

# 基于 Hadoop 的智能电网状态监测数据存储研究

刘树仁 宋亚奇 朱永利 王德文

(华北电力大学控制与计算机工程学院 保定 071003)

**摘要** 智能电网需要收集海量设备状态监测数据,这对数据存储与查询提出了更高的要求。为处理这些海量数据,设计并实现了基于 Hadoop 技术的数据存储系统,包括 Hadoop 集群、存储客户端和查询客户端。通过基准测试、存储结果验证和查询性能分析,验证了该系统具有分布式海量存储及高效查询的优势,适合智能电网环境下设备状态监测数据的存储。

**关键词** Hadoop, HDFS, MapReduce, 电力设备, 状态监测, 智能电网

中图分类号 TP391 文献标识码 A

## Research on Data Storage for Smart Grid Condition Monitoring Using Hadoop

LIU Shu-ren SONG Ya-qi ZHU Yong-li WANG De-wen

(School of Control and Computer Engineering, North China Electric Power University, Baoding 071003, China)

**Abstract** With the development of smart grid, the mass data collected for the equipment conditioning requires higher performance of data store and data inquiry. A storage system based on Hadoop for smart grid condition monitoring was designed and achieved. The system consists of three parts: Hadoop cluster, storage client and inquiry client. A quantity of experiments were carried out including benchmark on hadoop cluster, verification for storage effect and analysis on query performance, which validate that the system has the advantages of distributed, mass storage and efficient inquiry, and is suitable for dealing with the mass data in condition monitoring of power equipments in smart grid.

**Keywords** Hadoop, HDFS, MapReduce, Power equipment, Condition monitoring, Smart grid

## 1 引言

随着数字时代的到来,美国、欧盟及中国等相继提出“智能电网”的概念,并大力推进智能电网的研究和发展<sup>[1]</sup>,智能电网成为未来电网发展趋势。智能电网环境下设备状态数据量将巨增,对可靠性和实时性要求更高,远远超出传统电网状态监测的范畴。国内已有电力系统信息平台的建设大多采用常规的解决方案,即基础架构采用价格昂贵的大型服务器,存储硬件采用磁盘阵列,数据库管理软件采用关系数据库系统,紧密耦合类业务应用采用套装软件,导致系统扩展性较差、成本较高,难以适应智能电网对状态监测数据可靠性和实时性的更高要求<sup>[2]</sup>。

Hadoop 是一个开源的云计算模型,具备可靠性高、数据处理量大、灵活可扩展以及使用廉价机器等优势,正成为信息领域研究的热点,其广泛应用于语义分析<sup>[3]</sup>、搜索引擎、机器学习<sup>[4,5]</sup>和生物信息分析等<sup>[7,8]</sup>。从已发表的论文<sup>[7,8]</sup>看,在工业监测领域,尤其是电力设备状态监测领域,云计算尚处于设想和系统的架构设计阶段,未见成型系统的报道。

HDFS(Hadoop Distributed File System)是 Hadoop 上的分布式文件系统,采用 M/S(Master/Slave)架构, Hadoop 集

群上节点可采用廉价 PC。MapReduce 是一种处理海量数据的并行编程模型和计算框架,用于对大规模数据集的并行计算<sup>[6]</sup>,该计算模式在 Hadoop 上得到了实现。

通过上述技术,可利用已有设备搭建分布式、高性能、可伸缩的数据存储系统,从而为构建低成本、高效率的电力数据挖掘与分析系统提供可能性<sup>[7]</sup>。

本文针对智能电网下状态监测数据的特点,设计并实现了基于 Hadoop 平台的数据存储系统。

## 2 基于 Hadoop 的状态监测数据存储系统设计

根据 HDFS 文件系统特点,结合电力设备状态监测数据,构建存储系统。

图 1 所示为存储系统结构与访问关系图,存储系统由 3 部分组成,数据来源为包含大量传感器的 Zigbee 网络或者光纤以太网等其他网络。系统核心为处于中间的 Hadoop 集群,NameNode 是集群管理者,也是客户端访问文件系统的接口,电力设备状态监测数据存储在多台 DataNode 组成的文件系统里;左边为存储客户端,完成数据实时存储功能,放置在变电站;右边为查询客户端,处理海量数据查询,放置在工区或者数据中心。

到稿日期:2012-03-22 返修日期:2012-07-13 本文受中央高校基本科研业务费专项资金项目(10QG22)资助。

刘树仁(1988-),男,硕士生,主要研究方向为计算机应用,E-mail:491837633@qq.com;宋亚奇(1979-),男,博士生,讲师,主要研究方向为网络信息安全、分布式系统;朱永利(1963-),男,教授,博士生导师,主要研究方向为网络化监控及智能信息处理;王德文(1973-),男,副教授,主要研究方向为电力系统自动化和计算机网络。

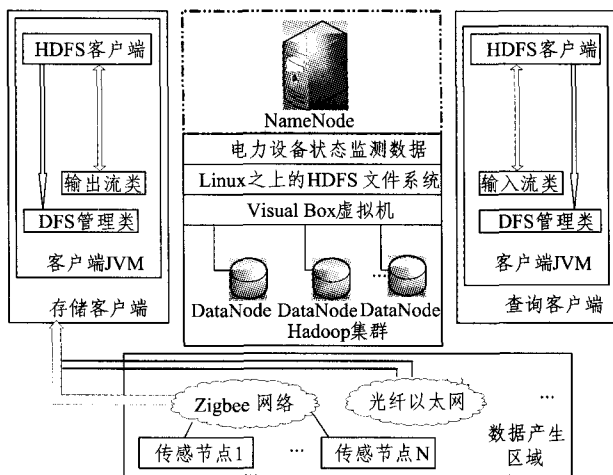


图1 存储系统结构与访问关系图

## 2.1 Hadoop 集群搭建设计

通过虚拟化技术,在安装 Windows 2008 操作系统的 PC 机上,安装 Visual Box,虚拟 Linux 环境,形成一个处于 192.168.0.0 段的内部网络。在各机上安装 JDK、SSH 和 Hadoop,搭建一个完全分布模式下的 Hadoop 集群。

## 2.2 存储客户端设计

智能电网环境下电力设备状态信息多样,有静态信息和动态信息之分。静态信息有用户信息、线路信息、杆塔信息、绝缘子信息、采集设备信息等,特点是占用空间少;动态信息有设备的温度、张力、倾角以及报警信息,其采样频率为几百 KHz,甚至高至 MHz,因此该类数据量的增长十分明显,随着时间增长,其所需存储空间越来越大,并逐渐超越传统数据库处理能力;同样在该类数据查询中,需要处理的数据量也是庞大的。

实验中静态信息仍然存储在关系数据库里,动态信息使用 HDFS 存储及 MapReduce 并行查询。

为模拟实际场景里的数据产生网络,实验中将一台 PC 机当作信息生成发送端。每条报警信息包括产生时间、信息编号、产生通道、设备节点物理地址、报警类型和报警内容,各个信息之间用逗号隔开;信息存储到以设备名和时间组成的文件里,如表 1 所列。

表 1 信息及文件格式

信息格式	20120305100806223,1900,001,00158D0000093145.4C, 11100001010101010
文件格式	T1900-20120305100806

实验中数据接收端也是存储客户端,完成从发送端接收实时数据并上传到 HDFS 的功能。

为提高存储效率和减少硬盘访问次数,数据首先放在内存缓冲区,达到一定条数再存入本地文件;HDFS 被设计为每个文件占用整个数据块时存储效率最高<sup>[9]</sup>,因此本地文件系统将文件大小控制在 64M 左右,再上传到 HDFS,这样每个文件正好占用一个 HDFS 数据块,最大限度地减少了 NameNode 里的元数据数量。

其中文件上传 HDFS 过程如图 2 所示。缓存是通过构建文件数组实现的。

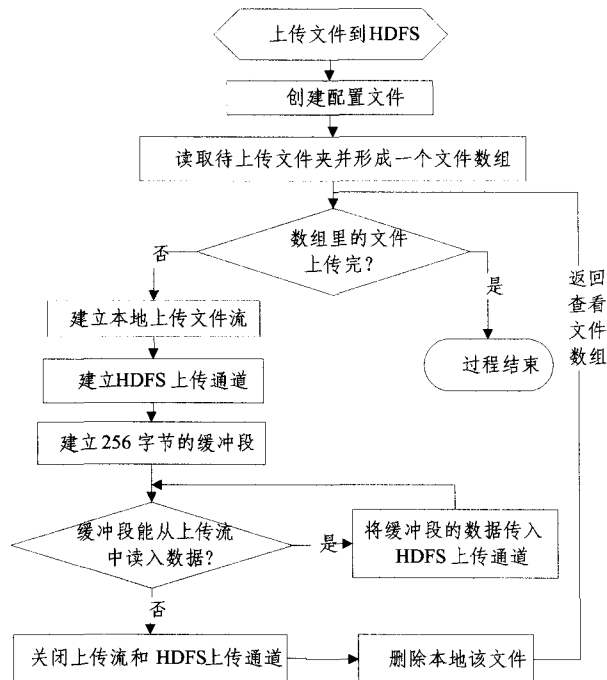


图 2 文件上传 HDFS 函数流程图

## 2.3 查询客户端设计

基于 Hadoop 并行查询里,最主要的就是设计 Map 和 Reduce 函数,包括输入输出键值对类型以及 Map 和 Reduce 函数的具体逻辑<sup>[10]</sup>。

主函数完成用户查询信息和 MapReduce 程序之间的交互,查询需要传入的变量有:待查询文件夹地址、输出结果文件地址、待查询关键词、Reducer 个数、关键词在信息行中所处位置。

实验的查询里,需要通过某一数据段是否满足要求来获取整条数据,并将信息输出到汇总文件,不需对汇总数据再作处理,因此该查询过程只需设计好 Map 函数和主函数。

### 2.3.1 Map 函数

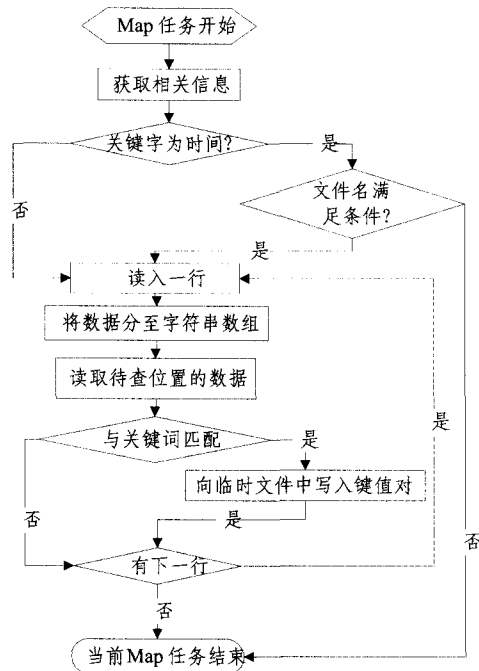


图 3 Map 函数处理过程

按照 Input Splits 划分,每个 Map 函数打开一个 Split,并分别对每行数据进行分析处理。考虑到文件命名里有时间段,为提高效率,当查找关键字是时间时,先判定文件名时间段是否满足关键字,不满足则停止对该文件的查询任务。整体查询过程如图 3 所示。

### 2.3.2 主函数

主函数的主要工作就是设置各种与作业执行有关的参数:

1)待查找的关键词:主函数通过 Configuration 对象传递全局作业参数,任务开始后,将具体信息散发给不同地点的 Map 任务。

2)设置 Map 类:对 Mapper 类进行了重写,接受用户待处理信息并在指定 TaskTracker 里完成任务。

3)设置 Reduce 类:对 Reducer 类进行了重写,读入 Map 函数输入的数据,没有对 Map 函数的输出值作出其他处理,只是将所有数据按照键排序后输出到结果文件里。

4)设置 Reducer 个数:应控制该数据小于待查文件夹下的文件数目和 DataNode 节点数两倍的最小值,如式(1)所示:

$$Reducers < \text{Min}\{FileNo, DataNodeNo \times 2\} \quad (1)$$

5)输出的类定义:定义为文本类型(Text),同时适用于 Map 和 Reduce 函数的输出的键值。

6)输入输出的格式:实验中,无论待查数据还是查出的结果数据,都是以文本的形式存在 HDFS 系统里,因此,使用输入输出的格式均为 TextInputFormat 所定义的格式。

## 3 实验与结果分析

### 3.1 实验环境及 HDFS 的 I/O 基准测试

文中所有的实验都是在实验室搭建的 Hadoop 平台上运行的。平台由 19 个节点组成,每个节点物理机配置为四核 CPU,主频 2.60Hz,内存 4G,网络带宽 100Mbps 以太网。NameNode 虚拟机配置为单核 CPU,2.5G 内存,硬盘 100G;18 个 DataNode 虚拟机配置为单核 CPU,2G 内存,硬盘 100G,Hadoop 版本 0.20.2,文件备份为 2。

在集群无其他任务运行的条件下,使用 TestDFSIO 基准测试程序测试 HDFS 的 I/O 性能。分别改变文件大小、文件数量,获取运行时间。

1)进行 10 个文件的写操作,文件大小从 10M 到 800M 进行变化。图 4 绘制了任务运行时间和平均运行时间。

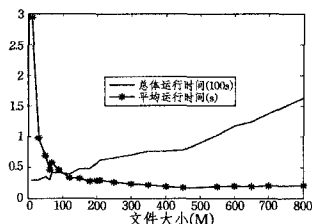


图 4 文件大小变化对运行时间的影响(read)

由图可知,随着文件大小的增加,总体运行时间呈现增长的状态,但相应的平均处理时间(每写 1M 文件的时间)一直下降。该组试验中,读文件的速率总是大于写文件的速率,但是其运行时间的波形图与写文件相似。

2)进行一组文件大小为 100M 的写操作,文件数量从 1 个到 50 个。表 2 列出部分数据,图 5 为任务运行时间和平均运行时间图。

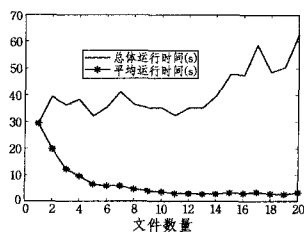


图 5 文件数量变化对运行时间的影响(read)

表 2 节点数 18 时 HDFS 读写时间

测试	文件个数	文件大小 (MB)	HDFS 字节数	Map 任务个数	运行时间 (S)
Write	5	100	500	5	61.898
Write	10	100	1000	10	59.616
Write	20	100	2000	20	69.727
Write	30	100	3000	30	79.822
Write	50	100	5000	50	123.647
Read	5	100	500	5	35.71
Read	10	100	1000	10	38.555
Read	20	100	2000	20	54.517
Read	30	100	3000	30	70.696
Read	50	100	5000	50	84.964

综合图表可知,随文件数量增加,平均处理时间(每写一个 100M 文件所花的时间)一直下降。该组实验中,读文件的运行时间波形图同样与写文件相似。

3)控制文件总量大小为 5000MB,文件数量从 100 个到 1000 个(文件大小从 50MB 到 5)进行变化,图 6 示出文件总量不变时文件数量与运行时间的关系。

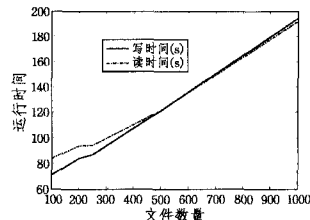


图 6 文件总量不变时文件数量与运行时间关系图

图 6 中,写操作和读操作运行时间曲线逐渐增长,这是因为当文件数量持续增大时,处理的数据为大量小文件,占用大量 NameNode 内存,同时分配任务花费的时间变得十分庞大。

因此,1)、2)两组实验中,从读写所花的平均时间来看,实验中所建的集群很适合于处理大规模数据读写;3)组实验中,说明集群并不适合处理小文件。综上,该集群适合处理大规模大文件的数据。

### 3.2 数据存储

```

NameNode 'NameNode:9000'
-----
Started:   Mon Mar 05 14:46:46 CST 2012
Version:   0.20.2, 1911707
Compiled:  Fri Feb 19 08:07:34 UTC 2010 by chrisko
Upgrades:  There are no upgrades in progress.

Browse the Hadoop
NameNode Logs
Go back to DFS home

Live Datanodes: 18
-----


| Node       | Last Contact | Admin State | Configured Capacity (GB) | Used (GB) |
|------------|--------------|-------------|--------------------------|-----------|
| DataNode07 | 2            | In Service  | 94.38                    | 1.29      |
| DataNode03 | 0            | In Service  | 94.38                    | 1.2       |
| DataNode13 | 2            | In Service  | 94.38                    | 1.15      |
| DataNode02 | 1            | In Service  | 94.38                    | 1.05      |
| DataNode15 | 2            | In Service  | 94.38                    | 1.03      |
| DataNode09 | 2            | In Service  | 94.38                    | 1.03      |
| DataNode10 | 1            | In Service  | 94.38                    | 0.98      |


```

图 7 HDFS 的 HTTP 接口显示的文件分布情况

启动平台,数据发送端运行 10 个线程,每个线程每秒产生 90 条数据(约 16kB)。运行数据上传端,上传频率为 1 小

时,每个上传文件约 60M。运行几小时后,在浏览器上通过 50070 端口查看文件分布情况,如图 7 所示。

从图 7 中可看到,文件被均匀地上传到集群各个节点,集群运行状况良好。查看文件系统里上传成功的文件,没有出现数据丢包现象,存储过程成功完成。

### 3.3 数据查询性能分析

调节数据发送端的发送频率,产生文件大小和数量不同的几组数据。启动查询客户端,查询各组数据。将同样的数据放到 Oracle 系统(四核 CPU,10 内存,硬盘 438G)里查询并统计时间。为减少偶然因素的影响,每个数据都是经 3 次以上的测试所得的平均值<sup>[11,12]</sup>。表 3 将两种情况进行对比。

表 3 查询时间对比情况

查询条数	Oracle(S)	Hadoop 集群(S)
5.4 万	0.8	10.4
9.7 万	5.3	12.5
31.7 万	14.3	20.2
63.5 万	15.6	25.5
127 万	20.4	29.3
253.8 万	56.3	55.1
507.7 万	113.5	85.1
1015.3 万	201.4	150.2

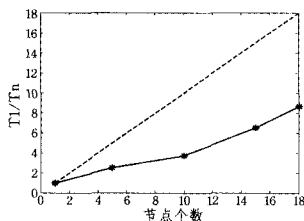


图 8 加速比性能测试结果

从表 3 中可以看出,当数据量低于 250 万条时,单机版系统有很大的优势;随着数据量增大,集群的查询优势越来越明显。这是由于与单机版相比,集群有额外的任务分配及调度时间,当查询数据量较少时,这个过程所用的时间是不可忽略的;当数据量非常庞大时,这部分所用的时间跟总体用时相比是非常微小的。

为测试查询算法的并行性,测试 100 万条数据集的加速比(speedup),实验结果如图 8 所示。从图中可看出,查询算法的加速比是接近线性的,考虑到 100 万条数据对于集群来说并不是很大的数据量,集群性能没有得到全面的体现,因此

图中曲线并没有靠近对角线。这再次说明平台更适合大文件大数据量的存储和查询。

**结束语** 针对智能电网下状态监测数据的海量、分布式等特点,设计并实现了智能电网状态监测数据的 Hadoop 存储系统。搭建具有 19 个普通 PC 机的 Hadoop 集群,对集群进行基准测试,表明集群适合处理大规模大文件的数据;应用 HDFS 设计实现了智能电网状态监测数据的分布式存储,文件均匀分布于集群各个节点,集群运行状况良好,没有出现数据丢失现象;应用 MapReduce 模型实现了状态监测数据的并行查询以及与单机关系数据库查询性能的比较。随着数据量的增大,集群的查询优势更加明显,因此它可以有效地应用于智能电网信息平台建设中。

### 参考文献

- [1] 毕天姝,素梅智. 智能电网含义及共性技术探讨[J]. 华北电力学报,2011,38(2):1-9
- [2] 张怀宇,朱松林,张杨,等. 输变电设备状态检修技术体系研究与实践[J]. 电网技术,2009,33(13):70-73
- [3] Newman A, Li Y F, Hunter J. Scalable Semantics-the Silver Lining of Cloud [C] // IEEE Forth International Conference on eScience, 2008
- [4] 于戈,谷峪,鲍玉斌,等. 云计算环境下的大规模图数据处理技术[J]. 计算机学报,2011,10(34):1753-1767
- [5] Gaggero M, Leo S, Mance S. Parallelizing bioinformatics applications with MapReduce[OL]. www.cca08.org/papers/Poster10-Simone-Leo.pdf
- [6] 潘巍,李战怀,伍赛,等. 基于消息传递机制的 MapReduce 图算法研究[J]. 计算机学报,2011,10(34):1768-1784
- [7] 王德文,宋亚奇,朱永利. 基于云计算的智能电网信息平台[J]. 电力系统自动化,2010,34(22):7-12
- [8] 吕跃刚,关晓慧,刘俊承. 风力发电机组状态监测系统研究[J]. 自动化与仪表,2012,27(1):6-10
- [9] White T. Hadoop 权威指南(第二版)[M]. 北京:清华大学出版社,2011:43-44
- [10] 赵卫中,马慧芳,傅燕翔,等. 基于云计算平台 Hadoop 的并行 k-means 聚类算法设计研究[J]. 计算机科学,2011,38(10):166-169
- [11] 刘鹏,黄宜华,陈卫卫. 实战 Hadoop[M]. 北京:电子工业出版社,2011:60-64
- [12] 陈勇. 基于 Hadoop 平台的通信数据分布式查询算法的设计与实现[D]. 北京:北京交通大学,2009

(上接第 58 页)

- [4] Cheng H, Guo W, Yu J D. Multichannel convolution blind separation algorithm for MIMO DSSS/CDMA system [J]. IEEE Trans. on Circuits Systems Signal Processing, 2007, 26(2):249-262
- [5] Zhang Yang, Kassam S A. Blind source separation and equalization in asynchronous MIMO DS-CDMA [C] // IEEE CISS. Princeton NJ: IEEE Press, 2008: 316-319
- [6] Raju K. Blind Source Separation For Interference Cancellation In CDMA Systems [D]. Finland: Helsinki University of Technology, 2006
- [7] Huovinen T. Independent component analysis in DS-CDMA multiuser detection and interference cancellation [D]. Finland: Tampere University of Technology, 2008
- [8] Nion D, Lathauwer L D. A block component model-based blind

- DS-CDMA receiver [J]. IEEE Trans. on Signal Processing, 2008, 56(11):5567-5579
- [9] Liu X, Xu Z Z, Lei L. Space-time blind multiuser detection for multiuser DS-CDMA and oversampled systems [R]. German: Springer Wireless Pers Commun, 2010
- [10] Fevotte C, Theis F J. Orthonormal approximate joint block-diagonalization [R]. Technical report GET/Telecom. Paris, 2007
- [11] Caiafa C F, Proto A N. Separation of statistically dependent sources using an L2-distance non-Gaussianity measure [J]. Signal Processing, 2006, 86:3404-3420
- [12] Novey M, Adali T. Complex ICA by Negentropy Maximization [J]. IEEE Trans. on Neural Networks, 2008, 19(4):596-609
- [13] Friedman J H. Exploratory Projection Pursuit [J]. J. Amer. Statist. Assoc., 1987, 82:249-266