

李永生, 曾沁, 徐美红, 等. 基于 Hadoop 的数值预报产品服务设计与实现. 应用气象学报, 2015, 26(1): 122-128.  
doi:10.11898/1001-7313.20150113

# 基于 Hadoop 的数值预报产品服务设计与实现

李永生\* 曾沁 徐美红 石小英

(广东省气象信息中心, 广州 510080)

## 摘 要

数值预报产品数据与日俱增, 采用传统的关系型数据库对其进行存储和管理存在效率低和存储能力不足的问题。另外, 基于文件的存储方式在数据存储处理、数据读取和算法计算等方面存在性能瓶颈。针对这一问题, 基于 Hadoop 技术体系设计了分布式的数据存储模型, 实现了数值预报产品数据的分布式存储和处理, 开发了数值预报产品数据接入处理模块; 并实现了基于 Rest Web Service 的获取数值预报产品要素场数据访问接口、时间序列数据访问接口、数据下载接口等业务应用接口。多业务用户的实际业务测试表明, 该平台在诸如数值预报产品气象数据处理和业务应用方面较传统技术架构具有一定优势。

**关键词:** Hadoop 技术体系; 气象数据; Web Service 接口

## 引 言

气象数据是气象业务和科研工作的基础, 数据蕴含巨大的价值。近年来, 我国气象事业有了长足进步, 也建设了大量的常规、特种自动气象探测设备, 其传输、处理和服务应用的信息量呈指数形式增长。这些数据包括结构化的数据, 如自动气象站观测数据、人工观测数据等, 也包括精细化数值预报产品、风云系列气象卫星产品和气象雷达产品等数据, 对于结构化的数据可以通过关系型数据库进行分析、处理和计算, 并取得了良好的业务应用效果, 对于数值预报产品等数据大多基于文件方式(如 grib 格式文件)存储和处理, 再通过专业的软件绘图后应用于实际业务中。因此, 探讨新的针对数值预报产品数据进行分析处理的方式是目前气象业务应用的迫切需求, 以实现这类气象数据资源的高效利用和共享。

如何做好气象数据服务一直是气象信息技术发展的重要方向之一, 李集明等<sup>[1]</sup>从数据资源管理和

共享发布两个方面分别介绍了元数据技术、数据网格技术、海量数据存储管理技术、数据挖掘技术等构建气象信息共享平台的关键技术; 王伯民等<sup>[2]</sup>对气象科学数据发布策略进行了研究, 本文以气象数值预报产品这类典型的气象数据为数据源, 基于 Hadoop 技术体系实现气象数据文件的分布式存储和管理。按照资源即服务的理念, 对数据服务进行基于 SOA 架构的封装, 使之 Web 服务化, 实际业务测试表明, 基于 Hadoop 的数值预报产品服务设计与实现与气象业务发展现状和需求结合紧密。

## 1 平台基础架构设计与实现

从数值预报产品数据的入库方式和性能、特定条件查询搜索、获取要素场数据接口设计等方面综合考虑, 以 Hadoop 基础上集成 Hive, Hadoop 与 HBase 技术组合以及 Hadoop 基础上整合 SciHadoop 3 种技术方案为备选方案, 制定了入库方式、入库记录查询方式、数据存储组织方式等考核指标, 经初步分析判断, 尝试选用后选取 Hadoop 与 HBase

2014-05-19 收到, 2014-09-28 收到再改稿。

资助项目: 广州市科技计划项目(2012Y2-00031, 2013Y2-00053, 2013Y2-00074), 公益性行业(气象)科研专项(GYHY201106009), 广东省气象局重点项目(2012A01)

\* email: lynne5101@163.com

技术组合作为系统的技术实现方案,并在此基础上设计了三层结构的总体框架,具体如图 1 所示。实

际测试结果表明,该技术方案能够满足业务实际需求。

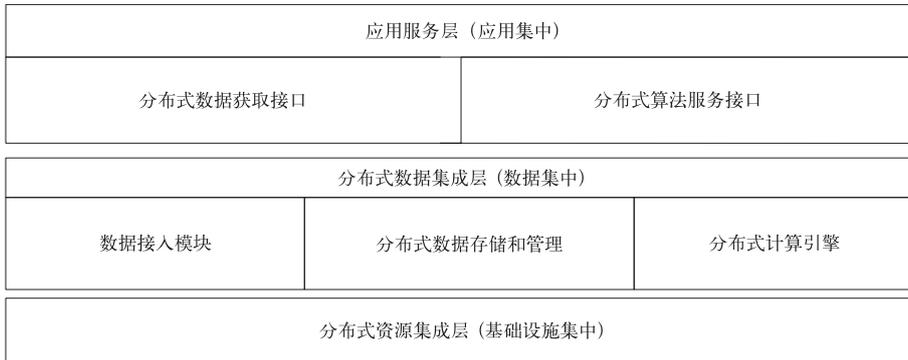


图 1 系统总体框架

Fig. 1 The diagram of the system overall frame

平台的系统架构由分布式资源集成层、分布式数据集成层和应用服务层 3 个部分组成。

分布式资源集成层充分利用 Hadoop 适合在廉价机器上对各种资源数据进行分布式存储和分布式管理,以及具有可伸缩性和高容错性等特点,基于 Hadoop 技术构建了分布式资源集成层<sup>[3]</sup>,该层的主要功能是实现基础设施的集成,即利用 Hadoop 技术将物理上分散的存储和计算资源整合为逻辑上统一的存储和计算资源。

分布式数据集成层包括数据接入模块、分布式数据存储和管理以及分布式计算引擎 3 个组成部分,其中数据接入模块对原始格式的数值预报产品进行解码和格式转换,并开发实现了将数据实时接入平台的功能;分布式数据存储和管理模块基于

HDFS 和 HBase,实现了数据的分布式存储和管理;在分布式计算引擎模块中,MapReduce 通过对数据集的大规模操作分发给 Hadoop 集群中的每个节点进行处理,每个节点会周期性地将完成的工作和状态同步更新到管理节点上<sup>[4]</sup>。该层的主要功能是实现数值预报产品数据的实时接入,并实现数据的分布式存储和管理。

应用服务层包括分布式数据获取接口和分布式算法服务接口<sup>[5]</sup>,为了更好地与其他业务系统实现对接,对外提供了基于 Rest(Representational State Transfer) Web Service 的标准化数据查询和访问接口,该层的功能主要是部署基于 Rest Web Service 数据服务程序,从而在逻辑上对接业务应用系统。

系统功能如图 2 所示。

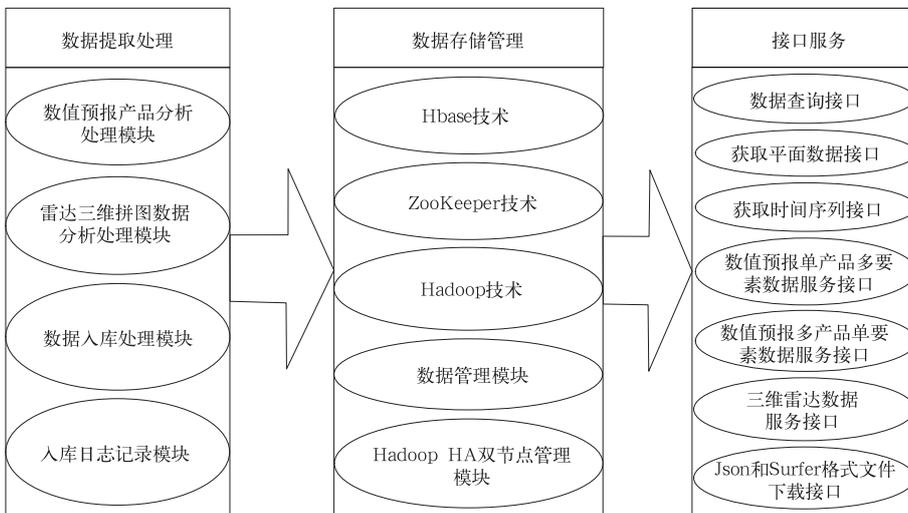


图 2 系统功能结构图

Fig. 2 The system function structure diagram

## 2 数据存储管理模型设计

平台将数值预报产品数据解析后存储在 HBase 中,每种数值预报产品的元数据信息和实体数据分别存储在两个表中,即元数据表和实体数据表,不同的数值预报产品分开存储在不同的一组元数据表和实体数据表中,目前试验平台中实际存储了 8 类数值预报产品,包括 ECMWF(欧洲中期数值预报中心)发布的  $0.25^{\circ} \times 0.25^{\circ}$  水平分辨率和  $2.5^{\circ} \times 2.5^{\circ}$  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,中国华南区域气象中心发布的 12 km 和 36 km 分辨率 GRAPES 产品,逐时同化预报产品(CHAF)以及中国国家气象中心发布的 T639 产品等。其中元数据表中存储产品的所有要素名称列表、时次列表、高度列表、经纬度列表、起报时间列表等元信息,实体数据表以行键、列族、数据的方式存储数值预报产品的实体数据。

### 2.1 元数据存储格式

元数据表中存储数值预报产品的所有要素名称列表、时次列表、高度列表、经纬度列表、起报时间列表等元数据,具体存储的信息如表 1 所示。

表 1 元数据表存储内容说明

Table 1 The description of metadata table content

存储列名	含义说明
meta: variables	当前产品元数据变量,描述了当前元数据表中的数据包含哪些变量,如经度、纬度、高度、日期、状态、要素、起报时间等
meta: dimensions	描述变量的维度信息,如有 56 个经向维度值、68 个纬向维度值等
meta: lat	当前产品纬向维度数据,描述当前产品的所有纬度值
meta: level	当前产品预报层数基本信息,如包含 10 个预报层次数
meta: lon	当前产品经向维度基本信息
meta: time	当前产品时间维度的基本信息,如包含 62 个时间维度

### 2.2 实体数据存储模型设计

气象数值预报产品属于格点场数据,即以二维阵列的格式记载高空要素场的分布,它的每一数据点称为格点,其中要素场包括高度场、风场(南北向和东西向)和温度场等。根据数值预报产品的实际特点,在进行实体数据存储模型设计时引入了命名格式为 AAAATTT 变量, AAAA 为 4 个字母长度的英文缩写,代表数值预报产品的预报要素名称; TTT 为预报时效,如 temp006 表示某一预报时刻起

点算起未来 6 h 的温度预报。变量 AAAATTT 在逻辑上是 4 个维度的变量,表示为 AAAATTT(lon, lat, level, time), 4 个维度均为离散值,其中 lon 是经向维度, lat 是纬向维度, level 是预报层次(如 850 hPa, 925 hPa 等), time 是数值预报开始预报的时刻,如 2013062700, 表示 2013 年 6 月 27 号 00:00(世界时,下同)。每种数值预报产品在元数据表中的存储格式如表 2 所示。

表 2 实体数据模型示意表

Table 2 The schematics table of entity data model

行键	时间戳	列族		
		列 1(如 500 hPa)	列 2(如 700 hPa)	列 N(如 850 hPa)
AAAATTT 变量:起报时间	t1	数据 1	数据 4	数据 7
AAAATTT 变量:起报时间	t2	数据 2	数据 5	数据 8
AAAATTT 变量:起报时间	tN	数据 3	数据 6	数据 9

行键(Rowkey): 数据表的主键。Rowkey 的值由变量 AAAATTT 和起报时间共同确定。如对于产品 CHAF 的温度要素(temp), 在 2012 年 9 月 1

日 00:00 起报, 预报时效为未来 6 h 的预报场, 其 Rowkey 为 temp006:2012090100。

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):每种数值预报产品不同预报层次的预报值存储在不同的列中。每个元数据表中的列族的列数根据具体预报产品决定。

数据(Values):记录该产品的每个行键实际对应的数据,如对于产品 CHAF 的 temp 要素,temp006:2012090100 的预报数据存储的数据(Values)如下:

```
{20.189807 21.235689 21.877865 20.234579
.....}
.....
{21.345678 21.235696 20.456324 20.567795
.....}。
```

### 2.3 入库记录存储格式

入库记录存储在 RowkeyTab 表中,以 Rowkey:temp000:2012080100 形式存储。其中,Rowkey 为 temp000:2012080100 的数据的入库信息,temp000:2012080100 为要素 temp 在 2012 年 8 月 1 日 00:00 起报时间的 000 时次的预报值。

## 3 数据服务接口

### 3.1 服务接口

平台采用 Rest Web Service 提供对外数据服务接口,针对目前的气象业务需求<sup>[6]</sup>,开发了数据查询接口、获取要素场数据接口、获取时间序列数据接口、数值预报单产品多要素数据服务接口、数值预报多产品单要素数据服务接口、三维雷达数据服务接口、Json 和 Surfer 格式文件下载接口等多个数据访问接口。

### 3.2 接口调用方法

接口调用可以通过调用 http 请求方式<sup>[7-9]</sup>嵌入应用程序,实际的调用过程与此类似,程序调用的过程如下:

```
InterfaceURL iu = new InterfaceURL("ip:端口号",InterfaceURL.DATA_LIST);
iu.addParameter("proName", "JMA");
iu.addParameter("factor", factor);
iu.addParameter("startDate", date);
iu.addParameter("endDate", date);
iu.addParameter("forecastStartTime", "00");
```

```
iu.addParameter("forecastTimePeriod", "000");
iu.addParameter("level", "1000.0");
iu.addParameter("startLatitude", "");
iu.addParameter("startLongitude", "");
iu.addParameter("endLatitude", "");
iu.addParameter("endLongitude", "");
iu.addParameter("standardDate", "2012-08-24");
```

其中,InterfaceURL iu = new InterfaceURL("ip:端口号",InterfaceURL.DATA\_LIST);用于指定开启了 Web Service 服务的机器,以及所需调用的接口,接口可参照 InterfaceURL 类中的定义。InterfaceURL.addParameter()方法用于添加请求 URL 的参数<sup>[10]</sup>。通过 InterfaceURL.toString()方法可以获取完整的请求 URL。

## 4 平台业务应用

### 4.1 平台实现

基于 Hadoop 技术在多台 X86 架构服务器<sup>[11]</sup>上构建了系统试验应用平台,系统实现时在现有多个 HDFS NameNode HA 的解决方案选取了能自动双向失效切换的 HA 方案,针对气象业务的实际特点<sup>[12]</sup>设计实现了主节点(NameNode)的自动双向失效切换,系统通过 Zookeeper 技术实现故障切换,从而解决了 Hadoop 技术体系中主节点单点故障问题,实现了平台的可靠性、动态可扩展以及安全一体化等特点,实际模拟试验时,将主节点的进程终止,系统能够正常切换,测试时切换时间未超过 2 min。经过近 1 年的业务试运行结果表明平台稳定可靠;基于 Rest Web Service 实现了数据服务接口的可靠访问,接口的各方面性能指标基本满足气象业务实际需求。

### 4.2 平台业务应用测试

#### 4.2.1 测试硬件环境说明

试验测试环境由型号为 X3650 的 X86 架构的服务器组成,服务器的具体配置:CPU 为 Xeon E5-2603 1.8 GHz,内存容量为 4 GB,每台服务器配 4 块硬盘,硬盘参数为 1 TB 7.2 K 6 Gbps NL SATA 3.5" G2HS HDD,网络环境为千兆局域网。

#### 4.2.2 测试对象和方法

业务试验测试以华南区域逐时循环同化分析与

精细模式预报系统(CHAF)为测试对象, CHAF产品基本信息 lonNum 为 309, latNum 为 267, level 为 1。每层的网格点位  $309 \times 267 = 82503$ 。每个网格点 4 字节存储, 每层数据量约为 330 KB。通过 Rest Web Service 获取数据。

#### 4.2.3 数据接入性能测试

以 CHAF 产品为测试对象, 每轮进行 3 次测试, 然后取平均值, 分别测试数据接入处理模块处理 100, 200, ..., 1000 个要素场数据所花费的平均时间, 测试结果如图 3 所示, 写入 1000 个具有 82503 个格点的数值预报产品数据仅耗时 12 s, 完全满足业务实际需求。

#### 4.2.4 数据接口读取测试

以 CHAF 产品为测试对象, 每轮进行 3 次测试, 然后取平均值, 分别测试通过接口读取 100,

200, ..., 1000 个要素场数据所花费的平均时间<sup>[13]</sup>, 测试结果如图 4 所示。数据读取性能随着数据量的增加缓慢上升, 读取 1000 个具有 82503 个格点的数值预报产品数据仅耗时 4 s。

#### 4.2.5 平台扩展性能测试

平台扩展性能测试以华南区域逐时循环同化分析与精细模式预报系统(CHAF)为测试对象。分别测试节点数不同情况下读取 200 个要素场数据的读取性能, 其中每个要素场的数据为 330 KB。测试每次为集群静态增加 2 个节点, 每个节点的物理存储空间为 4 TB, 修改集群主节点的配置文件后重新启动集群, 待整个平台数据自动再平衡<sup>[13]</sup>后进行读取测试。测试结果表明: 随着平台节点数量的增加, 数据读取性能总体平稳略优。测试结果如图 5 所示。

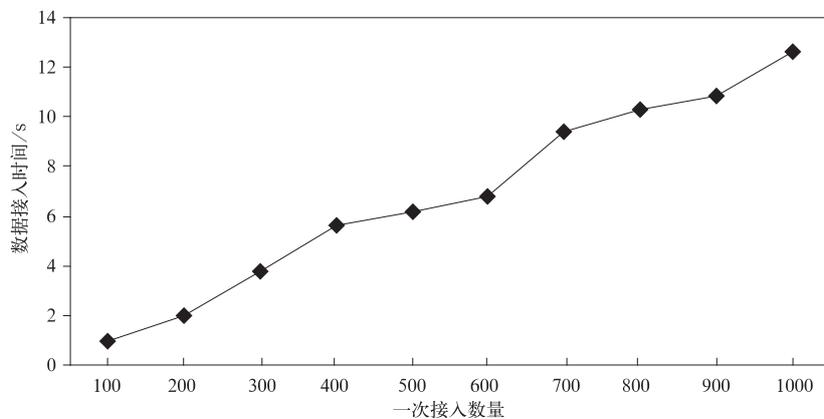


图 3 数据接入性能测试结果

Fig. 3 Results of access performance of data

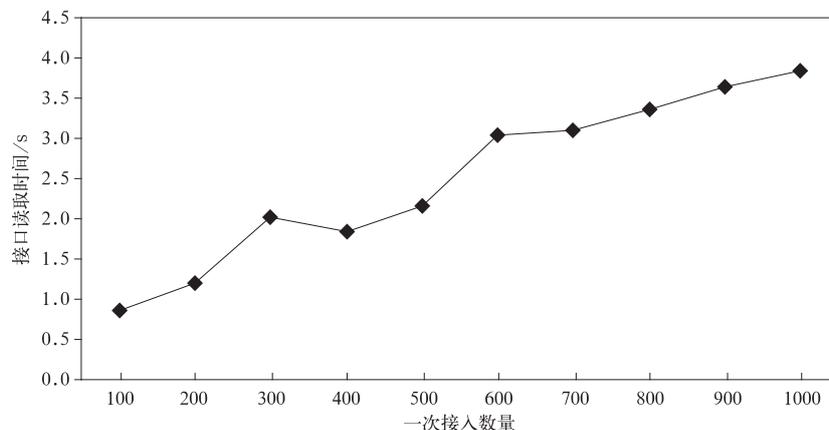


图 4 数据接口读取性能测试结果

Fig. 4 Results of reading performance of data interface

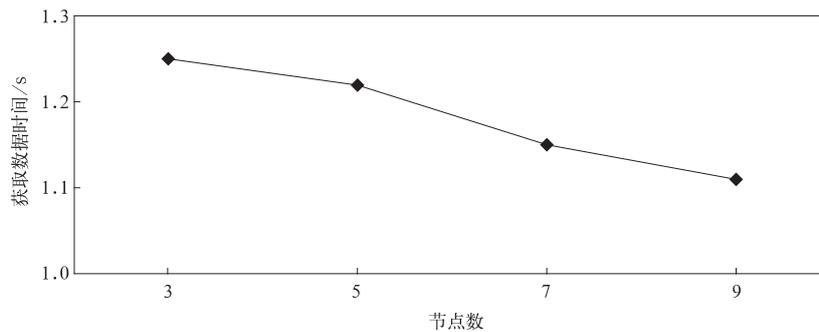


图 5 平台扩展性能测试结果

Fig. 5 Results of performance of platform extension

## 5 小 结

本文对基于 Hadoop 技术体系的数值预报产品服务台的总体功能、关键技术及其在实际业务中的实例应用进行了详细阐述,该平台能够快速收集各数值预报中心分发的数值预报产品并实现快速的数据解码和数据接入,同时对诸如数值预报产品等非结构化气象数据实现分布式的存储和处理,基于 Rest Web Service 开发的业务应用接口能够高效对接气象业务系统。实际业务测试表明:

1) 基于 Hadoop 技术体系搭建的平台具有较好的系统稳定性和扩展性,可根据实际业务需求对系统进行动态在线扩展。

2) 平台能够高效处理诸如数值预报产品等非结构化气象数据,Hadoop 与 HBase 技术组合是非结构化气象数据分析和处理技术的重要发展方向。

随着平台在气象业务中的推广应用,该平台将在气象业务现代化和一体化建设中发挥重要作用。当然,目前平台仍需要在实际应用中逐步优化,包括平台的可视化管理开发、可接入资料种类的扩展并进一步简化数据服务接口等<sup>[14-16]</sup>。另外,实时计算概念和技术的引入也是未来重要的研究方向。

## 参 考 文 献

- [1] 李集明,沈文海,王国复. 气象信息共享平台及其关键技术研究. 应用气象学报,2006,17(5):621-622.
- [2] 王伯民,李集明,吴增祥. 我国气象科学数据发布策略初步研究. 应用气象学报,2004,15(增刊 D):153-154.
- [3] 宋连春,肖风劲,李威. 我国现代化气候业务现状及未来发展趋势. 应用气象学报,2013,24(5):513-520.
- [4] Randriamampianina R. Radiance-bias correction for a limited area model. *Quarterly Journal of the Hungarian Meteorological Service*,2005(3):636.
- [5] 王萍,刘颖,王汉芝,等. 基于格点场数据的沙尘暴双预报模型. 天津大学学报,2006,3(25):329-330.
- [6] 吴焕萍,张永强,孙家民,等. 气候信息交互显示与分析平台(CIPAS)设计与实现. 应用气象学报,2013,24(5):631-640.
- [7] 王涵,李玲,孙学庆,等. CMACast 内蒙古气象信息 Web 平台的设计与实现. 气象科技,2014,42(2):283-286.
- [8] Aguilera M K, Merchant A, Shah M, et al. Sinfonia: A New Paradigm for Building Scalable Distributed Systems. Proc of the 21st ACM Symp on Operating Systems Principles,2007:159-174.
- [9] 杨锋,吴华瑞,朱华瑞,等. 基于 Hadoop 的海量农业数据资源管理平台. 计算机工程,2011,37(12):243-244.
- [10] 林春泽,智协飞,韩艳,等. 基于 TIGGE 资料的地面气温多模式超级集合预报. 应用气象学报,2009,20(6):706-712.
- [11] 王彬,肖文名,李永生,等. 华南区域中心计算资源管理系统的建立与应用. 气象,2011,37(6):764-770.
- [12] 王彬,常飏,朱江,等. 气象计算网格平台资源监视模块的设计与实现. 应用气象学报,2009,20(5):642-648.
- [13] 于重重,商利利,谭励,等. 半监督学习在不平衡样本集分类中的应用研究. 计算机应用研究,2013,30(4):1085-1089.
- [14] Ma T, Hempel M, Peng D M, et al. A survey of energy-efficient compression and communication techniques for multimedia in resource constrained systems. *IEEE Communications Surveys & Tutorials*,2012,14:1-10.
- [15] 应毅,任凯,曹阳. 基于改进的 MapReduce 模型的 Web 挖掘. 科学技术与工程,2013,5:78-80.
- [16] 任结,周余,于耀,等. 基于 ORB 自然特征的 AR 实时系统实现. 计算机应用研究,2012,29(9):3594-3596.

## Design and Implementation of NWP Data Service Platform Based on Hadoop Framework

Li Yongsheng Zeng Qin Xu Meihong Shi Xiaoying

(Guangdong Provincial Meteorological Information Center, Guangzhou 510080)

### Abstract

As the numerical weather prediction (NWP) products increase in huge amounts every day, traditional relational database has the problem of low efficiency in archiving capacity and management, while file based storage faces performance challenges in long-time-series data accessing and massive computation of spatial-temporal data. Therefore, a three-tier software framework is designed, which implements distributed data storage model, parallel data access service and distributed computation for frequently used statistical algorithms based on Hadoop framework.

Meteorological big data such as NWP products, radar 3D mosaic and satellite remote sensing are designed to be composed of metadata and data entity, which both are stored in Hbase data tables, and managed with HDFS file system. Metadata are defined by variable name, dimension, latitude, longitude, altitude and lead time etc., and data entity consists of row key, time stamp and column family to store the value at each grid point. A Rest (representational state transfer) Web Service is setup for direct NWP data acquisition, field data clipping and location based time-series accessing. File download services in "MICAPS", "surfer" and "json" format are also ready for the third-party meteorological software. System testing for data access of CHAF model shows that it costs only 12 seconds to write in 1000 NWP data fields each with 82503 grid points, and less than 4 seconds to read out the same amount of data from the distributed databases.

Map-reduce scheme are implemented for computation of meteorological algorithms, e. g., Kalman filter and successive regression. Most of meteorological statistical algorithms are time independent, which make it possible that a task is divided into small sub-tasks according to data slicing on time series, and assigned to different computational nodes in map programs. Reduce programs are to gather and summarize the result of sub-task computation. With data amount and users increasing, Hadoop framework deployed on several X86 PC servers demonstrates performance advantage over single IBM power system. And flexible hardware architecture from 3 computational nodes to 9 nodes show steady and better data access efficiency with good speed-up ratio, which brings more confidence for practical use in weather forecast.

Operational trial in multi-user environment further shows advantages of this cloud-like computing service over the traditional client-server model in meteorological data mining, such as NWP interpretation and model evaluation.

**Key words:** Hadoop framework; meteorological data; Web Service interface