基于分布流的城市照明监控系统优化策略研究

随着城市照明设施规模的日益扩大，照明设施的管理难度不断增加，丞需采用自动化技术手段提高处理问题的及时性和准确性。通过探讨城市照明监控系统的系统结构，设计了城市照明监控系统优化策略，采用的流计算框架 Storm，并配合 Flume 数据收集工具和 Kafka 消息队列，实现了数据的实时处理，利用 Redis 自带数据类型实现对象化存储。通过该优化策略，可充分满足多种的数据处理和存储需求，提高了数据的实时性和系统的横向扩展能力。

梁奕 严建鹏/国电南瑞科技股份有限公司

梁奕/高级工程师

关键词/Keywords
城市照明 · 数据采集 · Storm · 流数据处理 ·

0 引言

近年来，我国路灯建设发展迅猛，道路照明质量不断提高，促进了经济发展，方便了群众生活，对城市建设起到了非常重要的作用。随着城市照明设施规模的日益扩大，照明设施的管理难度不断增加，维护管理的工作量也成倍增加，亟需采用信息化、自动化的技术手段，提高发现问题、处理问题的及时性、准确性。

目前，国内外已经出现了一些路灯管理系统，这些系统是由过夜控制系统演变而来的，大多通过远程遥控电源柜达到控制路灯的目的，控制对象的数目小，应用规模局限性大，面对如今庞大的路灯体系而言很难做到实时监控、分析与预测。因此，为了快速将城市照明监控系统的数据汇集分析，挖掘数据内在的价值，基于分布流技术的城市照明监控系统具有较高的研究价值。

1 相关技术

1.1 Flume 采集技术

随着智慧城市工程的加速发展，城市照明监控系统的数据采集需要面对节点众多、数据庞杂和规约解析复杂等难题。Flume 是一套可用于有效地收集、聚合和搬运大量日志数据的分布式服务架构，可以用来解决分布式数据采集的问题。Flume 是由 Cloudera 公司开发的一款高性能、高可靠的分布式日志收集系统，它是一个完整的数据收集工具。Flume 采用了分层架构的设计思想，由 Agent、Collector 和 Storage 三层组成。Agent 主要由 Source、Channel 和 Sink 三部分构成，在数据流动过程中，数据从 Source 流入，Channel 作为数据流通和存储的通道，Sink 将数据发送到指定地方；Collector 的作用是将多个 Agent 的数据汇总后，加载到 Storage 中；Storage 是存储系统，既可以是普通文件，也可以是分布式存储。

1.2 Storm 框架

城市照明监控系统中存在连续的、不间断的和非结构化的数据队列，该队列满足流数据的特点，即持续流入的实时数据；处理是动态的随到随处理；处理是一次性的；处理系统无法对将要来的数据进行假设，无法预知也无法控制；有很多操作类似关系型操作等。Storm 正是用于处理流式数据的系统，Storm 是 Twitter 开发的一款分布式高容错的实时计算系统，该系统主要由 Spouts、Bolt 和 Topology 组成。Spout 主要负责对数据源进行读

中国知网 https://www.cnki.net

取，其传输方式十分可靠，在数据处理和传输过程中若出现数据处理错误，其会自动获取错误信息并将信息重新发送；Bolt 主要负责对数据的处理，其可完成指定的任务，如数据清洗、聚合、访问文件和数据库等操作，且可以将不同任务分为多个 Bolt 分别完成，达到降低耦合性的特点；Topology 为 Storm 的拓扑关系声明，其可以定义、组合 Spout 和 Bolt 对象。

1.3 Redis 存储系统

大量非结构化数据如图片、视频等信息和半结构化数据已经成为电力信息存储的重要组成部分，采用 Nosql 非关系型数据库，不需要固定的表结构，读写速度快，扩展便利，成本低廉，很好地满足了半结构化、非结构化的数据存储和访问的要求。Redis 数据库作为 Nosql 数据中的典型，是支持网络访问、基于内存和可持久化的 Key-Value 数据库，有着丰富的数据结构，可以确保多个主从服务拥有相同的数据库副本。Redis 支持持久化操作，其提供了两种持久化策略：快照方式（Snapshotting）；AOF 方式（Append-only file）。快照方式会定时将数据存入一个二进制文件中；AOF 方式会将 Redis 每次的写命令存入文件中。

2 城市照明监控系统设计

城市照明监控系统是一个连接整个城市照明设备网络的无线通信、监测控制和资源管理的综合平台，系统具有数据采集、在线监测和运行控制等基础功能，为城市照明管理行业规划、管理提供信息化的决策支持手段，同时有效地对本地设备运行状况进行监控。

2.1 层次结构

城市照明监控系统的总体层次架构分为单灯控制层、集控层、中间层和系统层四层。

如图 1 所示，单灯控制层监视各个单灯的运行状态，由一组单灯控制器组成。集控层分为两类：一类是路灯控制柜；一类是景观灯控制柜。路灯控制柜通过载波、无线等通信方式与下属的一组单灯进行通信；景观灯一般不采用单灯控制的方式（即景观灯控制柜没有下属的通信单元）；集中控制层通过集中器将单灯、控制柜等运行信息上送给主站系统，并接收处理主站系统下发的各类控制命令。中间层主要是通信层，考虑目前各地已建设路灯控制系统的现状，保证系统的适用性，通信层的架构分为两种方式：集控层通过 GPRS、3G 和 4G 等无线的方式直接接入主站；集控层通过已建的监控系统接入到主站。系统层是城市照明监控系统平台，需要与原信息化系统进行信息集成。

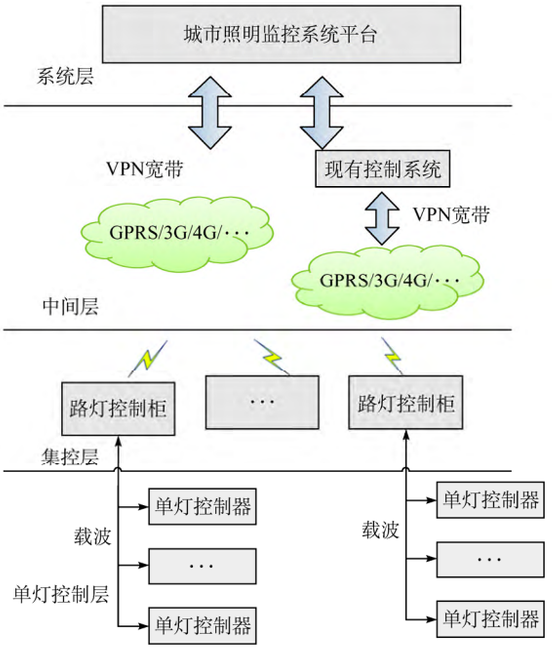

图 1 城市照明监控系统总体架构图

2.2 监控平台逻辑

城市照明监控系统逻辑结构如图 2 所示，由三个部分构成。

1）数据部分。主要由商用数据库和实时数据库组成。商用数据库又称历史数据库，历史数据库主要功能为存储数据模型、数据之间的逻辑关系、历史的数据信息以及各类记录信息（维修工单、人工情况和登录控制情况）。实时数据库主要作用是处理实时信息，分析趋势，将数据定时存放入历史数据库中。

2）应用部分。主要是面向智能监控、运行管理、资产管理、辅助决策、抢修管理和增值服务的“六大板块”服务，并从数据部分中获取数据，下发信息给采集部分。

3）采集部分。主要分为管理、规约和缓冲。管理主要功能为加载数据模型数据、设置配置信息和管理通信接口；规约主要负责对报文规约解

中国知网 https://www.cnki.net

析。其需要管理多累多批次规约程序，接收来自应用的采集数据、控制信息，发送告警信息；缓冲主要负责对通信链路的管理，对大批次报文进行缓存处理，防止数据丢失等情况。

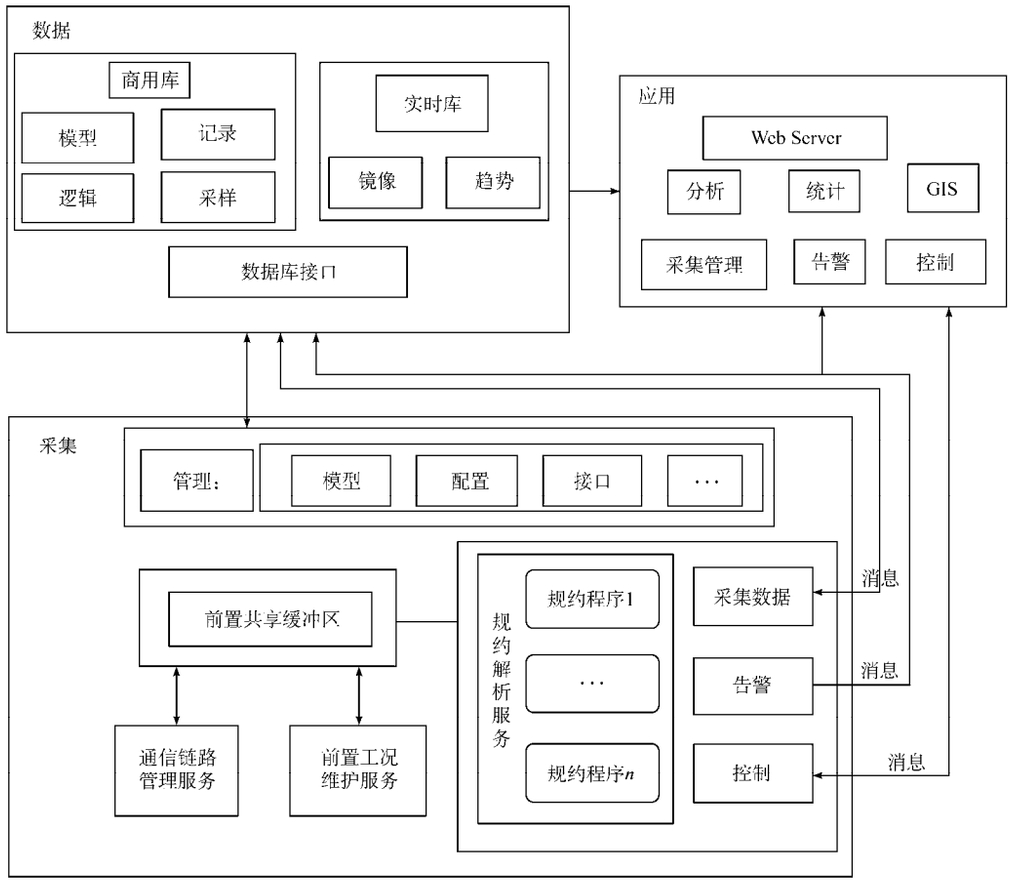

图 2　城市照明监控系统逻辑结构图

3　城市照明监控系统优化策略

城市照明监控系统的大量数据以流数据居多，需要进行实时处理、实时应用，因此优化策略采用分布式流框架来处理。主要包含基于当前流行的 Flume 框架设计海量日志采集、聚合和传输的模块，基于 Kafka 框架分布式消息缓存模块，基于 Storm 流计算框架消息处理模块。系统优化策略的数据处理流图如图 3 所示。

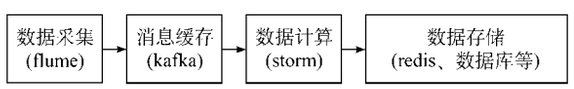

图 3　流数据处理数据流图

利用 Flume 收集报文信息，可以保证在数据采集过程中数据的完整性，提升报文汇集效率；利用 Kafka 消息缓存机制，可以灵活设定缓存数据大小，使用 Push 和 Pull 的消息推送方式，利用规约节点处理能力的差异合理分配任务，保证了数据的完整性，避免了以 Pull 形式发送给规约节点使其不堪重负的问题；利用 Storm 流处理技术，可以合理有效规约数据，其具备高容错性、快速起动和快速恢复的特点，解决了数据规约过程中数据系统容错率低的问题。

3.1 Flume 采集策略

城市照明监控系统的消息源为单灯控制器、电源柜等传感器发送的数据，这些数据通过前置机采集，以报文的形式呈现。Flume 中的 Agent 自带多类数据源，扩展性强，故在获取数据时，使用 Agent 分别接收不同消息源的实时数据，并通过 File Channel 对数据进行缓存，Sink 从 Channel 中提取数据，将数据发送至汇集 Agent 进行数据汇集，最后将 Kafka 作为 Flume 中的

中国知网　https://www.cnki.net

Collector。Flume 拓扑关系如图 4 所示。

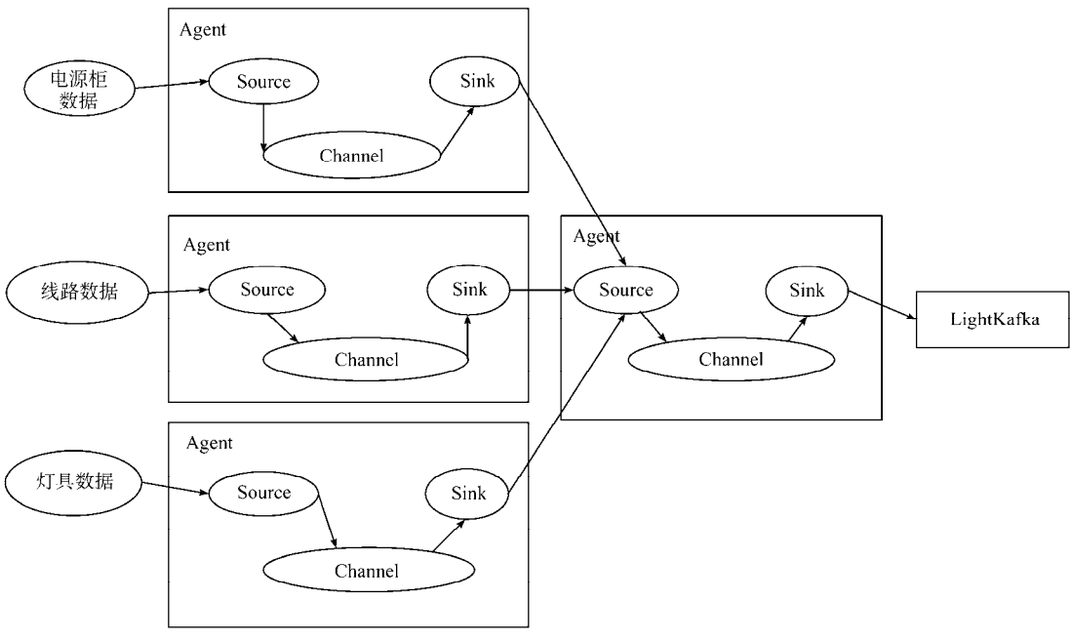

图 4　Flume 拓扑关系图

由于采集数据、报文规约解析速度不同，会出现不同程度的处理不同步，因此系统中添加一个基于 Kafka 框架的消息中间件。当出现电源柜、线路和灯具变化信息时，将 Flume 发出的数据固定存入 Broker 中。由于 Kafka 是基于 Zookeeper 集群的，Broker 和 Consumer 之间可以利用 Zookeeper 自身的负载均衡机制进行负载均衡。当某个 Broker 和 Consumer 发生了变化，Zookeeper 会通知所有其他的 Broker 和 Consumer，使其他的 Broker 和 Consumer 获取数据并记录。数据流如图 5 所示。

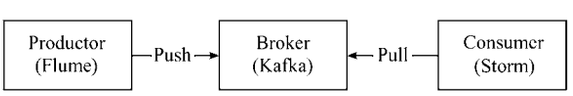

图 5　Kafka 数据流图

3.2 Storm 流计算处理策略

分布式流计算框架 Storm 是用来对数据进行报文解析、数据规约。其过程为：将数据规约过程放入 Bolts 中，将 Kafka 中的数据作为 Spout，为 Bolts 提供消息。同时，Bolts 根据获得的数据情况对 Redis 内存数据库和关系型数据库进行读写操作。实时采集数据的 Storm 分布式处理过程如图 6 所示。

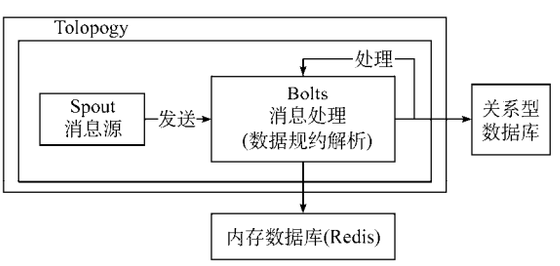

图 6　Storm 分布式处理过程

Spout 数据中主要包括电源柜、线路和灯具的实时报文数据，Spout 为从消息缓存 Kafka 中读到的数据，将 Spout 中的数据分别由 MsgStoreBolt 与 SplitBolt 读取。MsgStoreBolt 是将原报文保存起来，SplitBolt 是将消息分割，把消息头与消息体分离，随后 LineBolt、LampBolt 和 CubicleBolt 分别读取属于自己的数据，并将读取的数据进行规约解析后形成对象，将对象序列化后生成 Redis 脚本存入内存数据库中。流数据处理流程如图 7 所示。

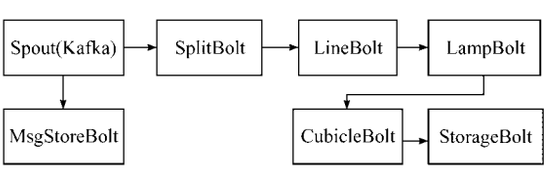

图 7　流数据处理流程图

中国知网 https://www.cnki.net

3.3 Redis 存储策略

由数据结构映射器将用户指定的数据结构从缓存中获取，并将其分析组合成为包含数据与结构的对象，最终将该对象存储到 Redis 数据库中。其主要的工作流程如图 8 所示。

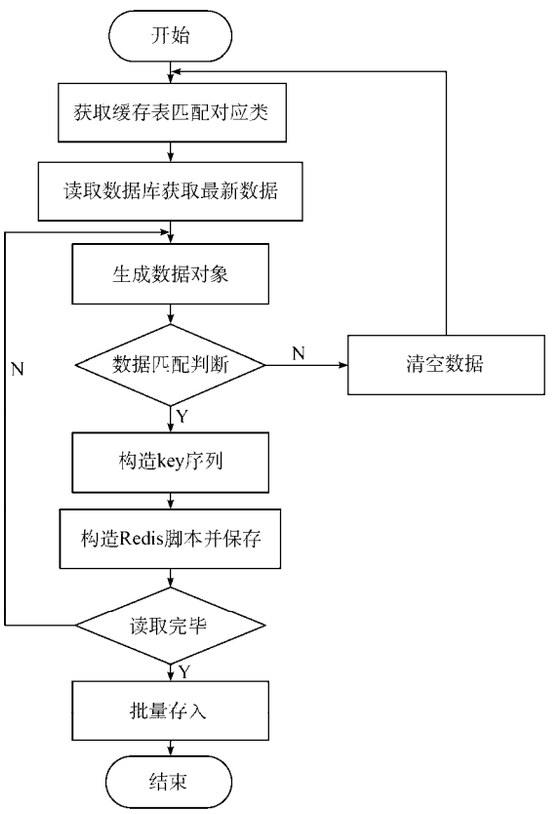

图 8　数据缓存工作流程图

首先获取缓存表，根据需要缓存表的数据结构，建立对应数据结构类。从数据库中将对应表中的数据获取，生成数据对象（电源柜、线路和灯具等以个为单位建立），检测其是否为最新数据，验证其数据结构是否合理，若合理则清空数据重新开始配对类；若成功则获取其中的 ID 与编号组成唯一的 key 值并且将对象属性与对应值生成 Redis 存储脚本，存入临时文件中。依次执行，直至该类型数据全部读取完成，通过 Redis 的 Client 连接数据库，执行存储脚本（批量操作）。

4　结束语

本文探讨了城市照明监控系统的系统结构，并针对系统中存在的路灯电源柜、单灯控制器节点众多、数据量大、并发数多和监控实时性强等特点，设计了城市照明监控系统优化策略，采用当前较为流行的流计算框架 Storm 并配合 Flume 数据收集工具和 Kafka 消息队列，实现了数据的实时处理；采用内存数据库解决传统数据库对扩展能力不强的缺点，搭建 Redis 数据库，设计数据结构映射器，利用其自带数据类型实现对象化存储。通过该系统优化策略，可充分满足多种业务场景下的数据处理和存储需求，并在保证数据安全性的前提下，提高了数据的实时性和系统的横向扩展能力。

参考文献

[1] Novak，T，Pollhammer K，Zeilinger H．et al．Intelligent streetlight management in smart city [J]．Emerging Technology and Factory Automation（ETFA），2014，1-8．

[2] Yusoff Y．M，Rosli R，Karnaluddin M．U．et al．Towards smart street lighting system in Malaysia [J]．Wireless Technology and Applications（ISWTA），2013，301-305．

[3] Atici C，Occelebi T，Lukkien J．J．System architecture for roadlighting[J]．2011 IEEE International Conference on ConsumerElectronics（ICCE），2011，681-682．

[4] Zhao L，Shi W．Research on optimization of application model based on storm[C]//Software Engineering and Service Science（ICSESS），2014 5th IEEE International Conference on．IEEE，2014：248-250．

[5] 杜鹏，陶洪铸．面向多应用的通用数据采集技术方案[J]．电力系统自动化，2015，01：26-30．

[6] 潘洪志．高性能 NOSQL 存储系统的研究与实现[D]．长春：吉林大学，2014．

———————

作者简介

梁奕（1977-），男，硕士，高级工程师，主要从事工作领域：电力系统自动化及轨道交通电气化。

（收稿日期：2015-12-10）EA

（编辑：史海疆）

中国知网　https://www.cnki.net