Journal of Computer Applications
计算机应用,2017,37(4):928 -935
ISSN 1001-9081
CODEN JYIIDU
2017-04-10
http://www.joca.cn

文章编号:1001-9081(2017)04-0928-08
DOI:10.11772/j.issn.1001-9081.2017.04.0928

基于 Spark Streaming 的实时能耗分项计量系统

武志学[1,2*]

(1.成都五舟汉云科技有限公司,成都 611731; 2.成都信息工程大学 信息安全工程学院,成都 610225)

(* 通信作者电子邮箱 zhixue.wu@gmail.com)

摘 要:能耗分项计量能够准确、及时、有效地发现能源使用问题,形成和实现最有效的节能措施。能耗分项计量系统需要对各项能源使用量在不同粒度上进行统计,既有实时性的需求,又需要涉及到聚合、去重、连接等较为复杂的统计需求。由于数据产生快、实时性强、数据量大,所以很难统一采集并入库存储后再作处理,这便导致传统的数据处理架构不能满足需求。为此,提出基于 Spark Streaming 大数据流式技术构建一个实时能耗分项计量系统,对实时能耗分项计量的系统架构和内部结构进行了详细介绍,并通过实验数据分析了系统的实时数据处理能力。与传统架构不同,实时能耗分项计量系统在数据流动的过程中实时地进行捕捉和处理,一方面把捕捉到的异常信息及时报警到前端,同时把分类分项统计处理的结果保存到数据库,以便进行离线分析和数据挖掘,能有效地解决上述数据处理过程中遇到的问题。

关键词:流式计算;能耗分项计量;Spark Streaming;Apache Kafka;大数据

中图分类号:TP391 **文献标志码**:A

Real-time detailed classification energy consumption measurement system based on Spark Streaming

WU Zhixue[1,2*]

(1. *Chengdu Wuzhou Handge Technology Limited*, *Chengdu Sichuan* 611731, *China*;
2. *School of Information Security Engineering*, *Chengdu University of Information Technology*, *Chengdu Sichuan* 610225, *China*)

Abstract: Detailed classification energy consumption measurement can discover energy consuming issues more accurately, timely and effectively, which can form and implement the most effective energy-saving measures. Detailed classification energy measurement system needs to calculate energy consumption amounts at multiple time scales according to detailed classification coding. Not only does it need to complete the tasks timely, but also need to deal with data aggregating, data de-duplication and data joining operations. Due to the fast speed of the data being generated, the requirement of the data being processed in real-time, and the big size of the data volume, it is difficult to store the data to a database system first, and then to process the data afterwards. Therefore, the traditional data processing infrastructure cannot fulfil the requirements of detailed classification energy consumption measurement system. A new real-time detailed classification energy consumption measurement system based on Spark Streaming technologies was designed and implemented, the system infrastructure and the internal structure of the system were introduced in detail, and its real-time data processing capabilities were proved through experiments. Different from the traditional ways, the proposed system processes energy consumption data in real-time to capture any unusual behaviour timely; at the same time, it separates the data and calculates the consumption usages according to the detailed classification coding, and stores the results to a database system for offline analysis and data mining, which can effectively solve the previously mentioned problems encountered in the data processing process.

Key words: stream computing; detailed classification energy consumption measurement; Spark Streaming; Apache Kafka; big data

0 引言

伴随着我国城市化进程的加快,大型公共建筑节能工作势在必行。如何达到既满足使用及舒适度的需求,又能科学、合理地节能降耗已经是全社会所要思考的问题。在大力推广节能减排的阶段,要达到最快、最明显的节能效果,不单是采用设备节能手段,更需要使用分项计量准确、及时、有效地发现能源使用问题,形成和实现最有效的节能措施。能耗分项计量是指对建筑的水、电、燃气、集中供热、集中供冷等各种能耗进行监测,从而得出建筑物的总能耗量和不同能源种类、不同功能系统的能耗量[1]。要实现分项计量,必须进行数据采集、数据传输、数据存储和数据分析等。所以,能耗分项计量是一个典型的流式大数据系统,具有数据量大、数据产生速度快、数据结构复杂等特点。

一般情况下,能耗分项计量包括空调系统、电梯系统、给排水系统、通风系统、照明系统及办公设备系统等。对于用能

收稿日期:2016-10-10;修回日期:2016-12-21。

作者简介:武志学(1960—),男,山西河津人,教授,博士,主要研究方向:云计算、流式数据处理、数据挖掘。

密度高、单体设备耗能大的集中空调系统,应进行更细致的计量,包括: 冷冻主机用电量、冷冻水泵用电量、冷却水泵用电量、冷却塔风机用电量、空调箱和新风机用电量等。所以进行能耗分项计量时,需要对各项能源使用量在不同粒度上对不同 数据进行统计,既有实时性的需求,又需要涉及到聚合、去重、连接等较为复杂的统计需求。由于数据产生快、实时性强、数据量大,如果采取传统的数据处理架构,首先对采集到的数据入库存储,然后再作处理,很难满足分项计量的需求。特别是为了找到能耗使用规律提出有效节能措施,不但需要部署大量能耗采集仪表,还需要进行更为复杂的数据处理,从而引起在单位时间内要处理的实时数据量和计算工作量同时大幅上升,这便导致传统的数据处理架构不能满足需要。为了解决这个问题,本文通过使用 Apache Kafka 和 Spark Streaming 模块构建了一个实时流式数据处理系统来进行能耗分项计量。与传统架构不同,实时流式数据处理系统在数据流动的过程中实时地进行捕捉和处理,并根据业务需求对数据进行计算分析,一方面把捕捉到的异常信息及时报警到前端,同时把分类分项统计处理的结果保存到数据库,以便进行离线分析和数据挖掘。本文将详细描述实时能耗分项计量的系统架构和内部结构,并对架构中所使用的大数据技术和系统进行介绍和分析,最后,通过实际测试结果对实时能耗分项计量系统的实时数据处理能力进行验证和分析。

1　相关研究

清华大学节能研究中心研制开发了能耗分项计量实时分析系统 EM-II [2],包括数据采集子系统、数据处理子系统、数据分析展示子系统三大核心部分,另外还有信息维护、数据上报、系统监测等几个子系统。数据采集子系统利用安装在现场的具有数字通信接口的电计量表和超声波冷热量表采集数据,并由数据采集器汇总接收通过网关由路由器连接到互联网,将数据远程传输回数据中心服务器。数据处理子系统负责校验解析接收到的原始数据,并根据能耗模型拆分计算得到分类分项数据。数据分析展示子系统将经过数据处理后的分类分项能耗数据进行分析、汇总和整合,一方面通过静态或者动态的图表方式将能耗数据展示出来,另一方面能够提供针对第三方的数据接入服务和数据发布服务。

Hysine 与多个高等院校及科研机构合作研制开发的 EMC-2000 建筑设备节能控制与管理系统[3],适用于新建、改建、扩建项目中建筑机电设备能效跟踪控制节能管理。整个能源管理系统由管理中心、主干通信网络、数据采集器、智能电表等组成,同时为与上一级能耗监测和管理系统连接预留系统接口。能源管理中心通过对现场数据采集器上传的数据进行存储、统计和分析,为业主提供有效的能源使用和持续的能源节约提供实施依据。

安科瑞开发的 Acrel-5000 建筑能耗分析管理系统[4] 以计算机、通信设备、测控单元为基本工具,根据现场实际情况采用现场总线、光纤环网或无线通信中的一种或多种结合的最优化的组网方式,为大型公共建筑的实时数据采集及远程管理与控制提供了基础平台,它可以和检测设备构成任意复杂的监控系统。

这些能耗分项计量系统都是参照国家住建部《国家机关办公建筑和大型公共建筑能耗监测系统》[5] 和《国家机关办公建筑和大型公共建筑能耗监测系统省、市级数据中心数据库结构文档》[6],采用了传统的数据处理模式,如图 1 所示。当数据采集程序接收到数据采集器发送的数据以后,首先把数据写入计量表原始数值数据库(D) ; 然后再由拆分程序按照各个仪表和能耗数据各级分项进行拆分和统计,并把结果写入分类分项能耗数据库(B) ; 最后再由分析展示程序基于建筑基本情况数据库(A) 、分类分项能耗数据库(B) 进行数据分析并展示给用户。

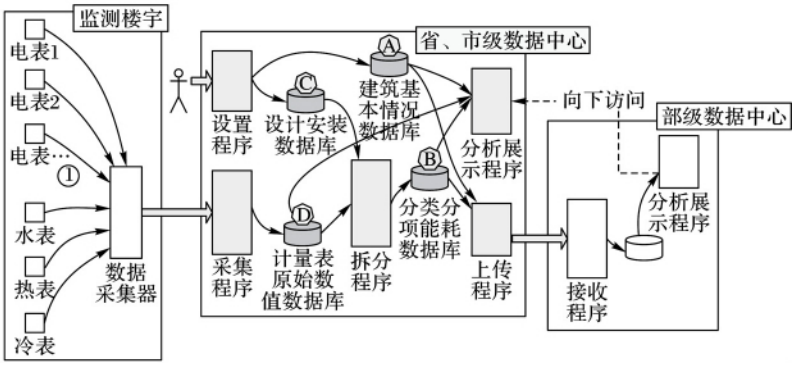

图 1　基于传统数据处理模式的分项计量系统

Fig. 1　Detailed classification energy consumption measurement system based on traditional data processing model

这种基于传统数据处理模式的分项计量系统只能适用于采集点数量比较少、统计分析比较简单的环境。在采集点数量达到上千时,随着时间的推移,分类分项能耗数据库的数据会不断累计快速增加,从而可能导致拆分程序无法及时完成对数据的拆分和统计。

2　实时能耗分项计量系统

为了解决基于传统数据处理模式的能耗分项计量系统存在的问题,本文设计并实现了一个基于 Spark Streaming 和 Apache Kafka 等大数据技术的实时能耗分项计量系统。在本章首先对 Spark Streaming 和 Apache Kafka 大数据技术进行简单介绍,然后描述如何使用 Spark Streaming 和 Apache Kafka 模块构建基于实时流式数据处理架构的实时能耗分项计量系统。

2.1　Spark Streaming

Apache Spark 是一个基于内存的、可以支持各种大数据应用场景的、高性能和高容错的开源大数据平台[7]。Spark Streaming 是 Apache Spark 的一个子项目,是一个运行在 Spark 引擎之上的实时处理工具[8]。

与 Hadoop[9] 大数据处理平台不同,Spark 建立在统一抽象的 RDD(Resilient Distributed Datasets) 之上,使得它可以以基本一致的方式应对各种大数据处理场景,包括 MapReduce、Streaming、SQL、Machine Learning 以及 Graph 等。

Spark 的另一个特点就是其高性能和容错性。Spark 是一种粗粒度数据并行的计算范式,计算的主体是数据集合 RDD,而非个别数据。RDD 集合内的所有数据都经过同样的算子序列,数据并行可编程性好,易于获得高并行性(与数据规模相关,而非与程序逻辑的并行性相关) ,也易于高效地映

射到底层的并行或分布式硬件上[10]。RDD是一个容错的、并行的数据结构，在保证容错的前提下，用内存来承载工作集。内存的存取速度快于磁盘多个数量级，从而可以极大提升性能[11]。Spark的容错是通过重放日志更新而取得的。因为Spark的函数式语义和幂等特性，重放日志更新RDD不会有副作用。另外，Spark记录的是粗粒度的RDD更新，所以容错的开销可以忽略不计。

Spark的实时性特点是通过Spark Streaming实现的。Spark Streaming将流式计算分解成一系列短小的批处理作业，也就是把输入数据流按照批次大小(如1 s)分成一段一段的数据形成DStream(Discretized Stream)，而每一段数据都转换成Spark中的RDD，如图2所示。

图2 DStream的组成

Fig. 2 Formation of DStream

Spark Streaming提供了两种操作类型，分别为Transformations和Output操作。对DStream的Transformation操作变为Spark中对RDD的Transformation操作，从一个已知的DStream经过转换得到一个新的DStream；而且Spark Streaming还额外增加了一类针对“窗口”(Window)的Transformation操作，可以更灵活地控制DStream的大小(时间间隔大小、数据元素个数等)。整个流式计算根据业务的需求可以对中间的结果进行叠加，或者使用Output操作将DStream数据输出到一个外部的存储系统，如数据库或文件系统等。

Spark具有极高的扩展性与吞吐量。根据Spark官方网站FAQ，最大的已知Spark集群有8 000个节点[12]；并且随着大数据增多，预计集群规模也会随之变大，以便继续满足吞吐量方面的预期。另外，使用Spark的EC2启动脚本可以轻松地在Amazon EC2上启动一个独立集群。Spark目前在EC2上已能够线性扩展到100个节点(每个节点4核)，可以以数秒的延迟每秒处理6 GB的数据量[11]。

一个Spark集群由多个工作节点(Worker Node)组成，每个工作节点可以运行一个或多个Executor，如图3所示。Executor是一个用于应用程序或者工作节点的进程，负责处理Tasks，并将数据保存到内存或者磁盘中。每个应用程序都有属于自己的Executor，一个Executor则包含了一定数量的Slots来运行分配给它的任务。在Spark中，每个作业(Job)都被分隔成多个彼此依赖称之为Stage的Task。一个Task就是一个工作单元，可以发送给一个Executor执行。Task是用来执行应用的实际计算工作。每个Task占用Executor的一个Slot。

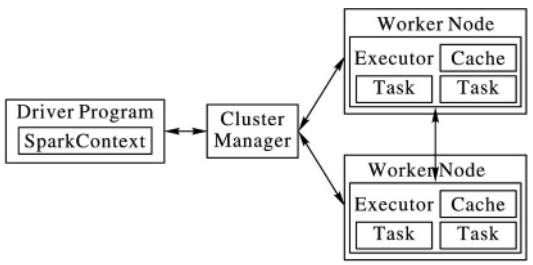

图3 Spark Streaming集群架构

Fig. 3 Cluster structure of Spark Streaming

Spark Streaming流计算可以在数据产生并流入系统时就进行处理并马上得出结果，非常适合能耗分项计量中能耗数据不断产生的场景和对实时性的需求。

选择Spark Streaming的另一个原因是因为Spark可以在支持实时流式处理的同时，进行高效的批处理、交互式SQL查询和数据挖掘，从而可以使能耗分项计量系统不但可以实时地为用户捕捉能耗异常情况进行报警，还可以提供离线统计分析和数据挖掘的服务。

2.2 Apache Kafka

Apache Kafka是一个分布式的、高吞吐量的、易于扩展的基于主题发布/订阅的消息系统，最早是由LinkedIn开发，并于2011年开源并贡献给Apache软件基金会[13]。

Kafka的逻辑架构如图4所示。Kafka对消息保存时根据话题(Topic)进行归类，发送消息者成为生产者(Producer)，消息接受者成为消费者(Consumer)。此外Kafka集群由多个服务器组成，每个服务器成为代理(Broker)。无论是Kafka集群，还是Producer和Consumer都依赖于Zookeeper来保证系统可用性。

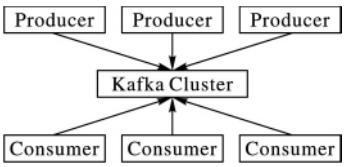

图4 Kafka的逻辑架构

Fig. 4 Kafka logical structure

一个话题可以认为是一类消息，每个话题将被分成多个分区(Partition)。设计分区的最根本原因是Kafka基于文件存储，通过分区可以将日志内容分散到多个服务器上，来均衡负载，保证消息保存/消费的效率。如果一个话题对应一个文件，那这个文件所在的机器IO将会成为这个话题的性能瓶颈，而有了分区后，不同的消息可以并行写入不同代理的不同分区里，属于顺序写磁盘，因此效率非常高，极大地提高了Kafka的吞吐率。所以，消息分区是Kafka高吞吐率的一个很重要的保证，即使在非常廉价的商用机器上也能做到单机支持每秒10^4条以上消息的传输[14]。此外，越多的分区意味着可以容纳更多的消费者，可以有效提升并发消费的能力。Kafka的消息分区结构如图5所示。

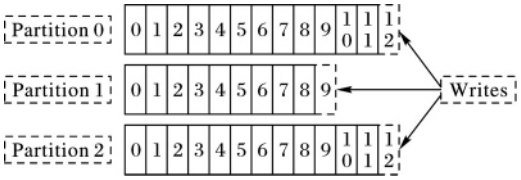

图5 Kafka消息分区结构

Fig. 5 Kafka message partition structure

与传统的消息系统不同，Kafka系统中存储的消息没有明确的消息ID，消息通过日志中的位置称为偏移量来唯一标记一条消息，这样就避免了维护密集寻址，用于映射消息ID到实际消息地址的随机存取索引结构的开销。这种设计大大提高了Kafka的性能。

Kafka的另外一个创新是即使消息被消费，消息仍然不会被立即删除。日志文件将会根据代理中的配置，保留一定的时间之后删除；比如日志文件保留2 d，那么之后文件会被清除，无论其中的消息是否被消费。Kafka通过这种简单的手段

来释放磁盘空间,从而可以减少消息消费之后对文件内容改动的磁盘 IO 开支。

Kafka 还有一个创新点就是 Kafka 代理是无状态的,由消费者维护已消费的状态信息。这种设计的一个好处就是消费者可以退回到老的偏移量再次消费数据。因为代理是无状态的,它不需要标记哪些消息被哪些消费者消费过,也不需要代理去保证同一个消费者组里只有一个消费者能消费某一条消息,因此也就不需要锁机制,这也为 Kafka 的高吞吐率提供了有力保障。

为了提高可用性,Kafka 可以配置分区需要备份的个数,每个分区将会被备份到多台 Kafka 服务器上,以提高可用性。每个分区都有一个 Kafka 服务器为领导者(Leader),负责所有的读写操作。如果领导者失效,那么将会有其他跟随者(Follower)来接管成为新的领导者。跟随者只是单调地和领导者跟进,同步消息即可。从集群的整体考虑,Kafka 会将领导者均衡地分散到每个 Kafka 服务器上,来确保整体的性能稳定。

Kafka 可以同时支持离线数据处理和实时数据处理。如图 6 所示,Kafka 同时支持点到点分发模型,即多个消费者共同消费队列中某个消息的单个副本,以及发布-订阅模型,即多个消费者接收自己的消息副本。根据这一特性,可以使用 Spark 实时流处理系统对消息进行实时在线处理,同时使用 Hadoop 这种批处理系统进行离线处理,还可以同时将数据进行实时备份,只需要保证这三个操作所使用的消费者属于不同的消费者即可。

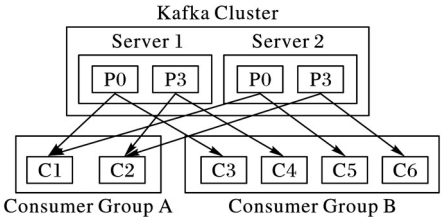

图 6　Kafka 消息分发模型

Fig. 6　Kafka message distribution model

总之,Kafka 是一种处理大量数据的新型消息系统,其高吞吐量、高可靠、高可用、易扩展的特性完全适应于能耗分项计量系统。此外,通过利用 Kafka 同时支持多种处理模型的特点,能耗分项计量系统可以在进行能耗数据在线处理的同时,对能耗数据进行备份和离线处理。

2.3　实时能耗分项计量系统架构

本文设计的实时能耗分项计量系统的整体架构如图 7 所示,主要包括后端能耗数据采集部分、Kafka 消息系统、Spark Streaming 集群、Hadoop 集群、前端实时展示应用和前端分析展示应用,以及分类分项能耗数据库和计量表原始数值数据库。

流式处理系统主要通过网络 Socket 通信来实现与外部 IO 系统的数据交互。由于网络通信的不可靠特点,发送端与接收端需要通过一定的协议来保证数据包的接收确认和失败重发机制。不是所有的 IO 系统都支持重发,这至少需要实现数据流的持久化,同时还要实现高吞吐和低时延。通过前面的介绍,可以确定 Kafka 具备这些特点,完全能够作为实时能耗分项计量系统的外部数据源。

除了把 Kafka 当成输入数据源之外,实时能耗分项计量系统还将其作为信息输出数据源向前端实时展示应用推送相关报警和实时流信息。

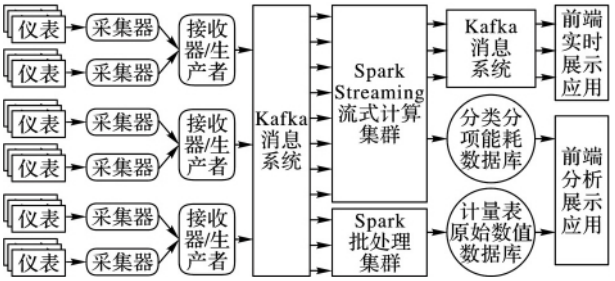

图 7　实时能耗分项计量系统整体架构

Fig. 7　Architecture of real-time detailed classification energy consumption measurement system

能耗数据采集部分包括能耗采集仪表和数据采集器。一般来讲,每个数据采集器可以连接 16 到 32 块采集仪表。数据采集器负责接收所连接采集仪表发来的能耗数据,并把数据整理为住建部所制定的能耗数据通信协议格式[15],然后按照设置的时间间隔发送到设定的数据接收器。每个数据接收器就是 Kafka 消息系统的消息生产者,负责把从数据采集器发来的数据写入 Kafka 消息系统,从而保证了数据的可靠性。

按照住建部要求,计量表采集到的能耗数据一方面必须写入计量表原始数值数据库,同时还需要按仪表、按分项进行拆分并把结果写入分类分项能耗数据库。在满足住建部基本要求的同时,实时能耗分项计量系统还对能耗数据进行实时分析以便能够及时捕捉能耗异常情况,并报警给用户。

为了能够保证能耗数据处理的实时性,实时能耗分项计量系统充分利用 Kafka 消息系统可以同时支持多个消费者组的能力,为能耗数据消息设置两个消费者组。一个是运行在 Spark Streaming 流式计算集群上的能耗数据实时数据拆分程序;另一个则是运行在 Spark 批处理集群上的计量表原始数值写入程序。

运行在 Spark Streaming 集群上的能耗数据拆分程序是实时能耗分项计量系统的核心模块。数据拆分程序以 Kafka 消息系统作为能耗数据输入流进行实时在线处理。首先,数据拆分程序对能耗数据进行分类分项拆分,并形成多个数据流供其他业务处理模块使用。第二,数据拆分程序把分类分项拆分结果按照不同时间粒度进行统计,并把统计结果写入分类分项能耗数据库。时间粒度分为 15 min、小时、天和月。

除了能耗数据拆分程序之外,Spark Streaming 集群还可以进行多种实时在线数据处理,比如能耗热点分析和能耗异常分析。这些能耗数据处理程序并不直接从 Kafka 消息系统中获取数据,而是使用能耗数据拆分程序生成的数据流进行数据处理,并把分析结果通过 Kafka 消息系统提供给前端实时展示应用。

实时能耗分项计量系统的另一部分功能是进行离线数据统计、分析以及数据挖掘。前端能耗分析程序基于能耗数据拆分程序写入到分类分项能耗数据库的数据,以及计量表原始数值数据进行各类能耗数据统计、分析以及数据挖掘,从而使用能单位可以掌握详细能耗使用情况,为制定节能策略提供科学依据。

因为能耗数据记录的数量远远超过了传统关系型数据库可以支持的容量,计量表原始数值数据库和分类分项能耗数据库均使用了 HBase 数据库。HBase 是运行在 Hadoop 上的 NoSQL 数据库,它是一个分布式的和可扩展的面向列的数据

库，可以在一组通用硬件上存储许多具有数十亿行和上百万列的大表[9]。HBase能够融合Key-Value数据模式带来实时查询的能力，以及通过MapReduce或Spark进行离线处理或者批处理的能力。总之，HBase能够存储大量的数据，让用户在大量的数据中查询记录，并从中获得综合分析报告。所以，HBase非常适合于存放计量表原始数据和分类分项能耗数据。HBase不但可以满足能耗分项计量系统每天几十万条记录的大数据量需求，还可以与Hadoop的MapReduce以及Spark SQL和Spark MLlib结合为用户提供高效能耗数据分析和数据挖掘工作。

2.4 实时能耗分项计量系统内部结构

实时能耗分项计量系统的内部结构如图8所示。Spark Streaming集群由多个工作者节点(Worker Node)组成，每个工作者节点包含一个或多个Spark Executor。同时，在每个工作节点还安装了用来存储能耗数据的数据库系统HBase和MySQL，以及数据仓库系统Hive。HBase用来存储能耗分类分项数据和计量表原始数值数据；MySQL用来存储与用能单位和分项计量系统各种设备部署情况的结构化数据；Hive用来按主题、多维度、多粒度对分类分项能耗数据进行存储和管理，为后期进行离线分析和数据挖掘提供良好的基础。

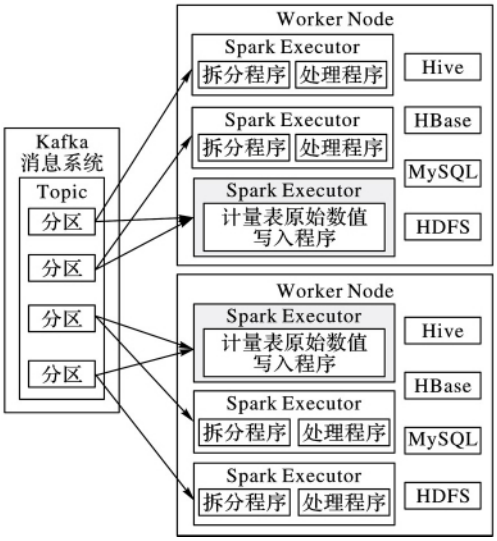

图8 实时能耗分项计量系统内部结构

Fig. 8 Internal structure of real-time detailed classification energy consumption measurement system

来自各个数据采集器的能耗数据使用同一个话题(Topic)写入Kafka消息系统。为了保证实时能耗分项计量系统的吞吐量，以便支持更多的计量表，我们对能耗数据消息话题进行了分区。通过消息分区可以提高消息生产者和消息消费者的并发能力。

在实时能耗分项计量系统中，每个消息分区将有两个消息消费者小组：一个是Spark Streaming流式集群的能耗数据拆分程序小组；另一个是Spark批处理集群的计量表原始数值数据写入程序小组。每个小组都由多个消费者组成，每个消息分区的数据都会被每个小组中的一个消费者接收。

图8描述了一个包含两个Worker Node的Spark集群，每个Worker Node运行了三个Spark Executor。每个Worker Node上有两个Executor属于能耗数据拆分程序小组，另外一个属于计量表原始数值写入程序小组。

能耗数据消息话题分成了四个分区，能耗数据拆分程序小组中的每个Executor消费一个消息分区的数据；计量表原始数据值写入程序小组中的每个Executor负责消费两个消息分区的数据。一般来讲，分区的个数最好是消费者小组中消费者的倍数，也就是说，同小组中的每个消费者负责处理的消息分区个数是等同的。在实际环境中消息话题分区的个数需要按照整个系统连接的计量仪表的个数来确定。

与Receiver方式相比，Direct方式虽然使用较为复杂，但是它能提供更好的灵活性和可靠性，所以本文选用Direct方式。Direct方式使用Kafka的基本API，由Spark Streaming负责记录在每个消息分区中的消费位移，也就是已经消费过的消息位置，并保存在Spark系统的检测点(Check Point)记录中。使用Direct方式，Spark Streaming会周期性地查询Kafka，来获得每个消息分区的最新的位移，从而定义每个数据块的数据范围。当处理消息的作业启动时，就会使用Kafka的简单消费API来获取Kafka指定范围的数据。Spark会创建跟Kafka分区一样多的RDD分区，并且会并行从Kafka中读取数据。所以在Kafka分区和RDD分区之间，有一个一对一的映射关系。采用Direct方式的另外一个优势就是可以利用Kafka保证数据的可靠性，并且可以保证数据是消费一次且仅消费一次。

在每个Spark Executor中，运行着能耗数据拆分程序和多个能耗数据实时处理程序。能耗拆分程序的功能在前面已经介绍。每个能耗数据处理程序基于拆分程序生成的实时数据流完成一定的数据处理工作，并把部分数据处理的结果写入Kafka消息系统供前端实时展示应用使用，同时还会把一些数据处理结果写入MySQL数据库供前端分析系统使用。

能耗用量异常分析程序是我们提供的一个能耗实时数据处理程序，它基于数据拆分程序提供的能耗使用数据流，根据用能单位设置的各类阈值以及正常能耗使用量发现用能异常情况，并通过Kafka消息系统及时报警给前端实时展示应用。比如，单位给某办公楼层设置的空调用电的阈值为每小时20度，能耗异常分析程序在对能耗使用数据流进行处理时就会检测该楼层的空调用电量，当用电量超过每小时20度时，就会产生报警消息通过Kafka提交给前端实时展示应用。同样的，假定某小区正常煤气流量为10 m^3/min左右，如果能耗用量异常分析程序发现该小区煤气流量远远超过了10 m^3/min，那么就有可能是发生了煤气管道漏气。这时，能耗用量异常分析程序就会产生报警消息。

能耗用量热点分析程序是我们提供的另一个能耗实时数据处理程序，它会实时统计每个计量点的每刻的能耗使用量并通过Kafka消息系统发布。前端实时展示应用可以获取感兴趣的计量点的流量统计来绘制能耗用量热点图，从而可以一目了然地及时了解所关心计量点的能耗使用状况。

2.5 实时能耗分项计量系统的优势

本文提出的实时能耗分项计量系统充分利用了最先进的大数据技术，特别是流计算技术，并针对能耗分项计量的特点对整体系统架构和内部结构进行了认真的研究与设计。与传统的能耗分项计量系统比，本文提出的实时能耗分项计量系统具有如下优势：

首先，实时能耗分项计量系统可以同时支持实时在线数据处理和离线数据统计分析，而传统的能耗分项计量系统只支持对能耗使用情况的离线统计和分析。实时能耗分项计量

系统的异常情况实时报警功能和能耗使用热点实时分析功能,不仅可以使用能单位在发生能耗异常情况时可以及时采取相应措施,防止异常情况蔓延,还可以让用能单位随时掌握整体能耗情况的实时现状。

其次,实时能耗分项计量系统具有很强的数据处理能力。整体系统架构使用了当前最先进的快速流式处理系统 Spark Streaming 和具有高可靠、高吞吐量的 Kafka 消息系统作为实时数据流处理的核心架构。整个数据处理过程是基于内存,而不像传统能耗分项计量系统需要把数据首先写入文件系统,然后再读入到内存进行处理,所以,实时能耗分项计量系统的处理效率会比传统能耗分项计量系统提高百倍以上。这意味着,在同样的硬件配置情况下,实时能耗分析计量系统可以支持的能耗采集点数可以提高上百倍。

第三,实时能耗分项计量系统具有很强的可扩展性。实时能耗分项计量系统架构中的 Kafka 消息系统、Spark 系统、HBase 系统和 Hadoop 系统都是分布式集群结构,并具有很强的扩展能力。所以,在使用实时能耗分项计量系统的每个阶段,用户只需要部署能够满足当时能耗监控需求的设备即可,而不需要考虑后期可能的需求。这一方面可以节省用户的投资成本,还减少了用户初期部署的设计负担。

第四,实时能耗分项计量系统提供快速数据挖掘能力。除了强大的实时数据处理能力以外,借助于 Spark 平台,实时能耗分项计量系统还可以利用 Spark MLlib 进行深度数据挖掘,发现复杂的能耗数据之间的关联关系,从而为制定有效的节能措施提供科学依据。基于 Spark MLlib 的数据挖掘效率会远远高于基于 MapReduce 模式的 Mahout 数据挖掘系统的效率。

第五,实时能耗分项计量系统可以很容易增加新的业务处理功能。在当前系统中,提供了能耗异常分析和能耗用量热点分析两个实时处理功能,但是今后可以根据用户需求很方便地添加新的业务处理能力。新添加的业务处理功能将会与原有的处理并行进行,并不会影响现有的实时业务处理能力。

3　实验与分析

为了检验实时能耗分项计量系统进行分项计量和实时数据处理的能力,实际部署了一套实时能耗分项计量系统,在对各种参数进行优化之后,进行了一系列的测试。

3.1　测试环境

测试环境是运行在云平台上的 7 台虚拟机组成。每台虚拟机的配置为 8 核 CPU,25 GB 内存,1 TB HDD 磁盘。图 9 描述了实时能耗分项计量系统各个模块的部署情况。

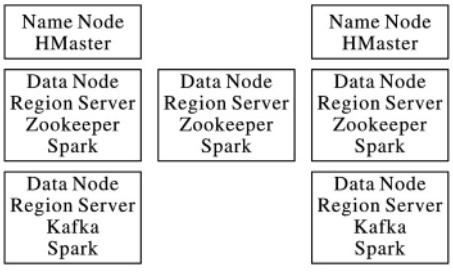

图 9　系统测试环境

Fig. 9　System testing environment

系统部署的指导思想是要保证整个系统的可靠性和可扩充性,并且保证节点之间的负载均衡性。具体部署情况如下:在两台服务器上部署了 Hadoop 的 Name Node 和 HBase 的 Master Server;三台服务器上部署了 Zookeeper、Spark、Hadoop 的 Data Node 和 HBase 的 Master Server;最后两台服务器上部署了 Kafka、Spark、Hadoop 的 Data Node 和 HBase 的 Master Server。

3.2　测试结果与分析

实时能耗分项计量系统的性能指标主要考虑的是系统的吞吐量和处理数据的延迟时间。系统的吞吐量一般用两个指标来衡量:一是单位时间内系统能够处理的能耗数据的条数;二是系统处理一条能耗数据所需要的时间。单位时间内处理的数据条数越多说明系统的吞吐量越高,系统处理数据的能力越强。处理能耗数据的延迟时间的指标也有两个:一个是从接收到一条能耗数据到开始处理该条数据之间的时间间隔称为调度延迟时间(Scheduling Delay);另一个是从接收到一条能耗数据到处理完该条数据之间的时间间隔称为总延迟时间(Total Delay)。处理能耗数据的延迟时间越小,说明系统处理数据越及时,系统实时性越强。

在测试中,通过给实时能耗分项计量系统的 Kafka 消息系统加载实际数据来测试系统的吞吐量和处理数据的延迟时间,测试结果如图 10 所示。图中展示的测试运行了 6 min 27 s,每秒加载一组能耗数据,每组数据大约包含 140 条能耗记录,总共处理了 387 组数据,64 968 条能耗记录。

图 10(a) 展示的是给系统加载能耗数据的速率(Input Rate)。可以看出给系统加载能耗数据的平均速率为每秒 167.88 条记录,瞬间最高值达到了每秒 300 条以上,绝大多数数据都是按每秒 140 到 200 条数据的速率发送的。

图 10(b) 展示的是数据的调度延迟时间。尽管显示的平均调度延迟时间为 10 ms,但从图中可以看出这主要是由于在测试刚开始启动时,第一批数据有一个 2 s 延迟而导致的。从右图可以看出,其余批次数据的调度延迟平均值在 0.2 ms 以内。

图 10(c) 展示的是处理一批能耗数据所需要的时间。图中显示处理每批数据的平均时间为 133 ms。如果考虑到除去系统刚启动运行的第一批数据,那么平均处理每一批数据的时间会在 100 ms 以内。从图的形状来看,除了第一批数据以外,系统整个处理过程非常平稳。

图 10(d) 展示的是系统处理能耗数据的总延迟的平均时间为 143 ms。类似于调度延迟时间,总延迟平均时间也因为第一批数据的延迟而拉高。如果剔除第一批数据,其余批次数据的总延迟时间均在 100 ms 以内。

图 11 以表格的形式展示了测试最后 26 批次数据的结果,包括每批数据的条数和提交时间,也就是包含多少条能耗数据、调度延迟时间、处理时间和总延迟时间。通过图 11 的数据,可以更进一步佐证上面对测试数据的分析结果。从图 11 可以看出每批数据平均包含 146.88 条能耗数据;平台调度延迟时间为 0.34 ms;每批数据的平均处理时间为 117.65 ms;平均总延迟时间为 118 ms。所以,实时能耗分项计量的吞吐量为每秒处理 1 248 条记录(146.88/117.65 × 1 000)。

图 11 的数据是在系统度过了初始阶段达到稳定以后的数据,结合图 10 的整体情况,可以知道图 11 的数据更能代表

实时能耗分项计量系统的特性。

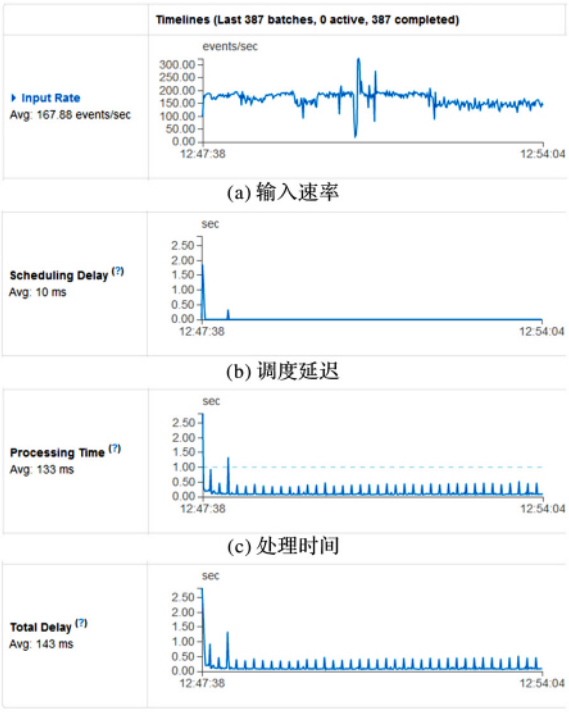

图 10 系统实时性测试结果

Fig. 10 Testing results of real-time performance

Completed Batches (last 387 out of 387)

Batch Time	Input Size	Scheduling Delay	Processing Time	Total Delay
2016/09/08 12:54:04	142 events	0 ms	91 ms	91 ms
2016/09/08 12:54:03	150 events	1 ms	89 ms	90 ms
2016/09/08 12:54:02	138 events	0 ms	82 ms	82 ms
2016/09/08 12:54:01	145 events	0 ms	81 ms	81 ms
2016/09/08 12:54:00	152 events	1 ms	93 ms	94 ms
2016/09/08 12:53:59	164 events	0 ms	79 ms	79 ms
2016/09/08 12:53:58	164 events	0 ms	77 ms	77 ms
2016/09/08 12:53:57	162 events	0 ms	0.5 s	0.5 s
2016/09/08 12:53:56	159 events	0 ms	85 ms	85 ms
2016/09/08 12:53:55	139 events	0 ms	87 ms	87 ms
2016/09/08 12:53:54	157 events	1 ms	97 ms	98 ms
2016/09/08 12:53:53	144 events	0 ms	0.1 s	0.1 s
2016/09/08 12:53:52	151 events	0 ms	78 ms	78 ms
2016/09/08 12:53:51	146 events	0 ms	0.1 s	0.1 s
2016/09/08 12:53:50	121 events	1 ms	0.1 s	0.1 s
2016/09/08 12:53:49	144 events	1 ms	80 ms	81 ms
2016/09/08 12:53:48	158 events	0 ms	78 ms	78 ms
2016/09/08 12:53:47	155 events	0 ms	0.5 s	0.5 s
2016/09/08 12:53:46	136 events	1 ms	83 ms	84 ms
2016/09/08 12:53:45	154 events	1 ms	0.1 s	0.1 s
2016/09/08 12:53:44	157 events	0 ms	81 ms	81 ms
2016/09/08 12:53:43	167 events	0 ms	83 ms	83 ms
2016/09/08 12:53:42	159 events	0 ms	86 ms	86 ms
2016/09/08 12:53:41	145 events	0 ms	86 ms	86 ms
2016/09/08 12:53:40	151 events	1 ms	66 ms	67 ms
2016/09/08 12:53:39	159 events	1 ms	77 ms	78 ms

图 11 系统实时性详细测试结果

Fig. 11 Detailed testing results of real-time performance

3.3 测试结论

从上面的实验结果可以看出，在实时能耗分项计量系统启动以后，只需要处理完第一批数据以后，就能达到稳定的运行状态，大约 3 s。平均实时能耗分项计量系统的吞吐量为每秒处理 1 248 条记录，平台调度延迟时间为 0.34 ms；每批数据的平均处理时间为 117.65 ms；平均总延迟时间为 118 ms。所以，实时能耗分项计量系统具有很高的吞吐量，实时性很强，并且系统数据处理速率很平稳。

按照国家住建部分项计量规则要求，每块分项计量仪表需要每 15 min 提交一次数据；而在 15 min 时间内，实时能耗分项计量系统可以处理超过 100 万条（15 × 60 × 1 200）数据。也就是说，在现有的系统配置环境下，实时能耗分项计量系统可以支持 100 万块仪表。因为传统的能耗分项计量系统需要先把数据写入磁盘文件，然后再读入进行数据处理，并且没有采用大数据并发处理技术，所以每套系统能支持的分项计量仪表一般都在 1 000 块左右，只适合于单个企事业单位的分项计量工作。实时能耗分项计量系统将处理能耗数据的能力提升了上千倍，完全可以满足同一个城市的所有公共事业单位提供分项计量服务。

4 结语

本文提出了一种基于 Spark Streaming 和 Apache Kafka 模块构建的用于能耗分项计量的实时流式处理系统，简称实时能耗分项计量系统。它能够满足能耗分项计量数据产生快、实时性强、数据量大的数据处理需求。与传统数据处理架构不同，实时能耗分项计量系统不仅提供离线数据的统计与分析，并且根据业务需求对数据进行实时在线处理，在数据流动的过程中实时地捕捉异常信息并进行处理，最终把结果保存或者分发给需要的组件。本文详细描述了实时能耗分项计量系统的整体架构和内部结构，阐述了其主要特点，并通过实际测试证明了其强大的数据处理能力和实时性。

从功能方面来讲，与传统的能耗分项计量系统相比，实时能耗分项计量系统的最大优点就是在支持离线能耗统计的同时，还可以支持实时在线数据处理和深度数据挖掘。比如，可以对能耗数据流进行实时分析，发现能耗用量异常情况，及时报警给用户，以便用能单位可以及时采取相应措施，防止异常情况蔓延。再比如，实时能耗分项计量系统还可以实时统计各计量点能耗情况并实时展示给用户，使用能单位及时掌握整体能耗的实时现状。

从性能方面来讲，本文提出的实时能耗分项计量系统进行能耗数据处理的能力远远超过传统的能耗分项计量系统，能够支持能耗数据采集点的个数高出上千倍。并且，实时能耗分项计量系统具有很强的扩展能力，可以通过增加服务器和存储设备来提高其总体处理能力，从而可以支持更多的能耗数据采集点。

总之，本文提出的实时能耗分项计量系统不论从性能方面、功能方面，还是从系统的可扩展方面都远优于传统的能耗分项计量系统。本系统的第一版开发已经完成，已经在 2016 年开始在四川省进行实地部署。此外，本文提出的实时流式数据处理系统还可以应用于其他流式数据处理场合，比如股市走向分析、气象数据测控、网站用户行为分析和公路卡口过车数据分析等。

参考文献（References）

[1] 清华大学建筑节能研究中心. 中国建筑节能年度发展研究报告 2010[M]. 北京: 中国建筑工业出版社, 2010: 105 – 130. (Building energy conservation research center of tsinghua university. Annual Report of China Building Energy Conservation 2010[M]. Beijing:

China Architecture and Building Press, 2010: 105 – 130.)

[2] 魏庆芃. 大型公共建筑能耗分项计量实时监测分析系统 EMS-II 的发展[J]. 建筑, 2009(3): 34 – 37. (WEI Q P. Development of the detailed classification energy consumption measurement system for large public building EMS-II [J]. Construction and Architecture, 2009(3): 34 – 37.)

[3] EMC-2000 能源管理系统[EB/OL]. [2016-09-10]. http://www.hysine.cn/web/list/369/1.html. (EMC-2000 energy management system [EB/OL]. [2016-09-10]. http://www.hysine.cn/web/list/369/1.html.)

[4] 黄斌, 杜运东, 曹雪华. 基于 Acrel-5000 的大型公共建筑能耗监测系统设计与应用[J]. 智能建筑电气技术, 2009, 3(5): 47 – 50. (HUANG B, DU Y D, CAO X H. Design and application of large public building energy consumption monitoring system Acrel-5000[J]. Electrical Technology of Intelligent Building, 2009, 3(5): 47 – 50.)

[5] 国家机关办公建筑和大型公共建筑能耗监测系统——分项计量数据采集技术导则[S]. 北京: 中华人民共和国住房和城乡建设部, 2008: 1 – 25. (Government offices and large public buildings energy consumption monitoring system — the technical guidance for detailed classification energy data collection[S]. Beijing: Ministry of Housing and Urban-Rural Development of the People's Republic of China, 2008: 1 – 25.

[6] 国家机关办公建筑和大型公共建筑能耗监测系统——省、市级数据中心数据库结构文档[S]. 北京: 中华人民共和国住房和城乡建设部, 2009: 1 – 12. (Government offices and large public buildings energy consumption monitoring system — provincial and municipal data center database structure document[S]. Beijing: Ministry of Housing and Urban-Rural Development of the People's Republic of China, 2009: 1 – 12.)

[7] Spark programming guide[EB/OL]. [2016-07-27]. http://spark.apache.org/docs/latest/programming-guide.html.

[8] Spark streaming programming guide [EB/OL]. [2016-07-27]. https://sp ark.apache.org/docs/latest/programming-guide.html.

[9] 陆嘉恒. Hadoop 实战[M]. 北京: 机械工业出版社, 2012: 1 – 121. (LU J H. Hadoop in Action[M]. Beijing: China Machine Press, 2012: 1 – 121.)

[10] KARAU H, KONWINSKI A, WENDELL P, et al. Spark 快速大数据分析[M]. 王道远, 译. 北京: 人民邮电出版社, 2015: 161 – 185. (KARAU H, KONWINSKI A, WENDELL P, et al. Learning Spark: Lighting-Fast Data Analysis [M]. WANG D Y, translated. Beijing: Posts and Telecom Press, 2015: 161 – 185.)

[11] 夏俊鸾, 邵赛赛. Spark Streaming: 大规模流式数据处理的新贵[J]. 程序员, 2014(2): 44 – 48. (XIA J L, SHAO S S. Spark streaming: large-scale streaming data processing upstart[J]. Programmer, 2014(2): 44 – 48.)

[12] Apache spark FAQ [EB/OL]. [2016-08-04]. https://spark.apache.org/faq.html.

[13] Apache Kafka : a high – throughput distributed messaging system [EB/OL]. [2016-01-09]. http://kafka.apache.org/documentation.html.

[14] KREPS J, NARKHED N, RAO J. Kafka: a distributed messaging system for log processing[C]// NetDB2011: Proceedings of the 6th International Workshop on Networking Meets Databases. New York: ACM, 2011: Article No. 12.

[15] 国家机关办公建筑和大型公共建筑能耗监测系统——数据上传 XML 格式文档[S]. 北京: 中华人民共和国住房和城乡建设部, 2009: 55 – 59. (Government offices and large public buildings energy consumption monitoring system — XML format for data uploading [S]. Beijing: Ministry of Housing and Urban-Rural Development of the People's Republic of China, 2009: 55 – 59.)

[16] GEORGE L. HBase 权威指南[M]. 代志远, 刘佳, 蒋杰, 译. 北京: 人民邮电出版社, 2013: 5 – 25. (GEORGE L. HBase: the Definitive Guide[M]. DAI Z Y, LIU J, JIANG J, translated. Beijing: Posts and Telecom Press, 2013: 5 – 25.)

WU Zhixue, born in 1960, Ph. D., professor. His research interests include cloud computing, streaming data processing, data mining.

(上接第 927 页)

[9] TRAPEZNIKOV K, SALIGRAMA V, CASTANON D. Multi-stage classifier design[J]. Machine Learning, 2013, 92(2): 479 – 502.

[10] KAYNAK C, ALPAYDIN E. Multistage cascading of multiple classifiers: one man's noise is another man's data[C]// ICML 2000: Proceedings of the Seventeenth International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 2000: 455 – 462.

[11] FUMERA G, ROLI F, GIACINTO G. Reject option with multiple thresholds[J]. Pattern Recognition, 2000, 33(12): 2099 – 2101.

[12] 裴胜玉, 周永权. 基于 Pareto 最优解集的多目标粒子群优化算法[J]. 计算机工程与科学, 2010, 32(11): 85 – 88. (PEI S Y, ZHOU Y Q. A multi-objective particle swarm algorithm based on the Pareto optimization solution set[J]. Computer Engineering and Science, 2010, 32(11): 85 – 88.)

[13] TEA T, BOGDAN F. Visualization of Pareto front approximations in evolutionary multiobjective optimization: a critical review and the prosection method [J]. IEEE Transactions on Evolutionary Computation, 2015, 19(2): 225 – 245.

[14] CHEN R-C, HSIEH C-H. Web page classification based on a support vector machine using a weighted vote schema[J]. Expert Systems with Applications, 2006, 31(2): 427 – 435.

[15] SEBASTIANI F. Machine learning in automated text categorization [J]. ACM Computing Surveys, 2002, 34(1): 1 – 47.

This work is partially supported by the National Natural Science Foundation of China (61501063, 61501064), the Scientific Research Foundation of Science and Technology Department of Sichuan Province (2016JY0240), the Scientific Research Foundation of Sichuan Education Department (15ZB0177).

WANG Yaqiang, born in 1984, Ph. D., lecturer. His research interests include big data, cloud computing, natural language processing, machine learning.

ZENG Qin, born in 1975, M. S., senior engineer. His research interests include big data, cloud computing, precise forecast, meteorological big data analysis.

TANG Dan, born in 1982, Ph. D., associate professor. His research interests include big data, cloud computing, coding theory.

SHU Hongping, born in 1974, Ph. D., professor. His research interests include big data, cloud computing.