

基于分布式流式计算的蛋鸡养殖实时监测与预警系统

陈红茜^{1,2} 滕光辉¹ 邱小彬² 孟超英³ 曹晏飞⁴ 汪成³

(1. 中国农业大学水利与土木工程学院, 北京 100083; 2. 中国农业大学网络中心, 北京 100083;

3. 中国农业大学信息与电气工程学院, 北京 100083; 4. 西北农林科技大学园艺学院, 陕西杨凌 712100)

摘要: 为实现蛋鸡养殖生产过程参数实时监测与预警, 研发了基于分布式流式计算框架 Data-Canal 的蛋鸡养殖实时监测与预警系统。Data-Canal 是面向数据流的分布式计算框架, 使用控制流集中、数据流分散的模式, 以分布式文件系统为中间结果的存储, 支持异地多数据源的实时采集和处理。系统以 Data-Canal 为基础设施, 在具有一定扩展性的情况下, 保证实时性。系统采用 Browser/Server 模式, 用户通过浏览器即可访问, 提升了信息共享的便捷性。系统实现了规模化蛋鸡生产过程实时数据采集与展示、生产信息管理、实时预警、决策分析和系统管理功能, 对蛋鸡养殖全生命周期进行了全方位的管理。运行效果表明, 该系统可以解决规模化蛋鸡生产过程中产生海量数据信息化和实时处理问题, 在部署 8 台机器的情况下, Data-Canal 集群的处理能力峰值达到 160 MB/s, 延迟在分钟级别, 在线上实验环境中, Data-Canal 集群每天处理约 25 GB 的数据, 而且系统后期维护和升级都极为便利。

关键词: 蛋鸡; 流式计算; 分布式; 实时监测; 预警系统

中图分类号: S815; TP391 **文献标识码:** A **文章编号:** 1000-1298(2016)01-0252-08

Real-time Monitoring and Early Warning System Based on Stream Computing for Laying Hens Raise

Chen Hongqian^{1,2} Teng Guanghui¹ Qiu Xiaobin² Meng Chaoying³ Cao Yanfei⁴ Wang Cheng³

(1. College of Water Resources and Civil Engineering, China Agricultural University, Beijing 100083, China

2. Network Center, China Agricultural University, Beijing 100083, China

3. College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China

4. College of Horticulture, Northwest A&F University, Yangling, Shaanxi 712100, China)

Abstract: With the rapid development of computer technology, it's possible to process multi-type and mass data real-timely. In order to achieve real-time monitoring and early warning in laying hen raise, a system based on a distributed streaming computing framework "Data-Canal" was developed. Data-Canal is a data flow oriented distributed computing framework with the control flow and data flow dispersion model, which using the distributed file system as the storage of intermediate result, supporting real-time acquisition and processing multiple remote data sources. Data-Canal is the basic facility of the system, which ensures the extend and real-time processing of system. The system was developed in Browser/Server mode. The users can access the system through the browser, which improves the convenience of informatization sharing. The system realizes real-time data acquisition and display, production information management, early warning, decision analysis, and system management functions. The result shows that the system solves the problem of information and real-time processing of mass data in laying hens raise. In the case of eight machines, the highest throughput of Data-Canal cluster reaches 160 MB/s, and the delay is at the minute level. In the online experiment, the Data-Canal cluster processed about 25 GB of data every day. The maintenance and upgrade of system are convenient. Further research will be done on the mobile client.

Key words: laying hens; stream computing; distributed; real-time monitoring; early warning system

收稿日期: 2015-07-13 修回日期: 2015-11-10

基金项目: "十二五" 国家科技支撑计划项目(2014BAD08B05)

作者简介: 陈红茜(1987—), 女, 博士生, 工程师, 主要从事农业信息化与设施养殖环境研究, E-mail: chen hongqian@cau.edu.cn

通信作者: 滕光辉(1963—), 男, 教授, 博士生导师, 主要从事设施环境检测与信息技术研究, E-mail: futong@cau.edu.cn

引言

我国蛋鸡养殖正逐步由小规模向集约化、工厂化发展,如何对养殖过程中产生的现场数据进行实时处理和分析,为蛋鸡规模化养殖的科学管理提供服务和支撑,是亟需解决的重要技术问题^[1-2]。随着物联网技术的不断发展与成熟,实时获取蛋鸡养殖生产过程中的各项生产数据及生产环境数据成为可能,在设施农业环境监测和畜禽精细化养殖监测网络等方面应用广泛^[3-5]。但对于规模化蛋鸡养殖企业来说,通过物联网技术获取的现场数据大多没有得到实时处理,仅停留在初步展示环节,主要问题在于:第一,由于缺乏相关算法和专家决策规则,虽然利用了传感器技术,但是采集到的数据主要还是一种展示或查询统计,没有真正用于科学决策和实时预警^[6];第二,现场数据并没有进行实时处理和分析,依然停留在事故发生之后由人工进行查看,这样就存在对突发时间的反应不及时、人工监控工作量较大等问题^[7]。

规模化蛋鸡场的现场数据类型多样,通常分为文本型和多媒体型,大多以流数据的形态出现,流数据与传统数据有着截然不同的特点,即数据产生速度快、规模宏大,使得许多面向传统数据的计算框架很难直接应用于流数据环境^[8-13]。而且现场数据表现出价值密度低、分布不规律、信息隐藏程度深等特征,这就要求数据计算系统具备实时性强、高性能、分布式、易扩展等特征。

因此,本研究在前期底层硬件和设施农业物联网研究基础上,研发分布式流式计算框架 Data-Canal,结合蛋鸡规模化养殖特点研发蛋鸡养殖实时监测与预警系统,目前该系统已经在中国农业大学上庄实验站和网络中心机房实施。

1 分布式流式计算 Data-Canal 概述

目前业界主流的分布式计算系统有 Apache 的 Hadoop、Yahoo 的 S4 以及 Twitter 的 Storm^[14-17]。Data-Canal 与它们的区别在于:

(1) Hadoop 是非常成熟的分布式解决方案,但它是批处理系统,无法保证实时性^[18-19]。

(2) S4 是分布式流式系统,但是可靠性差,在节点故障时,允许数据丢失。而且在集群运行时,无法动态增、删节点。

(3) Storm 也是分布式流式系统,相对于 S4,支持动态增、删节点,具有一定的扩展性。但是,在 failover 时,同样会有数据丢失的风险。同时,作为全局控制节点的 Nimbus 存在单点的风险^[20]。

Data-Canal 在舍弃小部分实时性的情况下,大幅提高可靠性和扩展性,系统的延迟保证在分钟级别,具有以下特点:简单、高可用、高吞吐、管理部署简单、具有一定的容错性。

Data-Canal 是面向数据流的分布式计算框架,不同类型的 Worker 可以消费输入 Task,并生成输出 Task,每个类型的 Worker 完成一个独立的功能,多种 Worker 就构成了一个多级分布式处理流水线。

Data-Canal 使用控制流集中、数据流分散的模型,不像传统的流式计算框架使用 RPC (Remote procedure call) 进行通信,该框架使用分布式文件系统作为中间结果的存储。这种方式可能会增加系统的整体延迟,但是却增强了可靠性和简单性。只要输入任务对应的文件没有被删除,数据处理流水线中的每一个步骤都是可以重试的,大大提升了容错性。图 1 是 Data-Canal 的拓扑,由一系列执行不同任务的 Worker 组成。

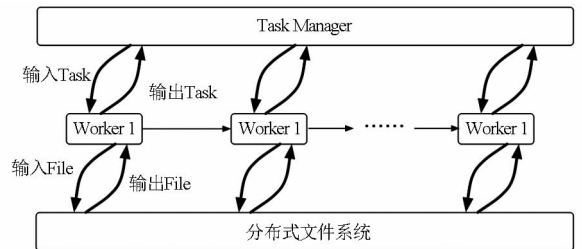


图 1 Data-Canal 的拓扑图

Fig. 1 Topology of Data-Canal

如图 1 所示,每个类型的 Worker 可以有多个相同的副本以避免单点,同时可以处理工作负载。该框架主要由以下组件组成:

(1) Task Manager: 跟踪管理所有的 Task。Task Manager 是被动的,不关心 Worker 的状态。Task Manager 相当于 RPC server,接收并处理 Worker 的请求。Task 被组织成多个 Group,每个 Group 对应一个类型的任务,会被一类 Worker 处理。实际上,Task Manager 可以看成是多个 Task 队列的集合,每个 Task Group 对应一个队列。

(2) Worker: 一系列 Worker 会被部署,用于处理不同类型的 Task。Worker 向 Task Manager 请求 Task,当任务处理成功后会创建新的 Task 并提交给 Task Manager。Worker 的工作模式是一个循环,不停的获取、处理并提交 Task。

(3) Bundler: 一种通用 Worker,用于对数据源进行打包,将实时的数据流打包成多个 Bundle,可以按照时间维度或者记录条数进行打包。Bundler 相当于 Data-Canal 的入口。

(4) File Deleter: 另一种通用 Worker。由于使用

文件作为中间存储,同时保证只在 Task 成功处理后再删除文件。所以,提供 File Deleter 用于删除中间临时文件。在一个 Worker 正确处理完 Task 后,会提交一个 File Delete 的 Task,用于删除对应的输入文件。系统会有多个 Worker 专门用于消费 File Delete 的 Task,删除中间文件。

Data-Canal 提供了一些通用算子 Worker,用于完成一些通用计算:

(1)Partitioner:根据 Partition id 将输入的数据流进行分片,分片规则可以自定义。一般情况下,Partitioner 的输入是一条数据流,输出是多条数据流。

(2)Merger:将输入的多条相同数据流进行合并,同样,合并的规则也是可以自定义的。

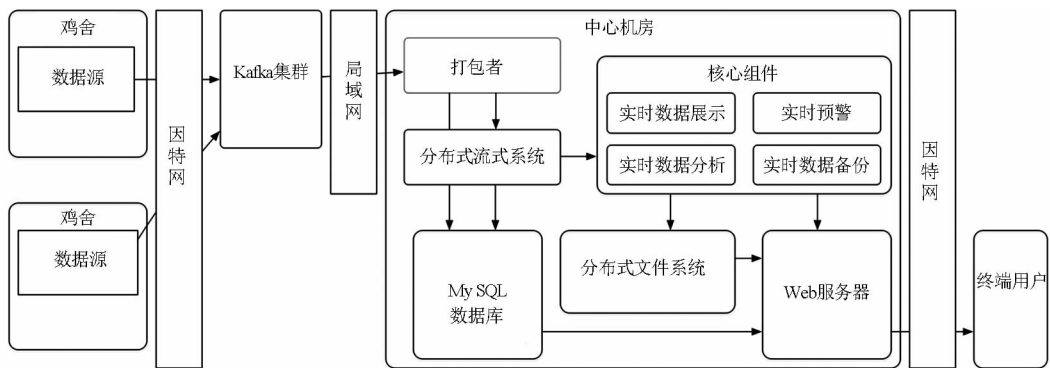


图2 系统架构

Fig. 2 System architecture

由于鸡舍网络环境不稳定,为了容错,各个鸡舍和中心机房的通信采用分布式消息中间件 Kafka。Kafka 具有分布式、高吞吐、可扩展和持久化等特点,能针对一个 Topic 的队列进行多个备份,保证不丢失数据,同时多个备份支持多个消费端同时消费,可以大幅提高系统吞吐量。

鸡舍节点首先将采集到的数据存储到本地 MySQL 数据库,同时部署数据监控程序监视数据库的变化,并将更改的数据异步地发送到 Kafka 集群。中心机房部署 Bundler 集群从 Kafka 集群订阅数据,并打包成 bundle,然后生成 Task 提交到 Data-Canal。Data-Canal 主要实现以下功能:①环境、生产数据实时入库,存储到 MySQL 数据库中。②音频、视频和图像实时存储到分布式文件系统中。③各类数据的实时备份。④数据分析,包括对生产数据的盈亏分析和针对图像、音频数据的行为分析。

通过 Data-Canal 将实时采集和处理后的数据分别存储至 MySQL 数据和分布式文件系统。中心机房部署 Web 服务器 Nginx,以及相应的应用服务器,终端用户通过浏览器访问 Web 服务器使用本系统。

2.2 系统功能设计

本系统功能分为 5 部分(图 3):

(3)Joiner:将输入的多条不同数据流进行拼接,在时间同步的情况下,多条数据流之间相互参考产生输出,具体拼接的规则也是自定义的。

通过通用算子,比如 Partitioner 和 Merger 即可简单实现一个分布式排序,大大提升开发效率。

2 系统设计与实现

2.1 系统总体架构

为了支持异地多鸡舍相关数据的实时采集和处理,整体系统采用分布式架构。各个鸡舍作为数据源,将采集到的生产过程参数、生长环境数据以及现场音、视频异步发送至中心机房,然后中心机房实时接收并进行处理,如图 2 所示。

(1)实时数据采集与展示:该功能实现对鸡舍现场环境、音视频、蛋鸡体质量、水电消耗等实时数据的采集,这些数据由架设各个鸡舍内外的监控节点采集。监控节点主要采用无线方式进行数据的采集和传输,依据鸡舍所在地不同的网络通信条件,系统可以通过 2G、3G、4G、无线 Wi-Fi、无线传感网络(如 ZigBee)等多类型网络技术异构融合,进一步提高网络传输的安全性和广泛应用性。采集的实时流式数据会进入 Data-Canal 中,完成对数据的实时处理和分析等,这些原始数据将作为蛋鸡养殖分析及预警的基础数据。不仅如此,系统还提供这些实时数据的查看,用户可以更直观地看到蛋鸡养殖现场情况。

(2)生产信息管理:针对规模化鸡场难于管理和操作,日常管理工作量大等问题,其功能包括鸡舍管理、日报表、转群计划、免疫计划、饲料管理、设备维护等。鸡场一线工作人员录入生产过程相关信息,可以通过计算机或手机客户端填写上传,由鸡场管理员进行审核。通过生产过程信息化更加方便了鸡场的管理,提高了工作效率,节省了管理成本,增加了鸡场效益。

(3)实时预警:本系统主要涉及 4 方面的预警:

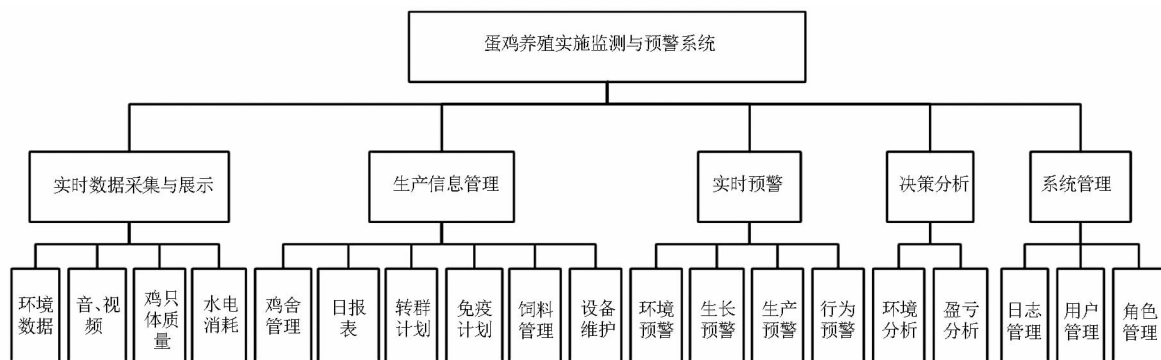


图 3 系统功能

Fig.3 Functions of system

环境预警、生长预警、生产预警和行为预警。环境预警是对鸡舍内外的环境异常进行告警,根据具体参数的不同,有以下策略:阈值、段累积量、段比例等。生长预警和生产预警参照《海兰褐蛋鸡性能标准手册 2011》中不同日龄对应的标准体质量和每日耗水、耗料标准。生长预警是针对采集到的单只蛋鸡的平均体质量与标准体质量进行比对进行预警。生产预警指采集各鸡舍耗水总量和耗料总量,按鸡只总数平均之后得到单只鸡的日均耗水量和耗料量与手册中的标准值进行比对,如果出现耗水量和耗料量偏离区间较大,系统将自动预警。行为预警通过拍摄蛋鸡图像和录制蛋鸡声音,利用图像和音频分析,识别蛋鸡异常行为^[21-22]。预警方式可以通过系统推送消息,向管理员发送邮件和短信。

(4)决策分析:该功能是对系统中生产数据和环境数据进行分析,将结果展示给用户。环境分析可针对不同的环境指标和分析条件,生成相应结果。盈亏分析根据各鸡舍的收入支出指标,综合分析构建鸡舍日盈亏曲线公式,绘制鸡舍每日盈亏曲线。鸡场管理人员可通过查看每日盈亏情况,及时调整生产。

(5)系统管理:该功能包括日志管理、用户管理和角色管理。日志管理会记录所有用户的操作记录,确保信息的可溯源性,保证系统的安全可靠性。用户管理完成对系统用户的添加、删除和修改,并为每个用户指定相应的角色。角色管理是对不同系统角色赋予不同的管理权限,实现系统的权限控制。

2.3 系统实现

开源软件具有自由分发、自由集成、公布源代码等优点,为了最大限度降低系统开发和运行成本,提高质量,系统应用了诸多开源框架,并对这些开源框架进行了扩展、定制。系统分成 4 部分:

(1)部署在鸡舍的数据采集程序。由 LabView 编写,将各种传感器采集到的数据写入数据库,音、视频文件写入文件系统。

(2)部署在鸡舍的数据发送程序,主要用于识别数据库的变更以及音视频文件,并将变更的数据实时发布到 Kafka。这部分程序通过 Python 编程语言实现。

(3)部署到中心机房的分布式流式计算 Data-Canal 集群,主要由 C++ 语言实现,其依赖的分布式文件系统使用的是 Hadoop 体系中的 HDFS。

(4)部署到中心机房的 Web 系统,供用户通过浏览器或者 APP 访问。后端实现语言选择的是 PHP,前端 Web 框架使用开源的 AngularJS 和 Bootstrap,Web 服务器选择的是 Nginx,数据库使用的是 MySQL,并通过开源的 redis 实现缓存。

2.4 流式计算实现

基于 Data-Canal 框架的实时流式处理的结构如图 4 所示。部署 2 个 Task Manager,一个作为 master,控制整个流式处理的每一步骤;另一个作为 slave,同步复制 master,用于冷备份,在 master 不可用时,请求切换为 slave 处理。系统运行时状态由 Task Manager 维护。

系统中部署了一系列的 Worker,每一个 Worker

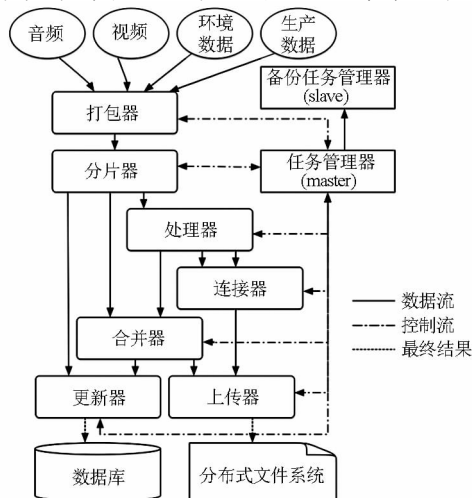


图 4 系统实现

Fig.4 System implementation

用于完成一种任务。每个类型的 Worker 会有多个副本同时运行,以分摊工作负载,提高系统吞吐量。每个 Worker 通过远程过程调用 RPC 与 Task Manager 交互,确定下一步的工作。当它完成后,删除输入 Task,并产生下游 Worker 的输入 Task。

一个典型 Worker 的代码如下:

```
TaskManagerClient workflow;
// Worker 工作主循环
while (true) {
    // 存放该 Worker 的输入 Task
    vector<Task * > input_tasks;
    // 请求 TaskManager, 获取其输入 Task
    workflow.QueryAndOwn(group, limit, lease
_seconds, &input_tasks);
    // 存放 Worker 输出文件及对应的 File
Delete Task
    vector<Task * > delete_output_file_tasks;
    vector<File * > output_files;
    // 创建输出文件及对应的 File Delete Task
    CreateFiles (&workflow, &output_files,
&delete_output_file_tasks);
    // 存放输入文件对应的 File Delete Task
    vector<Task * > delete_input_file_tasks;
    UpdateDeleteInputfileTasks (input_tasks,
&delete_input_file_tasks);
    // 存放 Worker 输出 Task
    vector<Task * > output_tasks;
    UpdateOutputTasks (output_file, &output_
tasks);
    // 存放要删除的 Task 集合
    vector<Task * > tasks_to_delete;
    // 将输入 Task 和输出文件的 File Delete
Task 添加到要删除的 Task 集合
    AppendTo(input_tasks, &tasks_to_delete);
    AppendTo(delete_output_file_tasks, &tasks
_to_delete);
    // 存放要更新的 Task 集合
    vector<Tasks * > tasks_to_update;
    // 将输出 Task 和输入文件的 File Delete
Task 添加到需要更新的 Task 集合
    AppendTo (output_tasks, &tasks_to_
update);
    AppendTo(delete_input_file_tasks, &tasks_
to_update);
    // 向 Task Manager 提交请求
    workflow.Update (&tasks_to_delete, &tasks_
```

```
to_update, NULL);
}
```

图 4 描述了蛋鸡生产过程流式数据处理的核心组件。如图所示,虚实线表示控制流,Task Manager 只与 Worker 交换控制信息;实线表示数据流,数据在 Worker 之间交换,通过分布式文件系统作为中间媒介;虚线表示最终结果。Bundler 的任务是将一系列连续的数据打包成处理单元,称为 bundle。一般情况下,1 个 bundle 中包含 1000 条记录,并且对 1 个 bundle 的操作是原子操作。以 bundle 为处理单元,而不是一条记录,主要目的是提升系统吞吐量,并且简化错误恢复。系统包含 4 类数据:环境、生产、视频和音频数据,对于每 1 类数据都对应 Kafka 中的 1 个 Topic。Bundler 订阅 Kafka 中对应的 Topic,输出 4 种 Task: VideoBundleTask、AudioBundleTask、EnvironmentBundleTask 和 ProductionBundleTask。

Partitioner 对输入数据流进行分片。由于数据量巨大,单机无法处理,Partitioner 按照业务逻辑对数据流进行分片,达到单机可以处理的粒度。输入数据按照“鸡场号+鸡舍号”进行分片,保证同一个鸡舍的数据路由到同一个下游,以便可以进行全局计算,如同一个鸡场的总值或均值等。Partitioner 会输出 3 类 Task: 第 1 类是输出给 Processor 的 Partitioner4ProcessorTask,包含 Processor 需要处理的源数据;第 2 类是输出给 Merger 的 Partitioner4MergerTask,只包含主键信息,用于对 Processor 输出的 Task 进行合并;第 3 类是输出给 Updater 的 Partitioner4UpdaterTask,只包含数据,用于记录实时数据。

Processor 进行业务处理,按照输入 Task 的不同,进行的计算逻辑不同。对于环境数据,进行阶段 5 min 均值、最值、多次迭代的数值计算,按照不同的环境指标生成相应曲线;同时针对系统中设置的各项环境指标,按照不同的策略进行实时预警。对于盈亏曲线,每天在固定时间按照生产数据对各鸡舍进行盈亏计算。对于音、视频采用分时变频技术,即根据蛋鸡的活动程度决定图像和音频的采集频率,经过压缩后,进入 Data-Canal 执行行为识别算法,提取图像和音频特征值,识别蛋鸡典型行为^[22,23]。系统根据算法识别结果进行显示或预警。

Joiner 对输入的 2 条数据流按照一定规则进行拼接。在本系统中用于实现音、视频同步参考的蛋鸡行为分析。输入的 2 条数据流分别是经过 Processor 处理过的音频流和视频流,输出的是给 Uploader 的 Joiner4UploaderTask。

Merger 用于对 Processor 处理的输出数据进行合并,输入 Task 包括 Partitioner 输出的包含主键信息的 Task,以及 Processor 处理过的输出 Task。Merger 的输出 Task 有 2 类,第 1 类是输出给 Updater 的 Merger4UpdaterTask,第 2 类是输出给 Uploader 的 Merger4UploaderTask。

Updater 和 Uploader 相当于是流式系统的输出 Worker。Updater 将 Partitioner 输出的实时数据以及 Merger 合并过的处理数据写入 MySQL 数据库。Uploader 将处理过的图像和音频数据按照一定规则上传到分布式文件系统。

使用 8 台机器对 Data-Canal 进行性能测试, Bundler 以 10000 个记录打包,Data-Canal 的吞吐量达到 156.99 MB/s,延迟在分钟级别,具体结果见表 1、2,表 1 为系统整体吞吐量以及各模块吞吐量,表 2 为执行一个 Task 在系统中的处理时间分布。

表 1 系统吞吐量

Tab.1 System throughput MB/s

系统吞吐量	Partitioner	Processor	Merger	Updater
156.99	241.6	160.1	170.3	168.8

表 2 系统延迟

Tab.2 System delay s

系统延迟	Partitioner	Processor	Merger	Updater
67.2	7.8	36.4	14.1	8.9

2.5 实验结果分析

以上庄实验站的 4 个鸡舍作为实验对象,其中每个鸡舍内分别部署 3 个摄像头,1 个麦克风,温湿度、氨气等传感器各 1 个。鸡舍外会增加气象站用于检测天气情况。

采集到的环境数据首先被存放到鸡舍本地 MySQL 数据库中,由数据发送程序以监控 binlog 的方式,实时监控数据库中增量数据的变化,然后将这些变更实时发送到远端。为了提高压缩比,数据发送程序会以 1 s 的粒度将 1 min 的数据进行打包。

对于音、视频等多媒体数据,首先以 3 min 为单位进行切分,存储到文件系统的固定路径,路径格式为: / [鸡舍号] / [日期] / [时间戳]. avi, 比如: /001/20150822/0800. avi。通过 Linux 操作系统提供的 Inotify 机制,可以监控一个文件路径的变化,当创建新文件时,回调注册的处理函数将新增文件压缩后,发送到远端。音、视频采集频率使用分时变频技术,夜间(当日 21:00—次日 05:00)在连续录制音频的同时,每 30 min 抓拍一幅图像,日间(05:00—21:00)自动调整为每 2 s 抓拍一幅图像^[23]。

按照上述数据采集频率,Data-Canal 集群部署

8 个节点,平均每天处理的数据量大约是 25 GB,延迟平均在 30 s 左右。

3 讨论

系统开发完成后,部署在中国农业大学上庄实验站和网络中心机房服务器上投入实际使用,为蛋鸡养殖人员、鸡场管理人员提供服务。利用计算机技术,实时监测鸡舍内外的环境、生产和音、视频数据,对养殖人员现场管理和企业管理者经营决策有重要的意义。

(1) 信息传递的实时性。相关人员可以实时地获取鸡舍的状态信息,在突发情况发生前预警,可以避免突发情况发生或者降低其影响。

(2) 基础数据支持。系统采集到的基础数据可以为后续的大数据分析、处理提供支持。

(3) 音、视频处理的自动化和实时化。在本系统部署之前,音、视频的采集需要通过人工拷贝、远程传输等形式,存在低效、无法获取所有有效数据等缺点。通过引入 Data-Canal 框架,将音、视频数据获取、处理自动化,大大提高了处理效率。

授权用户访问本系统,只需要连上互联网,输入本系统的网址、用户名及密码即可,本系统对多种内核的浏览器都是兼容的。进入系统首页后可以清楚看到各鸡舍的当前情况,如图 5a 为系统首页,包括存栏数、死淘数、产蛋、耗水、耗料等,点击鸡舍下方的功能菜单可以进入不同的功能页面,点击舍外可以看到鸡舍外部侧视图以及舍外的环境数据,进入舍内可以看到舍内的环境数据、舍内的实时视频等。预警记录中会记录系统自动发出的预警消息,管理员还能通过注册在系统中的邮箱和手机接收到邮件或短信通知,如图 5b 所示。



图 5 应用效果

Fig. 5 Application result

本研究以规模化蛋鸡养殖为对象,使用研发的分布式流式计算框架 Data-Canal,设计并实现了蛋鸡养殖实时监测与预警系统。结果显示,该系统的开发和运行成本明显低于同类型的商业化系统。针

对上庄实验站初步实现了对蛋鸡生产过程全周期的实时监测和预警。系统的后期维护和升级都极为便利,系统管理员只需要更新服务器端的程序,用户即可访问最新系统。用户只需要通过浏览器即可随时获取到最新的现场数据,若有异常情况,系统还可以通过短信和邮件进行预警。

4 结论

(1)系统使用研发的分布式流式计算框架

Data-Canal 实现了对现场流式数据的实时处理和分析,保证了系统的时效性。

(2)系统采用 B/S 模式和分层思想进行设计,实现了系统的高内聚、低耦合,充分考虑了未来扩展的可能性,有利于系统的维护和升级。

(3)通过本系统的功能模块可以有效的全方位实时监测蛋鸡整个生长过程,并能针对异常情况实时预警。

参 考 文 献

- 李凯年, 逯德山. “工厂化”、集约化养殖方式何以受到质疑(续二)——“工厂化”养殖对动物健康与动物福利的危害及选择[J]. 中国动物保健, 2008(10): 15-21.
- McCarthy U, Brennan L, Ward S, et al. Enhanced efficiencies in the poultry industry via real-time monitoring and cloud-enabled tracking[C]//The 6th European Conference on Precision Livestock Farming, 2013.
- 葛文杰, 赵春江. 农业物联网研究与应用现状及发展对策研究[J]. 农业机械学报, 2014, 45(7): 222-231.
Ge Wenjie, Zhao Chunjiang. State-of-the-art and developing strategies of agricultural internet of things[J]. Transactions of the Chinese Society for Agricultural Machinery, 2014, 45(7): 222-231. (in Chinese)
- 熊本海, 杨振刚, 杨亮, 等. 中国畜牧业物联网技术应用研究进展[J]. 农业工程学报, 2015, 31(增刊1): 237-246.
Xiong Benhai, Yang Zhengang, Yang Liang, et al. Review on application of Internet of Things technology in animal husbandry in China [J]. Transactions of the CSAE, 2015, 31(Supp. 1): 237-246. (in Chinese)
- 何勇, 聂鹏程, 刘飞. 农业物联网与传感器研究进展[J]. 农业机械学报, 2013, 44(10): 216-226.
He Yong, Nie Pengcheng, Liu Fei. Advancement and trend of internet of things in agriculture and sensing instrument [J]. Transactions of the Chinese Society for Agricultural Machinery, 2013, 44(10): 216-226. (in Chinese)
- 阎晓军, 王维瑞, 梁建平. 北京市设施农业物联网应用模式构建[J]. 农业工程学报, 2012, 28(4): 149-154.
Yan Xiaojun, Wang Weirui, Liang Jianping. Application mode construction of internet of things(IOT) for facility agriculture in Beijing[J]. Transactions of the CSAE, 2012, 28(4): 149-154. (in Chinese)
- 劳凤丹. 规模化蛋鸡舍蛋鸡生产过程实时监控体系研究[D]. 北京: 中国农业大学, 2013.
- 朱辉生, 汪卫, 施伯乐. 基于情节规则匹配的数据流预测[J]. 软件学报, 2012, 23(5): 1183-1194.
Zhu Huisheng, Wang Wei, Shi Bole. Data stream prediction based on episode rule matching [J]. Journal of Software, 2012, 23(5): 1183-1194. (in Chinese)
- Babcock B, Babu S, Datar M, et al. Models and issues in data streams[C]//Papa L. Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Madison: ACM Press, 2002: 1-16.
- Julisch K, Dacier M. Mining intrusion detection alarms for actionable knowledge[C]//Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. New York: ACM Press, 2002: 366-375.
- Ng A, Fu A W. Mining frequent episodes for relating financial events and stock trends[C]//Whang K Y, Jeon J, Shim K, Srivastava J. Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2003: 27-39.
- Cortes C, Fisher K, Pregibon D, et al. Hancock: a language for extracting signatures from data streams[C]//Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. New York: ACM Press, 2000: 9-17.
- Fletcher A K, Rangan S, Goyal V K. Estimation from lossy sensor data: jump linear modeling and Kalman filtering[C]//Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, 2004: 251-258.
- 孟小锋, 慈祥. 大数据管理: 概念、技术与挑战[J]. 计算机研究与发展, 2013, 50(1): 146-169.
Meng Xiaofeng, Ci Xiang. Big data management: concepts, techniques and challenges [J]. Journal of Computer Research and Development, 2013, 50(1): 146-169. (in Chinese)
- Lim L, Misra A, Mo T L. Adaptive data acquisition strategies for energy-efficient, smartphone-based, continuous processing of sensor streams[J]. Distributed and Parallel Databases, 2013, 31(2): 321-351.
- Li B D, Mazur E, Diao Y L. SCALLA: a platform for scalable one-pass analytics using MapReduce[J]. ACM Transactions on Database Systems, 2012, 37(4): 1-43.
- Yang D, Rundensteiner E A, Ward M. Mining neighbor-based patterns in data streams[J]. Information Systems, 2013, 38(3):

- 331 – 350.
- 18 Ghemawat S, Gobiuff H, Leung S-T. The Google file system[J]. Association for Computing Machinery, 2003,37(5): 29 – 43.
- 19 Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008,51(1): 107 – 113.
- 20 程学旗,靳小龙,王元卓,等. 大数据系统和分析技术综述[J]. 软件学报,2014,25(9):1889 – 1908.
Cheng Xueqi, Jin Xiaolong, Wang Yuanzhuo, et al. Survey on big data system and analytic technology[J]. Journal of Software, 2014,25(9):1889 – 1908. (in Chinese)
- 21 劳凤丹,滕光辉,李军,等. 机器视觉识别单只蛋鸡行为的方法[J]. 农业工程学报,2012,28(24):157 – 163.
Lao Fengdan, Teng Guanghui, Li Jun, et al. Behavior recognition method for individual laying hen based on computer vision[J]. Transactions of the CSAE, 2012, 28(24): 157 – 163. (in Chinese)
- 22 曹晏飞,陈红茜,滕光辉,等. 基于功率谱密度的蛋鸡声音检测方法[J]. 农业机械学报,2015,46(2):276 – 280.
Cao Yanfei, Chen Hongqian, Teng Guanghui, et al. Detection method of laying hens' vocalization based on power spectral density[J]. Transactions of the Chinese Society for Agricultural Machinery,2015,46(2):276 – 280. (in Chinese)
- 23 劳凤丹,滕光辉,余礼根,等. 蛋鸡养殖过程海量数据自动处理系统[J]. 农业机械学报,2012,43(8):193 – 197.
Lao Fengdan, Teng Guanghui, Yu Ligen, et al. Automatic processing system of massive data in laying hens raise [J]. Transactions of the Chinese Society for Agricultural Machinery, 2012, 43(8):193 – 197. (in Chinese)