

基于分布式流式计算的生猪养殖视频监测分析系统

邹远炳 孙龙清 李 玥 李亿杨
(中国农业大学信息与电气工程学院, 北京 100083)

摘要: 基于分布式流式计算框架,提出了节点资源调度器算法,构建了可插拔的分布式流式实时计算模型,研究开发了生猪养殖视频监测分析系统。系统实现了规模化生猪养殖视频流数据采集、任务调度、实时计算、可插拔式扩展和结果展示的功能。在由1个主节点和3个从节点构成的集群下,采用改进混合高斯模型的背景更新方式,实现集群下多摄像头多目标的实时检测。平均处理速度比传统混合高斯模型提高了29.00%,平均检测率为79.00%,平均误检率比传统混合高斯模型降低了70.96%。测试结果表明,可插拔分布式流式实时计算模型具有较好的可扩展性,视频流处理算法速度和实时性得到了提升,具有较高的检测率和较低的误检率。

关键词: 分布式流式计算; 生猪养殖; 视频分析; 实时监测

中图分类号: TP391 **文献标识码:** A **文章编号:** 1000-1298(2017)S0-0365-09

Video Monitoring and Analysis System for Pig Breeding Based on Distributed Flow Computing

ZOU Yuanbing SUN Longqing LI Yue LI Yiyang
(College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China)

Abstract: With the rapid development of computer technology, it is possible to process multi-type and mass data in real-time. In order to overcome the problems of distributed streaming computing in processing a large number of pig video streaming data when the delay was high and with poor scalability problems, a node resource scheduler algorithm was proposed, and a pluggable distributed real-time flow computation model was constructed. A system of video monitoring and analysis for pig breeding based on distributed flow calculation was developed. The system implemented the functions of pig video stream data acquisition, task scheduling, real-time calculation, pluggable expansion and result display. The test cluster consisted of a master node and three slave nodes. Under the cluster, the background refreshing method of improved hybrid Gaussian model was adopted to realize the multi-camera and multi-target detection in real-time. The average processing rate was 29.00% higher than the traditional mixed Gaussian model, the average detection rate was 79.00%, and the average false detection rate was 70.96% lower than that of the traditional mixed Gaussian model. The results showed that the pluggable distributed streaming real-time computing model had good scalability and low latency. The improved hybrid Gaussian model algorithm had high detection rate and low false detection rate.

Key words: distributed flow computing; pig breeding; video analysis; real-time monitoring

引言

中国是世界上生猪养殖和猪肉消费大国,随着生猪养殖从传统养殖向现代化、规模化养殖的转变^[1-4],农业物联网技术以及生猪养殖智能视频监控技术得到了广泛应用^[5-7]。据不完全统计,在规模化生猪养殖企业中,为了能够对养殖场进行人工观察,养殖企业都花费大量资金,在圈舍内安装了视频数据采集装备。针对规模巨大的视频数据流,目前仅用于日常人工观察,缺乏对生猪个体实时智能

收稿日期: 2017-07-14 修回日期: 2017-11-21
基金项目: 国家高技术研究发展计划(863计划)项目(2013AA102306)
作者简介: 邹远炳(1989—),男,硕士生,主要从事计算机图形图像处理研究,E-mail: zourose123@126.com
通信作者: 孙龙清(1964—),男,副教授,主要从事农业信息化研究,E-mail: sunlq@cau.edu.cn

处理和分析,失去了为养殖企业提供前期预警和逆向控制养殖设备的作用。为了能够充分发挥视频数据对生猪个体实时监测的作用,研究开发一套基于分布式流式计算的实时处理多摄像头多目标生猪监测分析系统,对规模化养殖场生猪实施个性化有效监控管理、及时发现存在的问题、改善养殖环境以及降低病害发生率等具有重要的意义^[8]。

分布式计算是将需要巨大计算能力的计算问题分解成许多小问题,进而将这些小问题分配给多台计算机进行处理,最后把这些计算结果综合起来得到最终结果^[9]。利用分布式平台提供的资源共享和协同计算的能力,可以很好地解决大规模数据的处理分析问题^[10]。传统分布式计算采用先存储后计算的批量处理机制,对大规模视频流数据处理延时高、实时性差。流式计算采用无需存储的直接处理模式,对采集的数据流直接在内存中对数据进行计算,数据处理具有实时性好、延时低^[11]等特点。近年来,国内外对基于分布式流式计算在处理大规模数据方面进行了大量的研究^[12-15]。针对流式计算数据处理扩展性问题,LI等^[16]设计了一种基于源和汇的有向图实时流计算模型实现系统的高可扩展性。CIPRIANI等^[17]通过编排数据流图为处理流水线建模,具有较好的可扩展性。URBANI等^[18]采用分布式多线程架构,利用动态分配的通用运算符来代替传统的 Map / Reduce 原语,设计了中间件 AJIRA,具有较好的可扩展性。针对流式计算处理任务时消息延时较高的问题,韩杰等^[19]通过节点性能监控并利用贪心置换策略动态调节 Worker Node 负载,改进 Storm 的默认调度器,降低消息处理的延时。VERMA等^[20]提出了基于异构最早完成时间(HEFT)预算和截止限制的启发式调度 workflow,实现执行时间和执行成本之间的平衡。SUN等^[21]提出了一种称为 Re-Stream 的资源调度和优化框架,利用符合架构要求的节能启发式和关键路径的分配任务调度机制,从而实现高能效和低响应时间,该方法采用的节能启发式和关键路径调度机制与任务关

联度较大,扩展性较差。ANIELLO等^[22]通过分析拓扑结构并调整其部署来进行脱机工作,具有更好的实时性。分布式流式计算技术在多个领域已有较为广泛的应用^[23],在农业领域,陈红茜等^[24]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,用于实现蛋鸡养殖生产过程参数实时监测与预警。在交通领域,CHEN等^[25]提出一种使计算任务直接在网络边缘完成的动态视频流处理方案,以满足智慧城市监控应用的需求。

为了克服分布式流式计算在处理大规模生猪视频流数据时的延时较高、扩展性差的问题,本文基于分布式流式计算框架 Storm,提出基于节点资源调度器算法,构建可插拔的分布式流式实时计算模型,研发生猪养殖实时监控系统。实现集群中各计算节点负载均衡、多种编码格式的视频数据无缝解析、生猪视频流数据的高效实时处理。

1 系统设计

1.1 系统需求

1.1.1 用户需求

分布式流式计算生猪养殖视频监测分析系统的主要用户包括:企业生产主管、养殖场场长、系统管理员 3 类,系统用例图如图 1 所示。企业生产主管通过系统实时监测本企业每个养殖场的生产现状和生产环境,以及查看视频监测分析结果,将监测分析结果作为其决策的依据,查看用户信息;系统管理员对用户进行管理,分配用户权限,根据需求配置数据源,分配计算节点,监控集群运行情况;养殖场场长通过系统实时监测和记录所属养殖场生猪采食、饮水、排泄、趴卧、站立等行为的时间以及次数等生猪活动信息,查看对生猪采食、饮水、排泄、趴卧、站立等行为的统计和分类结果、查看生猪运动曲线,管理生猪信息、监控数据显示等^[26]。

1.1.2 功能需求

基于分布式流式计算的生猪养殖视频监测分析系统主要功能有:①视频流数据采集,获取、解析多

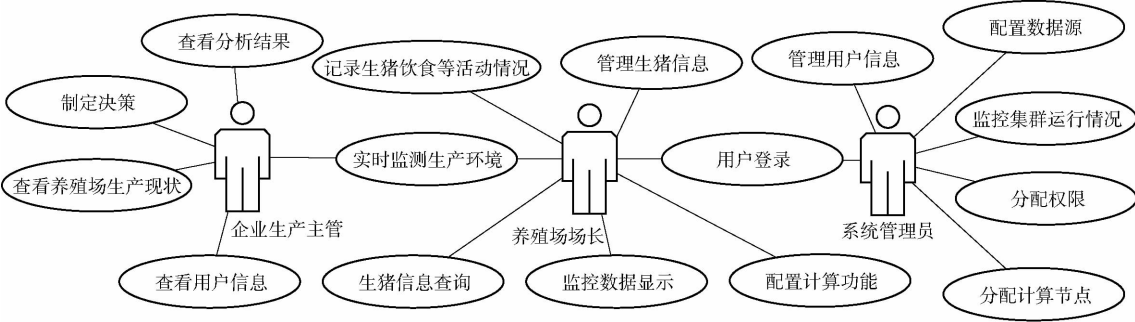


图 1 系统用例图

Fig. 1 Use case diagram of the system

个不同编码格式的视频数据源,对解析后的视频进行分组。②任务调度,定时获取节点的资源使用情况,基于节点资源调度器 (Based on node resources scheduler, BNRS) 算法进行任务调度,实现集群的负载均衡,最大限度减少数据处理延时。③实时计算,根据用户配置的功能映射信息,选择相应的处理算法,对视频流进行实时计算。如对生猪行为检测、跟踪、记录和分析等。④可插拔式扩展,用户根据需求配置功能模块与标识映射,动态选择和创建功能类,实现系统处理功能的扩展。⑤计算结果存储,将原始数据和计算结果数据存储于分布式文件系统或 MySQL 数据库中,以便后续的使用和查看。⑥结果展示,采用 B/S 架构,实时展示生猪活动信息以及计算分析结果。

1.1.3 系统性能需求

本系统基于分布式流式计算框架 Storm 进行实时处理,满足系统对实时性的需求。针对后续计算功能需要扩展的需求,建立了可插拔的分布式流式实时计算模型,使系统具有良好可扩展性。

1.2 系统设计

流式计算的生猪养殖实时监测分析平台(以下称平台)采用分布式面向服务的体系结构。系统结构如图 2 所示。

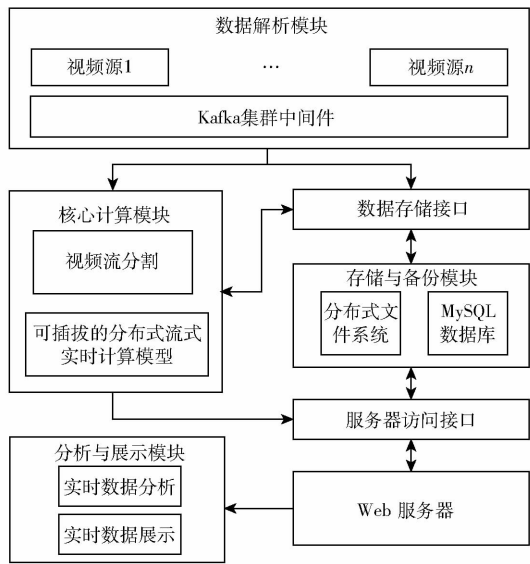


图 2 系统结构图

Fig. 2 Architecture diagram of the system

平台分为数据解析模块、核心计算模块、存储与备份模块以及分析与展示模块 4 部分:

(1) 数据解析模块

该模块实现生猪视频流数据实时解析、传输的功能。当视频数据采集装备所采集的视频流数据实时传输到分布式计算机时,数据解析模块对不同编码格式的视频流数据进行解析,即采用统一格式对

视频数据流进行重新包装,形成新的视频流数据;将解析后的视频流数据传送到核心计算模块。由于数据在传输过程中存在丢包的情况,为了容错,数据解析模块与核心计算模块的通信选用容错性能较好的消息中间件 Kafka。

(2) 核心计算模块

该模块基于分布式的流式计算框架 Storm^[27-30],包括输入流组件、基于 Ganglia 监控服务的 Topology 任务调度器、基于 Zookeeper^[31] 协调服务的控制节点集群协调器、持久化输出组件。通过输入流组件获取中间件 Kafka 传输过来的视频流,对视频流进行分割。可插拔的分布式流式实时计算模型对分割后的视频流进行统一结构封装分组,分组视频发送给各计算节点,在不同层上的计算节点可以插入相应的功能处理模块,对发送过来的分组视频进行处理,再发送到后续处理节点进行处理,实现目标生猪视频的实时监控。管理节点 Supervisor 通过实时监测 Ganglia 监控服务节点的资源信息来调度 Topology 任务,实现集群的负载均衡。核心计算模块对数据处理的结果可以用于分析与展示模块进行实时分析展示,也可直接用于存储与备份模块进行存储。

(3) 存储与备份模块

该模块实现实时存储流数据,进行多备份以保证不丢失数据。消息中间件 Kafka 将采集端异步发送过来的数据持久化后存储到本地数据库或分布式文件系统中。核心计算模块对视频流数据处理后,通过数据存储接口,将数据分别存储至 MySQL 数据库和分布式文件系统中。

(4) 分析与展示模块

该模块对核心计算模块产生的数据和历史数据进行分析,将处理分析结果展示给用户或工作人员。分析的内容主要有:对生猪采食、饮水、排泄、趴卧、站立等行为的时间以及次数等数据进行统计和分类,根据生猪采食、饮水等行为的时间和次数统计结果分析生猪的健康状况。展示的主要内容包括:生猪监控视频实时展示、生猪采食等行为的时间和次数的统计结果展示、生猪运动曲线展示等。展示方式:用户在客户端通过交互的方式发送展示请求,服务器获取请求并实时向客户端发送处理结果数据。服务器通过 Web Service 接口向各种终端提供实时流数据以及 XML 通用数据来实现跨平台展示。

2 关键算法及实现

本系统基于开源流计算框架 Storm0.9.4、开源图像处理软件 OpenCV2.4.9 等框架进行扩展与定

制。

2.1 基于节点资源调度器 (BNRS) 算法与实现

Storm 默认的调度器 DefaultScheduler 采用了轮询调度 (Round robin scheduling) 机制,该机制将系统中可用资源均匀地分配到配置好的任务队列中,并将所有任务尽量等量地分配到每台计算节点 (Worker)。这种简单的分配机制在实际应用中可能导致集群中资源分配不均匀,降低集群整体性能。

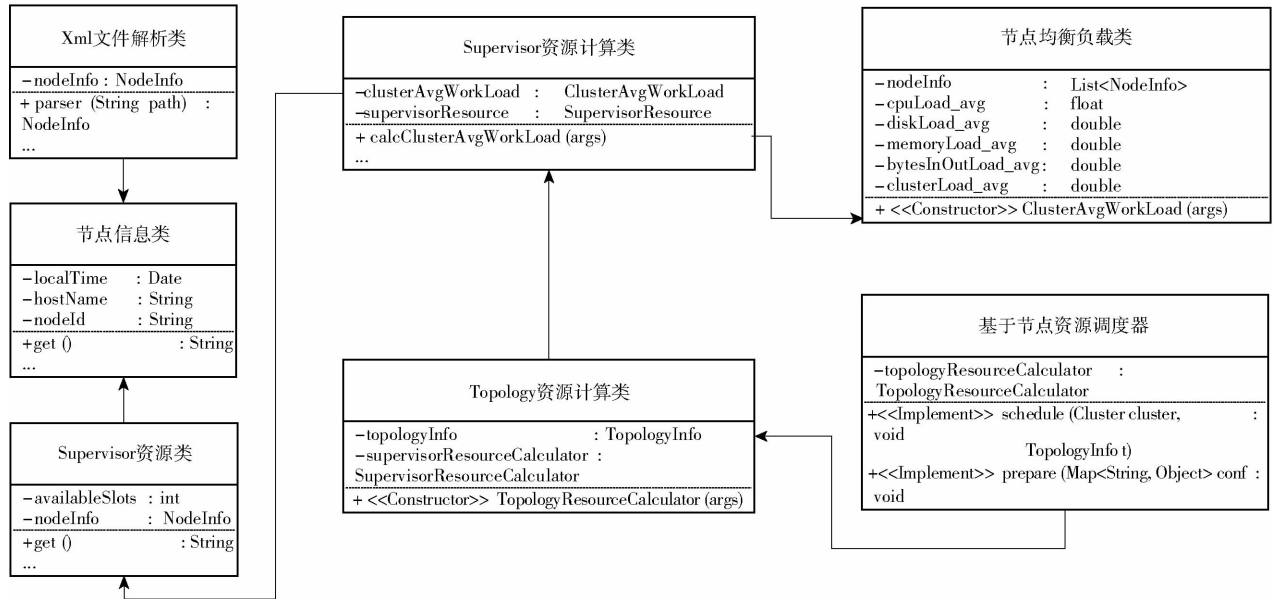


图 3 调度器的类图
Fig.3 Class diagram of scheduler

定义 1: t 时刻节点 i 的工作负载为 $R_{i,t}$, 计算公式为

$$R_{i,t} = R_{i,t}^c + R_{i,t}^n + R_{i,t}^m + R_{i,t}^d \quad (i = 1, 2, \dots) \quad (1)$$

式中 $R_{i,t}^c$ —— t 时刻 i 节点 CPU 占用率

$R_{i,t}^n$ —— t 时刻 i 节点网络吞吐量

$R_{i,t}^m$ —— t 时刻 i 节点内存占用率

$R_{i,t}^d$ —— t 时刻 i 节点磁盘占用率

定义 2: t 时刻集群平均负载情况 $R_{avg,t}$ 定义为

$$R_{avg,t} = R_{avg,t}^c + R_{avg,t}^n + R_{avg,t}^m + R_{avg,t}^d \quad (2)$$

式中 $R_{avg,t}^c$ —— t 时刻集群 CPU 平均占用率

$R_{avg,t}^n$ —— t 时刻集群平均网络吞吐量

$R_{avg,t}^m$ —— t 时刻集群内存平均占用率

$R_{avg,t}^d$ —— t 时刻集群磁盘平均占用率

定义 3: t 时刻 i 节点的处理能力 $P_{i,t}$ 的计算公式为

$$P_{i,t} = \frac{R_{i,t}^c}{R_{avg,t}^c} + \frac{R_{i,t}^n}{R_{avg,t}^n} + \frac{R_{i,t}^m}{R_{avg,t}^m} + \frac{R_{i,t}^d}{R_{avg,t}^d} \quad (3)$$

定义 4: t 时刻各节点负载能力均值为 L_{avg}^t , 则

$$L_{avg}^t = \frac{\sum R_{i,t}}{\sum P_{i,t}} \quad (4)$$

考虑到在实际生产过程中每个计算节点的计算能力、网络吞吐量、内存等处理性能存在差异,需要基于节点粒度来均衡分配任务^[32]。本系统对各个摄像头采集的数据进行实时、公平的处理,减少计算资源的浪费。根据负载均衡度量函数^[33],提出了基于节点资源的改进负载均衡度量函数和 Sotrm 提供的插拔式调度策略 Ischeduler 接口,实现基于节点资源调度器的调度功能。BNRS 类图如图 3 所示。

定义 5: 集群各节点负载度量函数 F_i^t , 当 F_i^t 小于 5%^[25] 时, 认为 t 时刻节点已达到负载平衡状态, 即

$$F_i^t = \left| 1 - \frac{R_{i,t}}{L_{avg}^t P_{i,t}} \right| < 5\% \quad (5)$$

采用 F_i^t 对计算节点是否均衡进行判断,若满足式(5),则认为 t 时刻计算节点负载均衡,否则负载不均衡。本文采用开源集群监视软件 Ganglia,在每个计算节点上安装 Gmetad 服务。通过该服务来监控计算节点的各项性能指标,从端口 8649 以 xml 格式发布节点性能指标信息。通过访问该端口并解析相应的 XML 文件即可获取节点相应的 CPU、内存、网络、磁盘等使用信息。通过利用 BNRS 对这些信息的处理,来进行任务的分配。

BNRS 处理步骤如下:

(1) 调用 GangliaXmlParser 解析从 8649 端口产生的 XML 格式数据,得到节点资源信息,并将节点资源信息存储在 NodeInfo 对象中。

(2) 调用 SupervisorResourceCalculator 获取 NodeInfo 对象中每个节点资源使用信息,计算每个节点的 F_i^t 。

K ——高斯分布的个数(一般取 $3 \sim 5$), 每个像素的 K 个高斯分布按 $\rho_{i,t} = \omega_{i,t} / \sigma_{i,t}$ 从高到低排列, $\sigma_{i,t}$ 为第 i 个高斯分布在 t 时刻的标准差

$\omega_{i,t}$ —— t 时刻高斯混合模型中第 i 个高斯分布的权重
 $\mu_{i,t}$ —— t 时刻高斯混合模型中第 i 个高斯分布的均值
 $\Sigma_{i,t}$ —— t 时刻高斯混合模型中第 i 个高斯分布的协方差矩阵
 η ——高斯分布的概率密度
高斯分布的概率密度函数为

$$\eta(\boldsymbol{X}_t, \boldsymbol{\mu}_{i,t}, \boldsymbol{\Sigma}_{i,t}) = \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}_{i,t}|^{-1}} e^{-\frac{1}{2}(\boldsymbol{X}_t - \boldsymbol{\mu}_{i,t})^T \boldsymbol{\Sigma}_{i,t}^{-1} (\boldsymbol{X}_t - \boldsymbol{\mu}_{i,t})} \quad (7)$$

式中 n —— $\boldsymbol{X}_t$ 的维数
 $n=1$ 时 $\boldsymbol{X}_t$ 代表像素点的灰度, $n=3$ 时 $\boldsymbol{X}_t$ 代表像素点的 RGB 值。

t 时刻像素 $\boldsymbol{X}_t$ 与按 $\rho_{i,t}$ 从高到低排列的 K 个高斯模型分布进行匹配, 匹配公式为

$$|\boldsymbol{X}_t - \boldsymbol{\mu}_{i,t-1}| \leq D\sigma_{i,t-1} \quad (8)$$

式中 D ——参数(一般取 2.5)
 $\mu_{i,t-1}$ ——第 i 个高斯分布在 $t-1$ 时刻的均值
 $\sigma_{i,t-1}$ ——第 i 个高斯分布在 $t-1$ 时刻的标准差

如满足式(8)则表示当前像素点 $\boldsymbol{X}_t$ 与该高斯分布模型匹配, 否则认为不匹配。若匹配, 则将匹配的高斯分布的权值按照

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha M_{i,t} \quad (9)$$

$$M_{i,t} = \begin{cases} 1 & (|\boldsymbol{X}_t - \boldsymbol{\mu}_{i,t}| \leq D\sigma_{i,t}) \\ 0 & (|\boldsymbol{X}_t - \boldsymbol{\mu}_{i,t}| > D\sigma_{i,t}) \end{cases} \quad (10)$$

式中 α ——学习率, 一般取 0.005
进行更新。

当像素匹配该高斯分布时, 置 $M_{i,t} = 1$, 否则置 $M_{i,t} = 0$ 。

匹配模式的均值和标准差更新公式为

$$\mu_{i,t} = (1 - \beta)\mu_{i,t-1} + \beta \boldsymbol{X}_t \quad (11)$$
$$\sigma_{i,t}^2 = (1 - \beta)\sigma_{i,t-1}^2 + \beta (\boldsymbol{X}_t - \mu_{i,t-1})^T (\boldsymbol{X}_t - \mu_{i,t-1}) \quad (12)$$
$$\beta = \alpha \eta(\boldsymbol{X}_t, \mu_{i,t}, \Sigma_{i,t}) \quad (13)$$

式中 β ——期望与方差的学习速率
若不匹配时, 则权重按式(9)更新, 均值、方差保持不变。权重更新只需令 $M_{i,t}$ 为 0; 对于其他高斯模型, 其均值和方差取值不变。对各高斯分布按 $\rho_{i,t}$ 的值进行降序排列, 权重大且标准差小的高斯分布会排在前面, 从排列在最前的第一个模型开始, 取 B 个高斯分布作为混合高斯分布的背景模型, B 按式(14)进行选取:

$$B = \arg \min_b \left(\sum_{i=1}^b \omega_{i,t} > T \right) \quad (14)$$

式中 T ——权重阈值
若当前像素点与背景模型中的任何一个模型匹配, 则该像素点为背景点, 否则为前景点。

2.3.2 改进背景模型更新方式

由于传统混合高斯模型连续使用新的像素观测值更新背景模型的权值和参数, 不仅计算量大, 在集群下很难实现实时检测和提取生猪目标, 由于生猪养殖场养殖环境、背景以及生猪运动状态变化复杂, 容易导致检测结果中噪声较多、生猪目标检测率低的问题。本文在混合高斯模型中引入背景学习参数和约束条件, 来优化传统混合高斯模型背景更新方式。定义为

$$V_{x,t}^a = (1 - \alpha_{x,t})M_{i,t-1} V_{x,t-1} + \alpha_{x,t} M_{i,t-1} \boldsymbol{X}_t \quad (15)$$

$$|\boldsymbol{X}_t - V_{x,t}^a| \leq V_{const} \quad (16)$$

式中 $V_{x,t-1}$ —— $t-1$ 时刻的背景像素
 $V_{x,t}^a$ —— t 时刻关于像素点 $\boldsymbol{X}_t$ 的背景学习参数, 用来记录 t 时刻像素点学习结果
 $\alpha_{x,t}$ —— t 时刻像素点 $\boldsymbol{X}_t$ 的学习率
 V_{const} ——用户自定义阈值

当 $t-1$ 时刻样本像素观测值 $\boldsymbol{X}_{t-1}$ 为前景点时, 则背景参数为 $V_{x,t}^a = (1 - \alpha_{x,t}) V_{x,t-1} + \alpha_{x,t} \boldsymbol{X}_t$, 当 $t-1$ 时刻样本像素观测值 $\boldsymbol{X}_{t-1}$ 为背景点时, 则背景参数为 $V_{x,t}^a = V_{x,t-1}$, 若当前像素值 $\boldsymbol{X}_t$ 与背景参数 $V_{x,t}^a$ 满足式(16), 则表示当前像素值与背景相近, 用混合高斯模型对其建模, 进行进一步判断; 若当前像素值与背景参数的关系不满足式(16), 则表示当前像素值与背景相差很大, 可直接判断该像素为前景点, 无须用混合高斯建模。

3 实验结果分析

本实验视频采集于北京市南口镇中国农业科学院饲料研究所试验基地生猪养殖场。预先录制了 6 段时长为 1 h 的生猪运动视频, 作为测试数据。集群由 1 个主节点(Master 节点)和 3 个从节点(Slave 节点)组成。Master 节点所用的计算机配置为 Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz, 4 GB 内存, Ubuntu16.04 操作系统, 用来运行 Nimbus 和 ZooKeeper 程序, 3 个 Slave 节点的配置为 CPU i5-4570, 运行 Supervisor 程序。

3.1 改进调度器效果分析

主要测试内容为: 系统在使用默认调度器和改进调度器下系统的平均延迟; 随着输入量的增长, 2 种调度器下集群的负载均衡情况。

(1)测试时给输入的数据对象添加时间戳, 在

系统最后一级处理节点解析初始时间并计算此数据流的延时,多次统计后计算平均值。测试数据源为6段时长为1 h的生猪运动视频,每个从节点同时处理两段视频。从图5可以看出,在任务启动阶段,由于系统负载较高,导致元组处理的延时较大。在相同时间间隔下使用默认调度器进行数据调度产生的消息延时均高于本文调度器。由于在250~400 s时,生猪运动较为剧烈,两种调度器下计算节点花费较多的时间进行生猪检测。

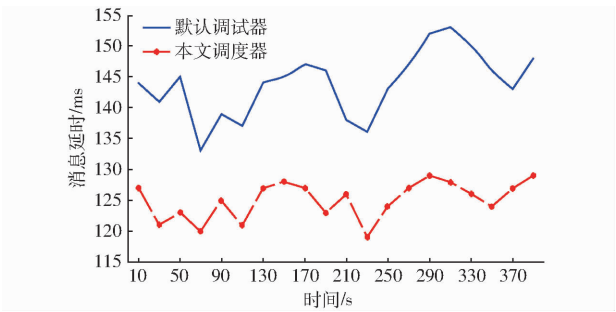


图5 两种调度器的消息延时比较

Fig.5 Comparison of message delay between two schedulers

(2)测试输入数据:使用默认调度器、改进的BNRS调度器进行视频数据调度。图6表明当输入1个视频和2个视频时,系统在2种调度器下工作的执行时间相差不大。随着输入数据源的数量增加,改进的BNRS调度器优势逐渐显现。当输入数

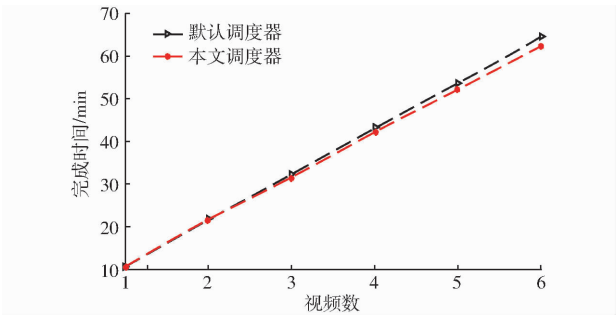


图6 不同视频数下的系统整体延时比较

Fig.6 Comparison of overall system delay under different video frequencies

据量不断增大,改进的BNRS调度器可以将部分任务调度到存在空闲资源且负载任务小的节点上执行,缩短了集群的处理时间,整个集群达到了负载均衡。

3.2 算法处理速度分析

为合理、客观地分析本文算法和传统混合高斯模型在集群下的处理速度,本文选取了6段不同的监控视频,其中3段监控视频采用传统混合高斯模型处理,另外3段视频采用本文算法进行处理,记录每段视频帧的帧数、开始处理时间和结束处理时间,并计算得到2种算法对每段视频处理速度,如表1所示。实验效果如图7所示,图7a~7c为传统混合高斯模型的检测效果,图7d~7f为本文算法的检测效果。

表1 各方法处理速度

Tab.1 Processing speed of different methods

序号	帧数	日期	开始时间	结束时间	背景建模方法	
					GMM 处理速度/(fps·s ⁻¹)	本文方法处理速度/(fps·s ⁻¹)
1	89997	2017-05-25	08:00:15	09:45:30	14.25	17.53
2	89042	2017-05-25	08:00:15	09:49:41	13.86	18.16
3	89996	2017-05-25	08:00:15	09:49:05	13.78	17.84
4	90024	2017-05-25	08:00:15	09:24:07	12.93	17.79
5	89995	2017-05-25	08:00:15	09:25:22	14.35	18.84
6	89993	2017-05-25	08:00:15	09:23:01	13.98	17.12
平均速度					13.86	17.88

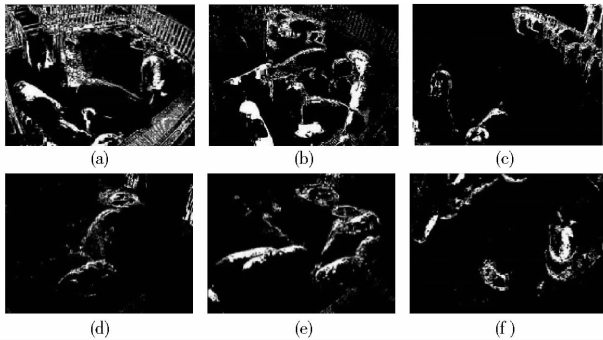


图7 传统混合高斯背景建模与本文算法检测效果

Fig.7 Detection results between GMM and our method

从表1可以看出,本文算法在集群下的处理速度与传统混合高斯处理速度相比有明显提升,平均处理速度分别为17.88 fps/s和13.86 fps/s,平均处理速度提高了29.00%。在运动目标处于不同运动状态下,当前像素点可直接根据式(16)判断是否为前景像素点而无需参加背景建模,提升了算法处理速度。

3.3 算法处理效果分析

由图7a~7c可看出,视频检测效果中噪声较多,检测效果较差,有较多的空洞现象,图7d~7f中,生猪目标检测效果的噪声较少,含有部分空洞。从图7中可以看出,本文算法构建的背景检测模型

在集群下检测的目标误检和漏检方面优于传统混合高斯模型。

为对检测效果精确的表示,采用人工手动提取视频前景对象值作为检测率和误检率评价的基准,与传统混合高斯模型进行检测率和误检率对比^[37]。

$$T_{ipr} = \frac{t_p}{t_p + f_n}$$

(17)

$$T_{fpr} = \frac{f_p}{t_p + f_p}$$

(18)

式中 T_{ipr} ——运动目标区域内像素的检测率
 t_p ——运动目标区域内正确检测到运动目标的像素数
 f_n ——运动目标区域内漏检的像素数
 T_{fpr} ——运动目标区域外错检为目标的误检率
 f_p ——错误检测为运动目标的像素数

为得到更为精确的检测效果,本文选取 6 组视频分别在近 10 000 帧时对传统混合高斯模型(GMM)和本文算法的检测效果进行分析,相关实验数据见表 2。从表 2 可以看出,本文算法和混合高斯模型背景建模平均误检率分别为 10.24% 和 35.26%,平均误检率降低了 70.96%;平均检测率分别为 79.00% 和 72.49%,本文算法的平均检测率高于混合高斯算法 8.98%。这主要由于在运动目标长时间不动或缓慢运动状态时,采用改进的背景更新方式,克服了前景消融问题,使得检测结果更加符合真实场景。

表 2 传统 GMM 算法与本文算法性能比较

Tab.2 Comparison of the traditional GMM algorithm and the algorithm %

序号	检测率		误检率	
	混合高斯算法	本文算法	混合高斯算法	本文算法
1	68.03	79.53	36.01	9.62
2	70.60	78.48	35.18	9.92
3	68.32	84.15	36.32	11.38
4	83.73	76.95	34.47	10.85
5	69.48	79.23	35.63	8.93
6	74.75	75.64	33.96	10.71
平均值	72.49	79.00	35.26	10.24

4 结 论

(1) 本文提出节点资源的改进负载均衡度量函数,结合 Sotrm 提供的插拔式调度策略 Ischeduler 接口,实现了基于节点资源调度器 BNRS,通过对现场流式数据的实时处理和分析,满足了系统的时效性。
(2) 系统设计并实现了可插拔的分布式流式实时计算模型,充分考虑了未来扩展的可能性,实现了高内聚、低耦合,有利于系统的维护和升级。
(3) 通过改进混合高斯模型的背景更新方式,实现了在集群下多摄像头多生猪目标的检测,处理速度和实时性得到了提升,具有较高的检测率和较低的误检率,平均处理速度比传统混合高斯模型提高了 29.00%,平均检测率为 79.00%,平均误检率为 10.24%,平均误检率比混合高斯模型降低了 70.96%。

参 考 文 献

1 李惠敏,连京华,孙凯,等. 家禽环境自动化控制技术研究进展[J]. 中国家禽, 2013, 35(14):41-44.
2 段玉瑶. 生猪目标跟踪及行为监测方法研究[D]. 北京:中国农业大学,2015.
3 肖德琴,冯爱晶,杨秋妹,等. 基于视频追踪的猪只运动快速检测方法[J/OL]. 农业机械学报,2016,47(10):351-357, 331. http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx?file_no=20161045&flag=1. DOI:10.6041/j.issn.1000-1298.2016.10.045.
XIAO Deqin, FENG Aijing, YANG Qiumei, et al. Fast motion detection for pigs based on video tracking[J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2016, 47(10):351-357, 331. (in Chinese)
4 孙鑫鑫. 基于粒子滤波的生猪采食行为跟踪技术研究[D]. 北京:中国农业大学,2014.
5 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)
6 何勇,聂鹏程,刘飞. 农业物联网与传感仪器研究进展[J/OL]. 农业机械学报, 2013, 44(10):216-226. http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx?file_no=20131035&flag=1. DOI:10.6041/j.issn.1000-1298.2013.10.035.
HE Yong, NIE Pengcheng, LIU Fei. Advancement and trend of internet of things in agriculture and sensing instrument[J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2013, 44(10):216-226. (in Chinese)
7 崔敏. 农业物联网研究与应用现状及发展对策[J]. 计算机与网络, 2016, 45(15):222-230.
8 秦琳琳,陆林箭,石春,等. 基于物联网的温室智能监控系统设计[J/OL]. 农业机械学报, 2015, 46(3):261-267. http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx?file_no=20150338&flag=1. DOI:10.6041/j.issn.1000-1298.2015.03.038.
QIN Linlin, LU Linjian, SHI Chun, et al. Implementation of IOT-based greenhouse intelligent monitoring system[J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2015, 46(3):261-267. (in Chinese)
9 胡建平,胡凯. 分布式计算系统导论:原理与组成[M]. 北京:清华大学出版社, 2014.
10 陈权. 基于分布式集群的多摄像头的目标检测和跟踪的并行算法[D]. 南京:南京理工大学, 2015.
11 ZHANG Q, SONG Y, ROURAY R R, et al. Adaptive block and batch sizing for batched stream processing system[C]// IEEE International Conference on Autonomic Computing. IEEE, 2016:35-44.

- 12 SUN D, TANG H, GAO S, et al. A strategy to improve accuracy of multi-dimensional feature forecasting in big data stream computing environments[C]// International Conference on Web Information Systems Engineering. Springer, Cham, 2016:405–413.
- 13 TA V D, LIU C M, NKABINDE G W. Big data stream computing in healthcare real-time analytics[C]// IEEE International Conference on Cloud Computing and Big Data Analysis. IEEE, 2016:37–42.
- 14 李招昕. 基于流式计算的大规模监控视频分析关键技术研究[D]. 上海:上海大学, 2015.
- 15 BACCARELLI E, CORDESCI N, MEI A, et al. Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study[J]. IEEE Network, 2016, 30(2):54–61.
- 16 LI Z, ZHANG C, XU K. SpeedStream: a real-time stream data processing platform in the cloud[C]// Computing and Communications Conference. IEEE, 2015:1–10.
- 17 CIPRIANI N, EISSELE M, BRODT A, et al. NexusDS: a flexible and extensible middleware for distributed stream processing [C]// International Database Engineering & Applications Symposium. ACM, 2009:152–161.
- 18 URBANI J, MARGARA A, JACOBS C, et al. AJIRA: a lightweight distributed middleware for map reduce and stream processing [C]// IEEE, International Conference on Distributed Computing Systems. IEEE Computer Society, 2014:545–554.
- 19 韩杰, 陈耀武. 基于 Storm 平台的实时视频分析系统[J]. 计算机工程, 2015, 41(12):26–29.
HAN J, CHEN Y W. Real-time video analysis system based on Storm platform[J]. Computer Engineering, 2015, 41(12):26–29. (in Chinese)
- 20 VERMA A, KAUSHAL S. Cost-time efficient scheduling plan for executing workflows in the cloud [J]. Journal of Grid Computing, 2015, 13(4):1–12.
- 21 SUN D, ZHANG G, YANG S, et al. Re-Stream: real-time and energy-efficient resource scheduling in big data stream computing environments[J]. Information Sciences, 2015, 319:92–112.
- 22 ANIELLO L, BALDONI R, QUERZONI L. Adaptive online scheduling in storm [C]// ACM International Conference on Distributed Event-Based Systems. ACM, 2013:207–218.
- 23 孙大为, 张广艳, 郑纬民. 大数据流式计算:关键技术及系统实例[J]. 软件学报, 2014, 25(4):839–862.
SUN D W, ZHANG G Y, ZHENG W M. Big data stream computing: technologies and instances[J]. Journal of Software, 2014, 25(4):839–862. (in Chinese)
- 24 陈红茜, 滕光辉, 邱小彬, 等. 基于分布式流式计算的蛋鸡养殖实时监测与预警系统[J/OL]. 农业机械学报, 2016, 47(1):252–259. http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx?file_no=20160134&flag=1. DOI:10.6041/j.issn.1000-1298.2016.01.034.
CHEN H Q, TENG G H, QIU X B, et al. Real-time monitoring and early warning system based on stream computing for laying hens raise[J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2016, 47(1):252–259. (in Chinese)
- 25 CHEN N, CHEN Y, YOU Y, et al. Dynamic urban surveillance video stream processing using fog computing[C]// IEEE Second International Conference on Multimedia Big Data. IEEE, 2016:105–112.
- 26 李亿杨, 孙龙清, 孙鑫鑫. 基于多特征融合的粒子滤波生猪采食行为跟踪[J]. 农业工程学报, 2017, 33(增刊1):246–252.
LI Yiyang, SUN Longqing, SUN Xinxin. Automatic tracking of pig feeding behavior based on particle filter with multi-feature fusion[J]. Transactions of the CSAE, 2017, 33(Supp.1):246–252. (in Chinese)
- 27 KE M A, LINGJUAN L I. Distributed real time stream data clustering algorithm and its implementation based on Storm[J]. Journal of Nanjing University of Posts & Telecommunications, 2016, 36:104–110.
- 28 SOLAIMANI M, KHAN L, THURASINGHAM B. Real-time anomaly detection over VMware performance data using storm[C]// IEEE, International Conference on Information Reuse and Integration. IEEE, 2015:458–465.
- 29 MASKE M M, PRASAD P. A real time processing and streaming of wireless network data using Storm[C]// International Conference on Computation of Power, Energy Information and Communication. IEEE, 2015.
- 30 LEIBIUSKY J, EISBRUCH G, SIMONASSI D. Getting Started with Storm[M]. O'Reilly Media, Inc., 2012.
- 31 HUNT P, KONAR M, JUNQUEIRA F P, et al. ZooKeeper: wait-free coordination for internet-scale systems[C]// Proceedings of the 2010 USENIX conference on USENIX annual technical conference. Berkeley, CA, USA, 2010:653–710.
- 32 黄长营. 分布式数据流处理系统的负载均衡技术研究[实现][D]. 成都:电子科技大学, 2012.
- 33 刘盼红. 大数据环境下 Hadoop 作业调度算法的研究[D]. 邯郸:河北工程大学, 2015.
- 34 GRIMSON W E L, STAUFFER C, ROMANO R, et al. Using adaptive tracking to classify and monitor activities in a site[C]// Computer Vision and Pattern Recognition, 1998 IEEE Computer Society Conference on. IEEE, 1998:22–29.
- 35 田杰, 韩冬, 胡秋霞, 等. 基于 PCA 和高斯混合模型的小麦病害彩色图像分割[J/OL]. 农业机械学报, 2014, 45(7):267–271. http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx?file_no=20140741&flag=1. DOI:10.6041/j.issn.1000-1298.2014.07.041.
TIAN Jie, HAN Dong, HU Qiuxia, et al. Segmentation of wheat rust lesion image using PCA and gaussian mix model[J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2014, 45(7):267–271. (in Chinese)
- 36 赵瑶池, 胡祝华, 白勇, 等. 基于纹理差异度引导的 DRLSE 病虫害图像精准分割方法[J/OL]. 农业机械学报, 2015, 46(2):14–19. http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx?file_no=20150203&flag=1. DOI:10.6041/j.issn.1000-1298.2015.02.003.
ZHAO Yaochi, HU Zhuhua, BAI Yong, et al. An accurate segmentation approach for disease and pest based on texture difference guided DRLSE[J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2015, 46(2):14–19. (in Chinese)
- 37 赵亚欣, 蔡华杰, 赵怀勋, 等. 基于改进高斯混合模型的自适应前景提取[J]. 计算机应用与软件, 2016, 33(11):161–163.