: Springer,2011.
- [9] GEORGE L.HBase: the definitive guide[M].Sebastopol: O'Reilly Media,2011.
- [10] STONEBRAKER M. SQL databases v. NoSQL databases[J].Communications of the ACM,2010,53(4):10-11.
- [11] 郝强,高占春.基于 SolrCloud 的网络百科检索服务的实现[J].软件,2015,36(12):103-107.
- [12] 付剑生,徐林龙,林文斌.分布式全网职位搜索引擎的研究与实现[J].计算机技术与发展,2015,25(5):6-9.
- [13] 杨润芝,沈文海,肖卫青,等.基于 MapReduce 计算模型的气象资料处理调优试验[J].应用气象学报,2014,25(5):618-628.
- [14] 李永生,曾沁,杨玉红,等.基于大数据技术的气象算法并行化研究[J].计算机技术与发展,2016,26(9):47-49.
- [15] 邹敏昊.基于 Lucene 的 HBase 全文检索功能的设计与实现[D].南京:南京大学,2013.
- [16] 张叶,许国艳,花青.基于 HBase 的矢量空间数据存储与访问优化[J].计算机应用,2015,35(11):3102-3105.