

7-2-2020

Approach to Process Smart Grid Time-Serial Big Data Based on HBase

Wang Yuan

1. *China Realtime Database Co., Ltd, State Grid Electric Power Institute, Nanjing 210012, China;;*

Tao Ye

2. *Department of Computer Science & Technology, Nanjing University, Nanjing 210046, China;*

Yuan Jun

1. *China Realtime Database Co., Ltd, State Grid Electric Power Institute, Nanjing 210012, China;;*

He Wei

1. *China Realtime Database Co., Ltd, State Grid Electric Power Institute, Nanjing 210012, China;;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Approach to Process Smart Grid Time-Serial Big Data Based on HBase

Abstract

Abstract: With the development of critical theories and technologies in Internet of things (IOT), more and more attentions have been focused on the IOT applications. Smart Grid is one of the typical IOT applications on which a huge number of sensors have been deployed to gather and generate time-serial data to make sense of the running states of the key devices. How to apply these data to make smart grid running secure and stable is a hot research topic. *By considering the fact that smart grid is characterized by a huge number of devices, a huge amount of data and the high speed data generating, an approach was proposed to store and process big time-serial data in smart grid based on the HBase. The emphasis is a policy-driven method to make the time-serial data organized in HBase on demand. The HBase cluster was deployed to verify effectiveness of the approach.*

Keywords

Big data, Internet of things, HBase, Time-serial data, performance optimization

Recommended Citation

Wang Yuan, Tao Ye, Yuan Jun, He Wei. Approach to Process Smart Grid Time-Serial Big Data Based on HBase[J]. Journal of System Simulation, 2016, 28(3): 559-568.

一种基于 HBase 的智能电网时序大数据处理方法

王远¹, 陶焯², 袁军¹, 何卫¹

(1. 国网电科院江苏瑞中数据股份有限公司, 江苏 南京 210012; 2. 南京大学 计算机科学与技术系, 江苏 南京 210046)

摘要: 随着物联网关键技术与理论的发展, 物联网应用受到了越来越多的关注。智能电网是一类典型的物联网应用, 遍布全网的传感器收集及产生了大量反映关键设备运行状态的时序数据。如何利用时序数据确保智能电网的安全以及稳定运行是当前的研究热点之一。针对智能电网时序数据设备多、数据规模大、产生速度快等特点, 提出了一种基于 HBase 的海量时序数据存储处理方法, 着重介绍了如何利用策略驱动技术实现时序数据的灵活存储与处理。通过构建 HBase 集群, 验证了该方法的有效性。

关键词: 大数据; 物联网; HBase; 时序数据; 性能优化

中图分类号: TP274+.2 文献标识码: A 文章编号: 1004-731X (2016) 03-0559-10

Approach to Process Smart Grid Time-Serial Big Data Based on HBase

Wang Yuan¹, Tao Ye², Yuan Jun¹, He Wei¹

(1. China Realtime Database Co., Ltd, State Grid Electric Power Institute, Nanjing 210012, China;

2. Department of Computer Science & Technology, Nanjing University, Nanjing 210046, China)

Abstract: With the development of critical theories and technologies in Internet of things (IOT), more and more attentions have been focused on the IOT applications. Smart Grid is one of the typical IOT applications on which a huge number of sensors have been deployed to gather and generate time-serial data to make sense of the running states of the key devices. How to apply these data to make smart grid running secure and stable is a hot research topic. *By considering the fact that smart grid is characterized by a huge number of devices, a huge amount of data and the high speed data generating, an approach was proposed to store and process big time-serial data in smart grid based on the HBase. The emphasis is a policy-driven method to make the time-serial data organized in HBase on demand. The HBase cluster was deployed to verify effectiveness of the approach.*

Keywords: Big data; Internet of things; HBase; Time-serial data; performance optimization

引言

随着“物联网”时代的来临, 新一代 IT 技术将被充分地运用在各行各业之中。传感器作为其基本设施, 被广泛部署在各行业中, 特别是在电力、交

通、石油、石化、航空、国防、水情监测、水利、冶金、环保等面向工业过程控制重要行业, 大量地使用传感器设备对重点设备、重点部位、重点场所进行监测。

智能电网作为一类典型的物联网, 具备数据密集型特征^[1-2]。海量时序数据已成功地应用于调度自动化、配电自动化、海量历史/实时数据管理平台以及发电等多个智能电网应用系统的环节中。但是, 在时序数据的采集、传输、存储和处理过程中面临如下挑战。



收稿日期: 2014-10-17 修回日期: 2014-11-05;
基金项目: 国家 863 计划(2011AA05A116); 国家电网科技项目(62KJ130501B1003620120);
作者简介: 王远(1980-), 男, 山东青岛, 博士, 高工, 研究方向为大规模并行处理技术、海量数据管理和软件体系结构; 陶焯(1981-), 女, 江苏常州, 硕士, 讲师, 研究方向为并行计算与分布式系统。

<http://www.china-simulation.com>

超大规模数据集。智能电网建设过程中汇集了电网生产运行所产生的各类数据,随着智能电网建设的不断完善,其测点的规模越来越大,预计将达到千万级以上,并且单个测点将以分钟级甚至秒级的频率产生数据。特别在针对智能电网构建集中式数据中心时,对数据的处理能力、处理速度提出了更高的要求^[3-6]。此外,随着互联网、传感器网络的发展,数据量近年来几乎成倍增长^[7]。应对大数据量是急需面对的挑战。

实时数据处理需求。随着智能电网的建设,包括 WAMS、调度自动化、配电自动化在内的诸多电力自动化系统对数据处理实时性的需求日益迫切,以 WAMS 系统为例,每秒需要处理的时序数据记录数可达到一千万,常规的数据处理方式根本无法应对如此高速的时序数据处理挑战,电网大数据对时序数据的处理规模、处理速度提出了更高的要求。

时序数据处理方式多样。利用智能电网大数据构建数据中心为智能电网的安全以及稳定运行提供数据支持,用于进行辅助决策、精细化运营及智

能战略规划等高端应用。不同的业务应用,对数据处理需求也不一样,目前比较缺乏统一的针对时序数据的存储管理方法,用于满足多变的应用需求,特别是跨部门、跨专业的业务应用。

智能电网海量时序数据作为一类典型的物联网大数据,具备潜在反映智能电网当前运行状态与未来发展趋势的能力。基于智能电网时序数据做深度挖掘,从而保证电网安全、稳定、可靠的运行是当前的研究热点之一^[8]。然而,要发挥海量时序数据的作用,高效的时序数据存储组织是必备前提之一。本文提出一种基于 HBase^[9]的智能电网时序数据处理方法,采用策略驱动的方式,按需存储组织海量时序数据,从而满足智能电网超大规模测点、实时数据处理以及灵活多变的数据访问等需求。

1 相关工作

物联网中的时序数据一般是指由设备/仪表产生,由传感器进行收集,与某一对象/设备具体相关,在时间上前后关联性的一类数据,如图 1 所示,电表、电压、电流就是典型的时序数据。

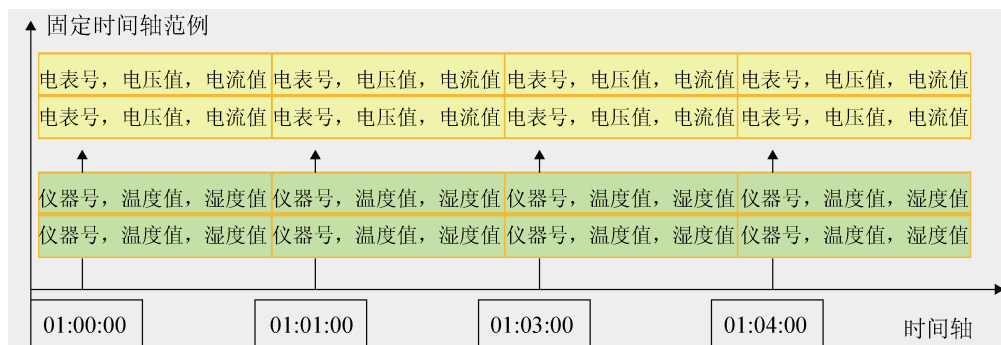


图 1 时序数据示意

时序数据记录一般可定义为三元组:

$\langle \text{DeviceID}, \text{TimeStamp}, \text{Value} \rangle$

其中: DeviceID 为产生该条数据的物理设备/传感器; TimeStamp 为记录产生的时间戳; Value 为具体的值。

1.1 利用关系型数据库处理时序数据

在目前智能电网相关应用系统中,一般使用

oracle, DB2 等关系型数据库处理时序数据。然而,传统的实体关系模型在应对时序数据时存在如下问题:

1) 数据存储孤立分离。按照实体关系模型,一条时序数据记录被保存为单独的一行,因此连续时间段内的数据被孤立分离地存储在数据库中。在大数据时代下,时序数据的数据量超出想象,若采

用关系型数据库存储时序数据, 在应对与时间相关的一批数据的访问需求时, 将对数据库造成极大压力, 甚至无法完成。

2) 数据的访问速度与数据量的大小成反比。随着数据量的增加, 访问速度越来越慢。同时, 大多数关系型数据库为了提升查询性能, 针对大量数据建立索引, 由此消耗大量的系统资源。

3) 无法满足时序数据的实时处理需求。实时产生的时序数据必须加载到数据库后, 再通过数据库访问。对于海量时序数据, 关系型数据库无法高速的加载并满足处理需求。

1.2 利用实时数据库进行处理

针对时序数据, 产生一类实时数据库^[10-13], 最典型的为 PI 数据库、eDNA 数据库与 Informix 数据库。然而实时数据库在应对智能电网大数据时存在如下问题:

1) 无法支持大规模测点管理。对于国家级或省级智能电网而言, 需要支持上亿甚至上十亿的测点规模。然而, 无论 PI, eDNA 还是 Informix, 均无法为大规模测点管理提供支撑。

2) 扩展能力不足。当前的实时数据库一般只支持单机部署, 无法集群部署, 无法支撑大规模数据应用。

3) 高可用与数据安全性不足。实时数据库一般不具备高可用特性, 对于所存储的数据的安全性, 也完全依赖于硬件与数据库软件本身, 不支持数据冗余备份, 无法应对大数据分析场景。

1.3 基于 Hadoop 处理物联网时序数据

作为最为热门的大数据的开源技术生态体系, Hadoop 目前被广泛应用, 其核心技术为 MapReduce 计算模型、HDFS 文件系统, 以及分布式列式数据库 HBase^[14-16]。Hadoop 体系架构面向普通硬件组成的大规模集群, 扩展能力得益于其 share-nothing 结构和较强的软件级容错能力。Hadoop 卓越的扩展能力已在工业界(Google, Facebook、淘宝、京东等)得到了充分验证。

HBase 作为 Bigtable^[18]在 Hadoop 体系中的开源实现, 为海量数据提供了面向列簇的存储支撑以及良好的在线应用支撑。与传统关系数据库不同, HBase 提供了一种灵活的、自定义的、面向列簇的数据组织模式, 对于实时在线应用具备良好的支撑能力。其数据模型如图 2 所示。

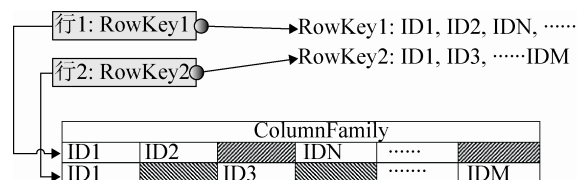


图 2 HBase 数据模型

对于 HBase 中的数据, 应用程序可通过 RowKey 直接定位获取, 也可通过扫描的方式按定制的模式进行检索。同一张数据表中不同的数据行可具备不同的列, 在实际的存储中, 采用稀疏矩阵的方式进行存储, 确保不存在的列不占用存储空间。这种灵活、松散的存储结构非常适合时序数据无法保证定时采集、可靠传输的特征。同时, HBase 基于 MemStore 和 HStore 的分级存储机制, 可较好的满足实时在线处理相关需求^[14]。

OpenTSDB^[18]是基于 HBase 的开源时序数据库, 它利用 HBase 支撑底层存储, 设计了适合时序数据访问的数据模型, 如图 3 所示。OpenTSDB 由指标、时间戳以及标签对列表组成。指标可由用户自定义, 任何用户感兴趣的对象或量测值均可定义为指标。时间戳标识指标收集/产生的时间点。标签对列表主要用于对指标进行扩展, 保证应用的灵活性。OpenTSDB 中的一行数据表示了一条时序数据链, 每一列中保存了指标在特定时刻的取值, 列标识符为该取值具体发生时间相对于指定时间戳的偏移。OpenTSDB 将从指定时间点(时间戳)开始、指定时间间隔(如 1 h)内的指标时序数据保存到一行中, 从而提升时序数据的存取效率。

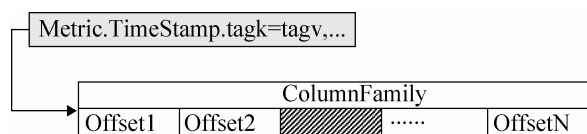


图 3 OpenTSDB 数据模型

但是,智能电网相关应用对于时序数据的存取访问方式主要有 2 种:

- 1) 获取指定设备一段时间内的相关量测值,如查询指定电表在一天之内的有功电能量。
- 2) 获取特定时间点特定区域内所有设备的相关量测值。

从时间的维度考虑,前者称为折线查询,后者称为断面查询。OpenTSDB 可以很好地处理前者,对于后者的支持则不够高效。分析其原因就在于,OpenTSDB 的数据存储组织方式与折线查询模式非常吻合:每一行记录都代表了一个定长“线段”,查询的速度与折线的长度成正比。而断面查询则与 OpenTSDB 的数据组织方式存在较大差异。

2 策略驱动的时序数据存储方法

本节针对 OpenTSDB 的不足,提出一种策略驱动的时序数据存储方法,其核心思想是根据应用系统访问数据的方式来存储和组织数据,每一类策略(Policy)都描述了一类时序数据组织方式,应用系统依据策略来存储时序数据。

要定义策略,必须先分析应用时序数据的访问模式。时序数据访问模式可表示为 3 部分:对象、时间戳和特征集。

1) 对于折线查询,对象为监测的设备或关注的指标,特征集可标识为 $FSet = \langle (t_1, v_1), (t_2, v_2), \dots, (t_n, v_n) \rangle$, 其中 (t_1, t_2, \dots, t_n) 为特定的时间序列,使用基于时间戳的偏移表示, (v_1, v_2, \dots, v_n) 为与时间序列相对应的量测值。

2) 对于断面查询,对象为关注的区域/逻辑对象(如街道/生活区/电厂),特征集为可标识为 $FSet = \langle (d_1, v_1), (d_2, v_2), \dots, (d_n, v_n) \rangle$, 其中 (d_1, d_2, \dots, d_n) 为区域/逻辑对象所包含的设备/测点, (v_1, v_2, \dots, v_n) 为对应的设备/测点在指定时间戳的取值。

3) 特征集由与特定对象相关的时序数据按照特定规则构成。对于任何一类应用,都需要将时序数据组织成与访问模式相一致的特征集,才能提高应用程序访问时序数据的效率。综上所述,策略

(Policy)可定义为如下四元组: $\langle DataFormat, RKPattern, Mapping, RowLength \rangle$ 其中:

- 1) DataFormat 标识设备/测点采集的时序数据,形如 $\langle DeviceID, TimeStamp, Value \rangle$ 。
- 2) RKPattern 定义了 RowKey 的模式,由 3 部分组成:测量对象、时间戳模式和量测值描述,记为 $\langle Obj, TSP, Metric \rangle$ 。Obj 为指定对象,定义为 DeviceID 的集合, TSP 为时间戳模式,可精确至微秒, Metric 为对象相关的量测值描述,采用名值对集合的形式描述。
- 3) Mapping 则定义了时序数据(DataFormat)与特征集之间的映射关系,特征集对应为 HBase 表中具体的列簇。
- 4) RowLength 定义了特征集的最大长度,即 HBase 中一行的最大列数量。

在具体的策略解析中,同一个对象(Obj)符合时间戳模式的所有时序数据在不超过 RowLength 限制的前提下,将被组织为一行数据,存储在 HBase 中。

为满足折线查询和断面查询两种应用需求,本文定义两类策略: Line-Policy 和 Polygon-Policy。具体如下:

Line-Policy:

```
{
  DataFormat=<ID,TS,Float>
  RKPattern=<S1(ID),Hour(TS),Metric>
  Mapping=
  <(DataFormat.TS-Hour(TS),CF:qualifier),
  (DataFormat.value,value)>
  RowLength=60
}
```

Polygon-Policy:

```
{
  DataFormat=<ID,TS,Float>
  RKPattern=<Sn(ID),Min(TS),Metric>
  Mapping=
  <(DataFormat.ID,CF:qualifier),
  (DataFormat.value,value)>
  RowLength=300
}
```

两种策略对应的具体数据存储方式如图 4 所示。图 4 是基于智能电网用电信息采集场景的时序数据存储结构。在智能电网的建设过程中采取“一

户一表”的方式, 每块电表将采集十类数据(表 1 所示)。每一类数据均依据存储策略分类存储。

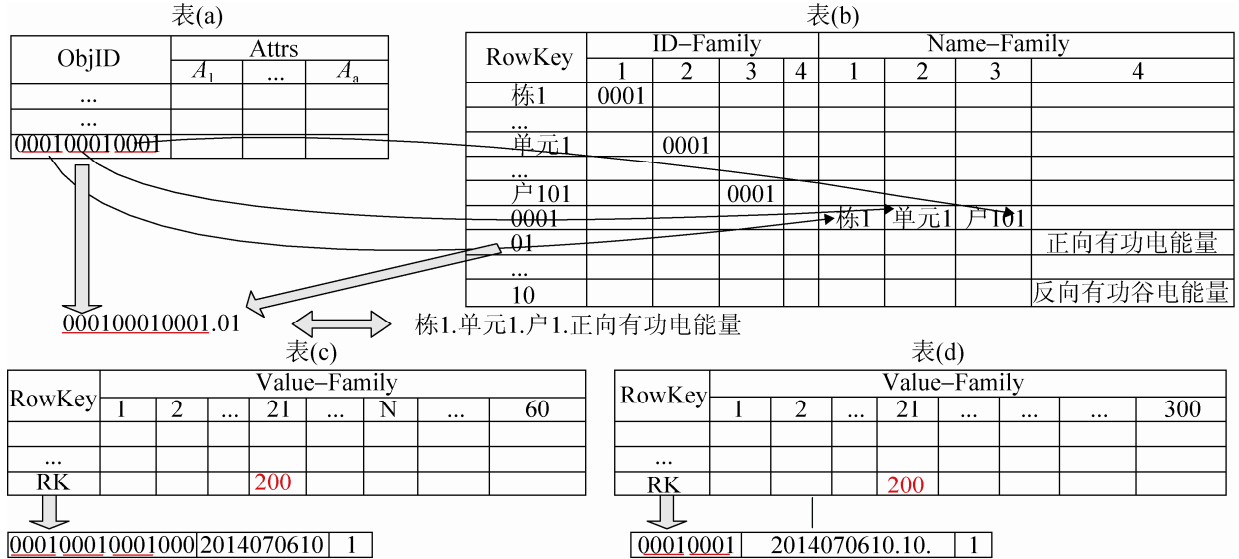


图 4 时序数据存储结构

表 1 智能电表量测值

智能电表属性	测量值数据类型
正向有功电能量	浮点型
反向有功电能量	浮点型
正向有功尖电能量	浮点型
正向有功峰电能量	浮点型
正向有功平电能量	浮点型
正向有功谷电能量	浮点型
反向有功尖电能量	浮点型
反向有功峰电能量	浮点型
反向有功平电能量	浮点型
反向有功谷电能量	浮点型

图 4 中, 表(a)为对象表, 定义了系统所关注的对象及其属性, 对象由 objID 唯一标识。objID 由 6 个字段标识, 分为 3 段(每段 2 字节), 分别对应楼宇、单元与住户。表(b)为映射表, 定义了 objID 与名称之间的对应关系, 方便应用按名存取时序数据。表(c)和表(d)则表示了正向有功电能量对应于两类存储策略的具体存储方式。表(c)表示 line-policy 所对应的存储结构, 它将 1 栋 1 单元 101 户电表的正向有功电量按小时分割, 每小时的数据存储在同一行上, 其中列标识符为相对于 RK 所示

时间的偏移, 其中 200 表示: 2014-07-06T10:21:00 1 栋 1 单元 101 户的正向有功电能量为 200 W; 表(d)表示 polygon-policy 所对应的存储结构, 它将 1 栋 1 单元 300 户电表每分钟所采集的正向有功电量存储到一行上, 其中 200 表示: 2014-07-06T10:10 1 栋 1 单元编号为 21 的住户的正向有功电能量为 200 W。图 4 所示的数据存储结构将智能电网用电信息数据按存储策略分类存储, 分别应对折线查询与断面查询。

3 时序数据处理系统的设计与实现

本节设计并实现了一个策略驱动的时序数据处理系统, 如图 5 所示。系统实现时重点考虑了以下要素:

1) 数据加载时的负载均衡。对于智能电网时序数据而言, 在固定的时间间隔中会产生大量的时序数据, 无论是智能电表、变电站设备还是变压器、开关等通用设备, 它们都将按照固定的时间间隔产生时序数据。因此, 如何在限定的时间内完成数据加载是需要重点考虑的问题。

2) 策略驱动的数据组织方式。利用时间序列数据引擎解析并执行策略，完成数据的存储与访问，并为上层应用提供统一的接口。

3) 高扩展性支持。无论是数据预处理还是数据加载，都要考虑系统的横向扩展能力。对于不同的数据类型、应用系统而言，对扩展性有着不同的需求。

系统在 OpenTSDB 架构的基础上，对其负载均衡能力进行了扩充，增加连接负载均衡器，并重写了 TSD 引擎，使其支持策略驱动的数据管理方式。系统架构分为 4 个层次，至下而上分别为存储层、引擎层、链路层以及客户端。

1) 存储层。存储层基于 HBase 构建，为引擎层提供基础的分布式数据存储服务，从而使引擎能够高效的处理时间序列数据。每一个存储策略都在存储层中对应为 3 类表：对象表、映射表和时序数据表(如图 4 所示)。

2) 引擎层。时间序列数据库(Time-Serial Database, TSD)引擎为在 HBase 上设计的时间序列数据处理引擎。其主要功能为存储策略的解析与执行。具体表现为：数据加载时，接受采集器传送的时间序列数据，按照对应的存储策略进行预加工，然后传送到对应的数据加载模块，由数据加载模块加载至存储层。数据查询时，接收客户端发送的数

据插入、查询、删除请求，依据存储策略进行请求解析后，访问具体的数据表，并返回结果。TSD 内部实现如图 6 所示。TSD 引擎采用分布式设计，支持多个 TSD 引擎并发的处理数据请求，多个 TSD 引擎既可以集中部署也可以部署在集群中的每个节点上。

3) 链路层。链路层主要由一个连接负载均衡器构成。一方面，能够智能的根据 TSD 的连接负载情况，选取合适的连接进行分发，及时收回并销毁过期或无效连接。做到连接的负载均衡化，合理利用集群性能；另一方面，将根据 HBase 中的 Region 分布，对原始数据进行切分，确保属于同一个 RegionServer 的数据落到相同的 TSD 引擎中，降低数据加载中的 I/O 冲突和网络开销。

4) 客户端。客户端主要包括针对各个 servers 的数据采集器，及各种应用平台的数据请求。同时包含一个 B/S 架构的 Web UI 界面，提供基础的数据查询、添加、更新、删除等操作。

在实际部署中，整个系统可以启动任意数量的 TSD 引擎，每个 TSD 引擎之间相互独立，无主从关系且不共享任何资源。TSD 的用户从来不需要去直接访问存储层，而是通过远程登录式的协议(Telnet), HTTP 协议或 RPC 调用去和 TSD 通信，然后由 TSD 完成数据的访问。

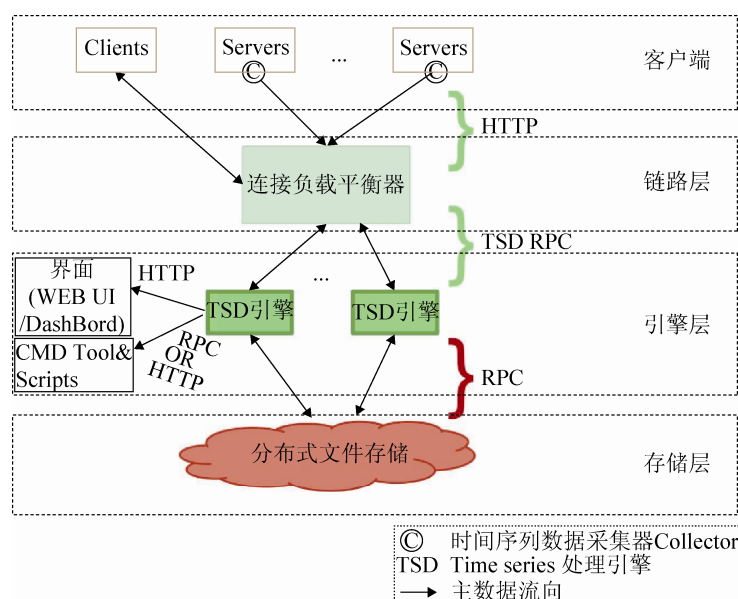


图 5 基于 HBase 的存储系统架构

<http://www.china-simulation.com>

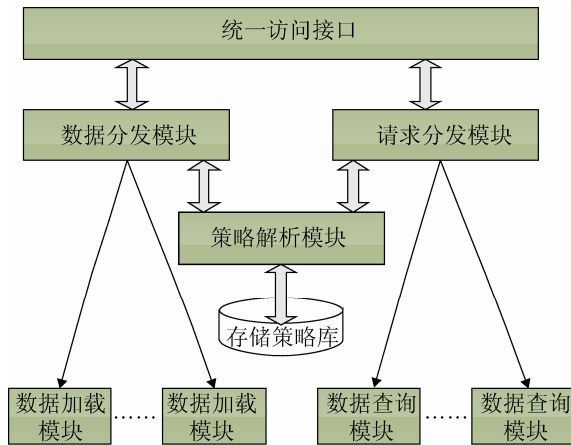


图 6 TSD 内部结构

4 实验与分析

本节针对上述工作进行了模拟实验与分析, 实验场景主要针对 Line-Policy 策略的数据加载展开。

4.1 实验场景

实验集群环境配置如下:

表 2 HBase 集群硬件配置

主机	namenode0	datanode1	datanode2
IP	192.168.0.1	192.168.0.3	192.168.0.5
角色	HMaster	RegionServer	RegionServer
CPU	4 核, 1.6GHZ	4 核, 1.6GHZ	4 核, 1.6GHZ
内存	DDR3, 4G	DDR3, 4G	DDR3, 4G
硬盘	SATA 7200 转, 500GB*1	SATA 7200 转, 500GB*1	SATA 7200 转, 500GB*1

为了真实模拟生产环境中海量时序数据对数据管理系统造成的冲击, 本节根据实验环境硬件配置, 模拟了大测点规模、高数据产生频率的场景: 分钟级的智能电表数据采样(生产环境中, 智能电表数据采样通常为小时级)。

实验场景: 6 万块智能电表, 10 个量测项/智能电表, 共 6 万个量测项, 均匀分布于 25 栋楼宇之中。楼宇、单元、电表组织形式如下:

表 3 楼宇/单元组织

楼宇	A~Y, 共 25 栋
单元/楼	1~8, 共 8 个
电表/单元	300 块

数据采样频率为 60 帧/小时, 模拟时间段为 1 h, 共 3 600 万条数据记录。原始文件大小为 2.1 GB, 数据格式如下:

```
A.unit1.15 1402201861 32157 direction=fd type=shark
A.unit1.15 1402201861 354 direction=fd type=peak
B.unit1.15 1402201861 4209 direction=fd type=shoulder
B.unit1.15 1402201861 15298 direction=fd type=offpeak
C.unit1.15 1402201861 26842 direction=fd
C.unit1.15 1402201861 29856 direction=bd type=shark
D.unit1.15 1402201861 32522 direction=bd type=peak
D.unit1.15 1402201861 10414 direction=bd type=offpeak
.....
```

“A”表示楼宇编号; “unit1”为 1 单元; “15”为电表编号; “fd”表示“前向”; “bd”表示“后向”; “shark”表示“波峰”。

4.2 实验结果

本实验重点记录了 namenode0, datanode1, datanode2 三台机器上的 CPU、磁盘和内存消耗, 分别由图 7 的(a), (b), (c)所示。

本实验将生成的样本数据均匀分成了 3 份, 每份 1 200 万条记录, 分别放在每台机器的本地硬盘上。每个 TSD 实例只对本地数据进行加载。同时, 为了降低不同机器间的网络、磁盘 I/O 冲突, 本实验还进行了如下优化^[21]:

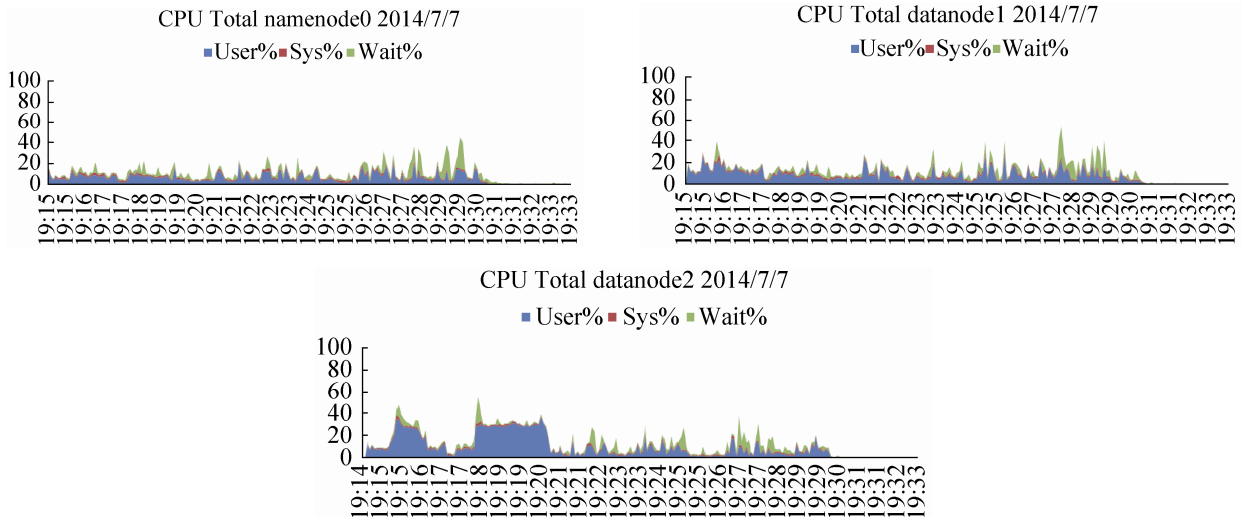
1) 对 HBase 数据表进行了 Region 预分配, 每张表预分配了 10 个 Region, 确保数据在加载过程中不会引起 HBase 表的 split 操作。

2) Region 的 startKey 和 endKey 的设定与测点 ID 分布一致, 确保生成的数据尽量均匀的分布于各个 Region 中。

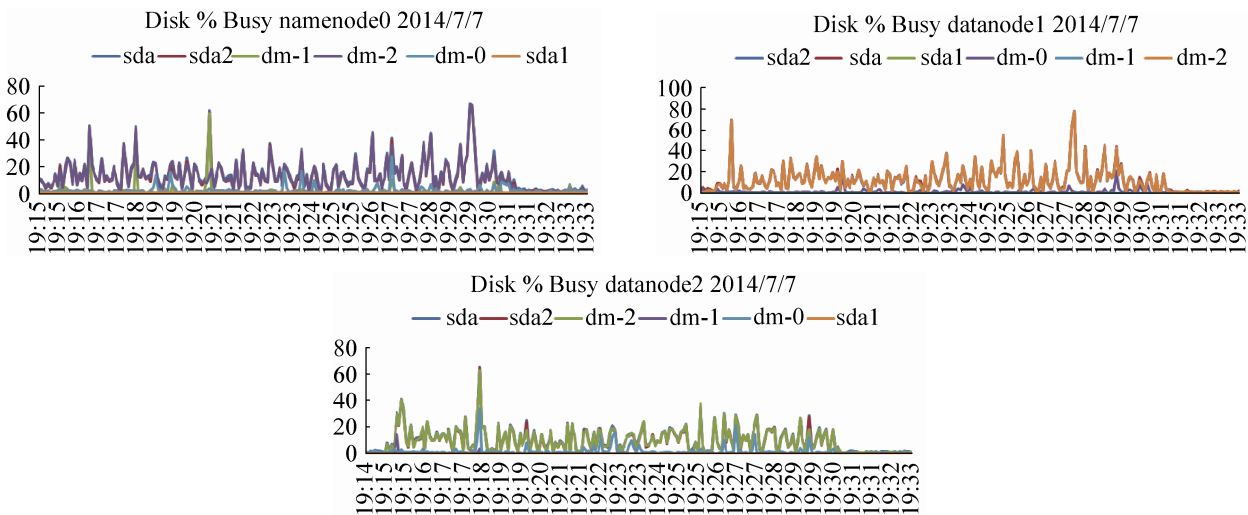
3) 对原始文件进行预处理, 使得单个原始文件所包含的数据记录尽量分布于同一个 Region Server 所管理的 Region 中。

4) 将原始文件放到对应的 RegionServer 上, 减少加载时网络传输量。

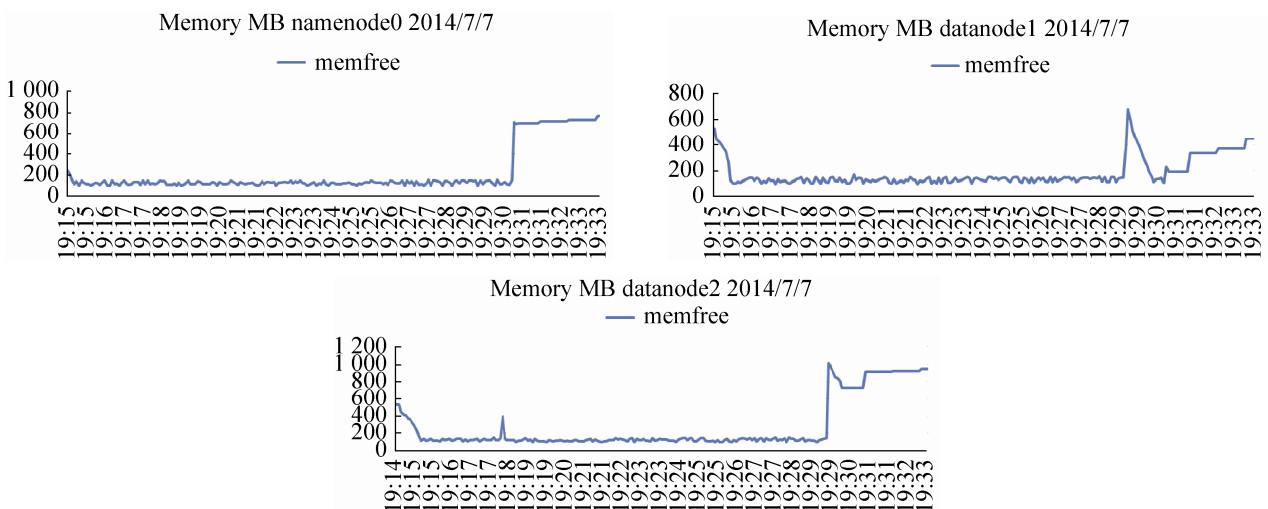
其中 3)、4)由图 5 中的连接负载平衡器模块完成。实验结果如表 4 所示。



(a) CPU 利用率



(b) 磁盘利用率



(c) 内存使用

图 7 实验数据

<http://www.china-simulation.com>

表 4 运行时间

节点名称	耗时/s	加载速度(条/秒)
namenode0	402.117	29 842.1
datanode1	372.790	32 189.7
datanode2	377.470	31 790.6

4.3 分析

分析结果可知,在采用上述方式加载时序数据时,CPU 利用率不高,说明生产环境中可进一步降低 CPU 核数、内存大小与硬盘块数之间的比值。磁盘利用率曲线显示磁盘 I/O 相对均衡,这也说明本文的 Region 切分方式和原始数据预处理方式发挥了作用,降低了不同 RegionServer 间的 I/O 冲突。集群的性能瓶颈体现在内存上,特别是在数据加载的过程中,整个加载过程中可用内存不足 100 M。

为了更好的评估实验结果,并指导生产环境部署,本文采用三元组模型 $\langle C, M, H \rangle$ 来评估实验结果,其中 C 为 CPU 核数, M 为内存大小(单位 G), H 为硬盘块数。 f 表示 $\langle C, M, H \rangle$ 配置的服务器器的最大吞吐率(单位:条/秒)。

$$\text{ThroughOut} = f(\langle C, M, H \rangle) \quad (1)$$

结果显示,在实验环境中单块硬盘加载速度可达 30 000 条/秒,即 $f(\langle 4, 4, 1 \rangle) = 30\,000$ (如表 2)。此时,磁盘 I/O 利用率约为 50%, CPU 利用率约为 20%,内存消耗约为 4 G。根据 Hadoop 集群中 CPU 利用率、I/O 利用率以及可用内存之间的线性比值关系^[15],可以推算 $\langle C, M, H \rangle = \langle 4, 8, 1 \rangle$ 时,吞吐率约为 6 万,即 $f(\langle 4, 8, 1 \rangle) = 60\,000$,此时单块磁盘 I/O 利用率约为 100%,成为性能瓶颈。因此在实际的生产环境中,单台服务器的配置满足如下规则(其中 N 为正整数)时,整个集群性能较优。

$$\langle C, M, H \rangle = N \times \langle 4, 8, 1 \rangle \quad (2)$$

进一步观察 NameNode0, DataNode1 和 DataNode2 的系统负载情况可知,在批量数据加载时,每个节点的资源消耗基本相当,并且几乎同时(19:15~19:29)由于内存消耗而达到了瓶颈,这充分说明了数据加载过程的并行性,以及 HBase 集群在本文所示的数据组织方式下的线性扩展能力。

5 结论

本文提出了一种基于 HBase 的智能电网时序大数据处理方法,该方法适用于物联网时序数据的存储管理。该方法采用一种策略驱动的存储机制按需存储组织数据。存储策略的制定充分考虑了时序数据的产生特点与使用特点,将同一时间产生的数据尽量分散存储,在数据加载时充分发挥整个集群的 I/O 能力,提升加载性能;在数据存储组织方面,将有可能同时访问的数据在物理存储位置上尽量集中且顺序组织,提升整个集群的查询分析能力。在扩展能力、负载均衡和冗余备份方面,充分利用了 Hadoop 与 HBase 的相关特性,降低了应用开发的复杂性。

未来将进一步研究 HBase 高速加载技术以及基于 HBase 的时序数据分析挖掘技术,研制面向物联网大数据的分布式时序数据库。

参考文献:

- [1] HayT, Tansleys, TolleK. The Fourth Paradigm: Data-Intensive Science Discovery [M]. Redmond, Washington, USA: Microsoft Research, 2009.
- [2] 宫学庆,金澈清,王晓玲,等.数据密集型科学与工程:需求和挑战[J].计算机学报,2012,35(8):1563-1578.
- [3] 王珊,王会举,覃雄派,等.架构大数据:挑战、现状与展望[J].计算机学报,2011,34(10):1741-1752.
- [4] 王元卓,靳小龙,程学旗,等.网络大数据:现状与挑战[J].计算机学报,2013,36(6):1125-1138.
- [5] 覃雄派,王会举,杜小勇,等.大数据分析—RDBMS 与 MapReduce 的竞争与共生[J].软件学报,2012,23(1):32-45.
- [6] 覃雄派,王会举,李芙蓉,等.数据管理技术的新格局[J].软件学报,2013,24(2):175-197.
- [7] Gray J, Liu DT, Nieto-Santesteban M, et al. Scientific data management in the coming decade [J]. ACM SIGMOD Record, 2005, 34(4): 34-41.
- [8] 丁道奇.复杂大电网安全性分析[M].北京:中国电力出版社,2010.
- [9] Apache Software Foundation. The Apache HBase Reference Guide [EB/OL]. (20140-06-10) [2014-08-12]. <http://hbase.apache.org/book/book.html>
- [10] Prichard J. RTSQL: Extending the SQL Standard To

- Support Real—Time Database [D]// Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Applied Mathematical Sciences. USA: University of Rhode Island, 1995.
- [11] Azer Bestavros, Kwei-Jay Lin, Sang Hyuk Son. Real-time database systems: Issues and applications [M]. Springer Science & Business Media, 2012.
- [12] Pugll W, Marlow. Compiler and tool support for real—time systems [C]// Proceedings of the ACM SIGPLAN workshop on language, ACM SIGPLAN. USA: ACM, 1994.
- [13] Ramamritham K. Real—time databases [J]. International Journal of Distributed and Paralled Databases (S0926-8782), 1993, 1(2): 199-226.
- [14] Jeffrey Dean, Sanjay Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, Operating Systems Design and Implementation [J]. Communications of the ACM (S0001-0782), 2008, 51(1): 107-113.
- [15] Apache Software Foundation. HDFS Users Guide [EB/OL]. (2014-08-01) [2014-07-20]. <http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html>
- [16] Ghemawat S, Gobihoff H, Leung S-T. The Google file system [C]// Proc. of the 19th ACM SOSP (Dec. 2003). USA: ACM, 2003: 29-43.
- [17] Opentsdb.net. OpenTSDB - A Distributed, Scalable Monitoring System [EB/OL]. (2014) [2014-08-30]. <http://opentsdb.net/>.
- [18] Chang F, Dean J, Ghemawat S, et al. Bigtable: A Distributed Storage System for Structured Data [J]. ACM Transactions on Computer Systems (TOCS) (S0734-2071), 2006, 26(2): 205-218.
- [19] Paul Zikopoulos, Chris Eaton. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data [M]. USA: McGraw-Hill Osborne Media, 2011.
- [20] N Dimiduk, A Khurana, MH Ryan, et al. HBase in Action [M]. USA: O'Reilly Media, 2012.
- [21] Yifeng Jiang. HBase Cookbook Administration. Birmingham [M]. UK: PACKT Publishing, 2012.

(上接第 541 页)

了解多重灾难环境下疏散人群的实时动态,分析疏散中人员运动规律,有助于各类灾难环境下人员疏散的研究工作。

参考文献:

- [1] Helbing D, Farkas I, Vicsek T, et al. Simulating Dynamical Features of Escape Panic [J]. Nature (S0028-0836), 2000, 407(6803): 487-490.
- [2] Varas A, Cornejoa M D, Mainemera D, et al. Cellular automaton Model for Evacuation Process with Obstacles [J]. Physica A (S0378-4371), 2007, 38(2): 631-642.
- [3] 刘磊, 刘群, 高铁, 等. 多出口条件下基于排队时间的疏散仿真研究 [J]. 计算机工程, 2011, 37(2): 349-352.
- [4] 徐高. 基于智能体技术的人员疏散仿真模型 [J]. 西南交通大学学报, 2003, 38(3): 301-303.
- [5] 史建勇, 任爱珠. 基于智能体的大型公共建筑人员火灾疏散模型研究 [J]. 系统仿真学报, 2008, 20(20): 5677-5681.
- [6] 陈佳俊, 安晓宇, 蔡希辉, 等. 基于 Agent 的人员疏散系统设计与实现 [J]. 计算机工程, 2010, 36(14): 264-266.
- [7] 黄希发, 王科俊, 郭莲英, 等. 基于 Agent 技术的人员疏散微观仿真模型研究 [J]. 系统仿真学报, 2009, 21(15): 4568-4572.
- [8] 朱哲民, 龚秀芬, 杜功焕. 声学基础 [M]. 南京: 南京大学出版社, 2001: 430-432.
- [9] 白音, 杨海勇, 石永久, 等. 大空间结构各种火灾场景下的温度场分布 [J]. 清华大学学报(自然科学版), 2011, 51(8): 1122-1127.
- [10] 周曲波. 用 $r(x)$ 函数推导高斯烟流模式 [J]. 环境科学研究, 1992, 5(1): 51-52.
- [11] 江辉仙, 林广发, 江立辉, 等. 校园楼宇火灾疏散路径分析和应用 [J]. 系统仿真学报, 2013, 25(9): 2171-2177.
- [12] Stahl F. BFIREs-II, A Behavior Based Computer Simulation of Emergency Egress During Fires [J]. Fire Technology (S0015-2684), 1982, 18(1): 49-65.