

Design of Data Management System for Electric Hoist Energy Efficiency Measurement Based on HBase

Jun Mei¹, Songlei Wang², Guobao Zhang¹

¹Automation School of Southeast University, Nanjing Jiangsu

²Jiangsu Province Special Equipment Safety Supervision Inspection Institute, Branch of WuXi, Wuxi Jiangsu
Email: 220161393@seu.edu.cn

Received: Dec. 31st, 2018; accepted: Jan. 15th, 2019; published: Jan. 22nd, 2019

Abstract

The factors affecting the energy efficiency of electric hoist belong to the multidimensional relationship matrix. For subsequent data analysis, statistical model and energy efficiency grading standard, this paper proposes a web data management system design using HBase to store multi-dimensional factor data. Simultaneously, MySQL and Redis are used to store regular data and hotspot data respectively. Combining the advantages of service and micro-services, the web back-end system in this paper adopts the SSM architecture, and divides the functions into common module subsystems (permissions, messages, logs, configurations, documents, etc.) and business subsystems (architecture, task scheduling, RPC, deployment, etc.). While ensuring soundness, the system coupling degree is the lowest and the scalability is guaranteed. Finally, the results of back-end system stress testing, database performance testing and HBase performance comparison testing show the optimal design effect of the management system. The system can meet the information management requirements of energy efficiency testing of various types of electric hoist in various conditions, and provide support for energy efficiency data analysis.

Keywords

Data Collection, Web Services, Storage Optimization, HBase

基于HBase的电动葫芦能测数据管理系统设计

梅俊¹, 王松雷², 章国宝¹

¹东南大学自动化学院, 江苏 南京

²江苏省特种设备安全监督检验研究院无锡分院, 江苏 无锡

Email: 220161393@seu.edu.cn

收稿日期: 2018年12月31日; 录用日期: 2019年1月15日; 发布日期: 2019年1月22日

摘要

针对影响电动葫芦能效的因素属于多维关系矩阵, 为了便于分析能效数据、建立统计模型以及指定能效分级标准等, 本文提出了基于HBase存储多维因素数据的后端数据管理系统设计方案, 运用优化配置的MySQL以及Redis分别存储常规数据以及热点数据。本文设计后端系统结合服务化与微服务的优点, 采用Spring Boot + MyBatis架构, 按功能划分为公共模块子系统(权限、消息、日志、配置、文档等)与业务子系统(架构、任务调度、RPC、部署等), 在保证健全性的同时, 使得系统耦合度最低, 可扩展性得到保障。最后, 后台压力测试、数据库性能测试、HBase性能对比测试的结果展现出了管理系统的优化设计效果, 系统可满足不同型号电动葫芦多种测试条件的能效检测的信息管理需求, 为能效数据分析提供支持。

关键词

数据采集, 后台服务, 存储优化, HBase

Copyright © 2019 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

特种设备的能源消耗巨大, 近年来国际上对于特种设备的能效监管与检测极为重视。作为特种设备重要的组成部分, 起重机械关于能效测试、能效分级等仍没有确立非试行的统一规范或标准[1]。电动葫芦的能效研究也将为后续展开的其他起重机械能效评价研究提供非常有益的借鉴[2]。

应江苏省特种设备检查监督研究院无锡分院的要求, 进行电动葫芦的能效测试及能耗分级, 能效影响因素涵盖多类: 电动葫芦型号(电机、电气、机械传动部分)、电机运行状态(运行速度、电机功率)、测试环境(不同工作级别、起重量、起升高度)等[3] [4]。但国内外对于起重机械类特种设备基本是关于数据检测终端、远程监控系统的研发, 电动葫芦能效因素统计模型、能效分级等数据分析理论都基于少量数据集的仿真基础建立, 使得最终结果缺乏说服力。在此背景下, 为了适用于多维因素大数据集进行数据分析、建立统计模型, 本文提出了基于HBase的能效检测数据存储系统设计方案, 根据配置注册电动葫芦型号以及其历史多维因素数据、常规数据, 通过RPC获取深度学习模块的最优选择后推送合适的检测任务, 任务调度模块适时采集热点数据并存储。

2. 后台设计

本文设计系统基于Maven构建、Jenkins作为持续集成, 项目后端使用Spring Boot (Spring + Spring MVC) + MyBatis框架搭建, 去除XML配置; 项目前端基于Angular JS + Bootstrap + jQuery框架, 使用Thymeleaf模板引擎生成复用静态页面。后端系统采用微服务化的特点[5], 分为业务与公共模块子系统, 各子系统均按功能模块化设计, 系统组件图如图1。

能效数据管理系统

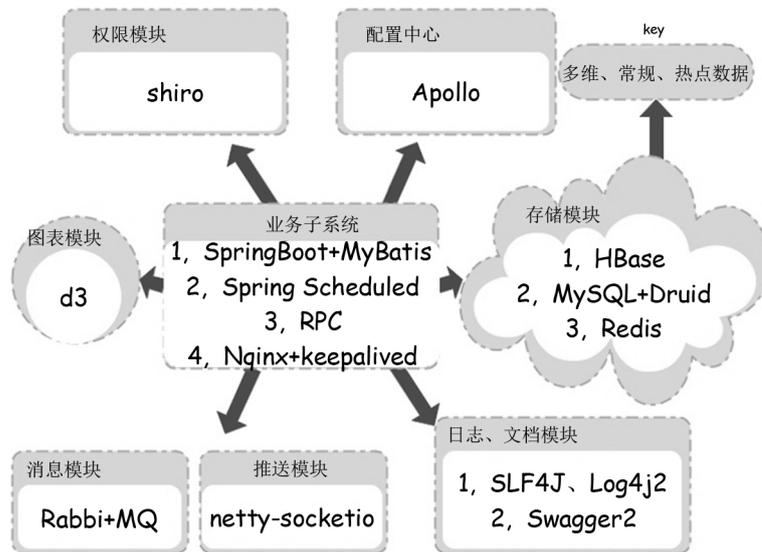


Figure 1. Energy efficiency detection data management system component diagram
图 1. 能测数据管理系统组件图

2.1. 业务子系统

业务子系统包括 SSM 架构、任务调度模块、RPC 调用模块、部署模块以及数据解析模块。各模块对应技术组件以及功能实现见表 1。

Table 1. Business subsystem function realization table
表 1. 业务子系统各模块功能实现表

模块	组件	功能实现
架构	SSM	详见 2.1.1 小节
任务调度	Spring Scheduled	结合 cron 表达式注解定期采集特定检测周期数据
RPC	Dubbo [6]	使得主业务子系统调用额外部署的远程模块，如数据分析模块等
部署	Nginx + keepalived	详见 2.1.2 小节
数据解析	FastJson	序列化与反序列化 json，使用 Fast Json Http Message Converter，提高 JSON 序列化速度

2.1.1. SSM 架构

SSM 框架是 Spring MVC，Spring 以及 MyBatis 的整合，属于 MVC 设计模式，系统架构按层次、模块划分，其中，Spring MVC 主要转发请求、分派视图，Spring 处理请求，MyBatis 持久化数据对象。能效数据管理系统基于此架构设计，架构层面较为突出的特点如下：

- 1) 使用 Model、Mapper、MapperXML、Service、ServiceImpl、Controller 等后缀命名模型层、持久层、控制层基础代码，使得代码结构清晰、层次分明；
- 2) 基于 Spring 注解，统一管理各 Exception 类型、创建 Rest 风格 Controller、AOP 切面处理事务等，去 XML 配置，配合注解模块化统一管理异常、Controller 业务方法 Bean、事务管理等；
- 3) 持久化集成 MyBatis、通用 Mapper 组件、Page Helper 物理分页组件，减少单表业务 SQL 开发量。

2.1.2. 部署模块

负载均衡使用 nginx 分发数据包与请求, 同时实现主从热备; 通过使用 keepalived 检测 nginx 状态从而达到高可用(HA, High Availability)的目的, nginx 与 keepalived 之间的关系如图 2。

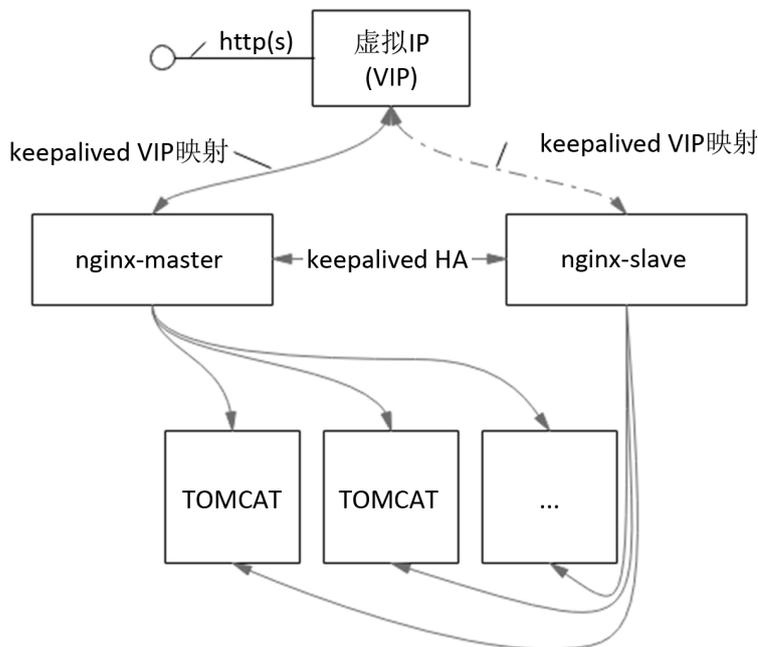


Figure 2. Nginx/Keepalived Component relation diagram

图 2. Nginx + keepalived 组件关系图

本文应用部署时使用虚拟 IP (VIP)作为用户访问、内部访问的统一入口, nginx 配置具体路由, 外界通过虚拟 IP 访问具体 Nginx 服务器, keepalived 配置后, nginx-master 会向 nginx-slave 持续发送心跳, 此心跳信息表征 nginx-master 的状态, 从而保证 Nginx 的高可用。

2.2. 公共模块子系统

公共模块子系统包括权限模块、配置中心模块、图表显示模块、消息模块、推送模块、日志模块、文档模块以及存储模块。各模块对应技术组件以及功能实现见表 2。

Table 2. Common module subsystem function realization table

表 2. 公共模块子系统各模块功能实现表

模块	组件	功能实现
权限	Shiro	详见 2.2.1 小节
配置中心	Apollo	管理多维因素数据、常规数据、热点数据所在的数据库服务器地址
图表显示	d3	图表组件实时显示采集能效相关数据
消息	Rabbit MQ	详见 2.2.2 小节
推送	netty-socketio	基于 Netty 框架的 Socket.io 服务器端的 java 实现, 允许客户端无请求的情况下在服务器端主动推送检测任务流程
日志	SLF4J + Log4j2	业务日志和调试日志分开打印, 同时基于时间和文件大小分割日志文件
文档	Swagger2	Swagger2 生成 RESTful APIs 文档
存储	HBase 等	具体实现与优化详见第三章

2.2.1. 权限模块

Apache Shiro 支持认证、授权、加密和会话管理，能测数据管理系统通过实现 Realm 定义系统内部的权限规则，Security Manager 通过从 Realm 获取用户的信息与权限认证验证 Subject。下述是权限模块的核心功能以及实现：

1) 使用配置文件定义 Realm 进行用户登录、URL 权限管理；

2) 常规方案仅使用 Redis 存取会话信息，本文使用 Shiro session 配合 Redis 实现“安全型”会话管理。

引入 spring-session-data-redis, @Enable Redis Http Session 配置 session 过期时间，redis 缓存会话数据实现分布式 session 同步，重启服务会话不丢失。

2.2.2. 消息模块

消息中间件采用 RabbitMQ [7]，使发消息者(多终端、各类型数据)与收消息者(各持久化服务)完全解耦，同时在发消息者与队列之间加了一层抽象——交换机，发消息者传输数据给交换机，交换机以“Direct”匹配 key 再转发的方式将数据投送至绑定的队列。

3. 存储设计

存储模块主要包括多维因素数据(机械传动部分、电机运行特性、测试环境等因素) HBase 存储、常规数据(电动葫芦基本信息注册数据、多种检测任务周期信息等) MySQL 存储以及热点数据(检测任务周期的实时数据) Redis 存储，实际部署环境拓扑图如图 3。

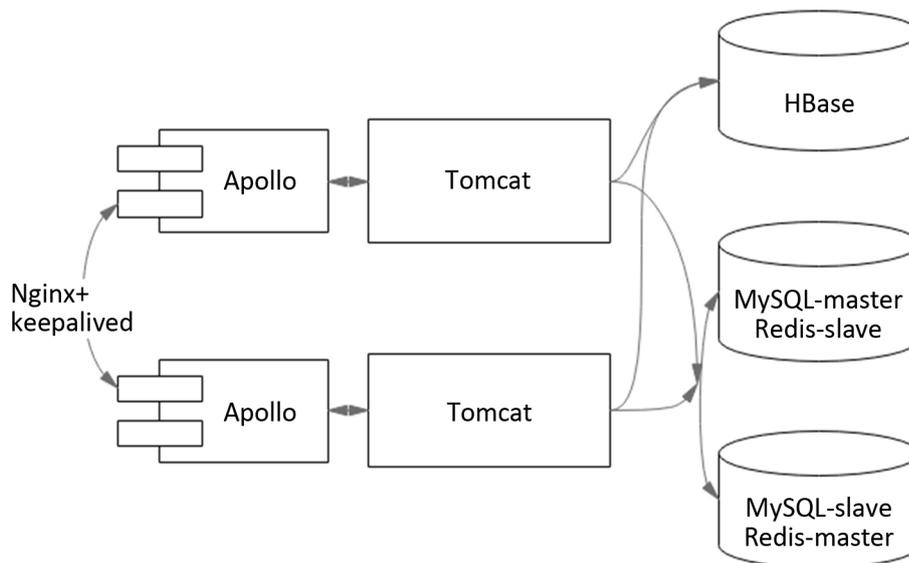


Figure 3. Topology diagram of the database deployment

图 3. 部署数据库拓扑图

3.1. 多维因素 HBase 数据库

HBase 是一个列存储、多维度[8]、可伸缩、结构稀疏的分布式 nosql 数据库，数学模型可以看成以行键(Row Key)、单元(Cell)、时间戳(timestamp)标识的 Map 数据结构[9]的数据库，其中，单元包括列族(Column Family)、列标识(Column Qualifier)，每个列族包括多个列标识，创建表时，仅需定义表名与列族，列标识也无需预定义，因此不同行键对应的同一列族的列标识可以不一致。定位数据也必须指定行键、列族名、列标识名、版本号(时间戳)。

列存储中查询中的索引规则是通过列来定义, 因此 HBase 中数据本身即索引, 每次查询只会访问涉及的列, 极大的降低了 IO 操作频率, 同时, 每一列的字段数据类型一致, 列聚集存储可以进行高效压缩, 节省物理空间。能效数据统计模型、能效分级等数据分析需求的数据量较为庞大, 出于存储、索引性能、多维因素存储的考虑, HBase 成为最佳选择。能测信息管理系统整合 HBase 存储多维因素数据步骤如下:

1) 在 Spring Boot 中引入 hbase-client 以及 spring-data-hadoop 依赖;

2) 设置好 quorum 与 port 后, 配置 Hbase Properties 与 HBase Configuration 获取 Hbase Template 与 Connection 实例; 应用 Connection 实例, 获取表名称(Table Name)、行键(Row Key)、列族(Column Family)、列标识(Column Qualifier)等信息实现 create、put、scan、get、count 等操作, 并封装成 Service 类;

3) 调用 HBase Service 类创建表, 列族(含有多个列标识)对应机械传动部分因素(电动葫芦型号、电机、传动摩擦)、电机运行状态(运行速度、电机功率)、测试环境(不同工作级别、起重量、起升高度)。

为了便于展示列族与列标识, 针对 Row Key 为 id1 时, 展开列族的数据结构见表 3。

Table 3. Attributes of column family in multidimensional factor data table

表 3. 多维因素数据表的列族属性

TableName	Row Key	Column Family	Column Qualifier
multid_factor	id1	mechanical	type, motor, friction
		electrical	speed, power
		environmental	level, weight, height

多维因素数据表 multid_factor 数据模型见表 4。

Table 4. HBase multidimensional factor data table multid_factor

表 4. HBase 多维因素数据表 multid_factor

Row Key	timestamp	Column Family: mechanical			Column Family: electrical		...
		column: type	column: motor	column: friction	column: speed	column: power	
id1	1539576300	{type:, motor:, friction:}			{speed:, power:}		...
id2	1539576330
...

优化配置措施:

1) 由于过多的 Column Family 之间会互相影响, 本系统限制每张表列族不超过 4 个; 列族名、列标识名的长度影响 I/O 读写的性能以及发送的数据量, 设置简洁。

2) Row Key 是 HBase 的 key-value 存储中的 key, HBase 通过对 Row Key 按字典序(存储排序)来建立索引, 本系统考虑存储数据 scan 范围较大的特点采取散列字段(高 2 个字节, 0~65,535 程序循环产生)+时间字段(低 4 个字节, 保留小时分钟毫秒)的组成方式优化 Row Key 的 Hash 以获得固定长度的 Row Key, 使数据分布均匀, 均衡负载。

3.2. 常规 MySQL 数据库

MySQL 是一个关系型数据库管理系统, 表结构涉及字段名称、字段类型、字段长度、是否为空、主键; 表操作支持 insert、delete、update、select (条件、排序、分组、多表等); 支持事务以及配置事务粒

度；表间关系支持多种关联映射；表数据有多种约束类型，如 primary key(PK)、unique 等。

能测信息管理系统使用 MyBatis 及相关组件整合 MySQL 步骤如下：

1) 在 Spring Boot 中引入 mybatis-spring-boot-starter, 添加 mysql、jdbc 相关依赖 spring-boot-starter-jdbc、mysql-connector-java;

2) 配置 MySQL 远程地址、登录信息、驱动名称、线程池信息等;

3) 创建常规数据(检测执行人、检测时间、工作周期等基本信息)、实时热点数据(时间、高度等)的实体类, 创建数据操作相关方法的接口 Interface, 使用通用 Mapper 组件配置生成对应的 MapperXML 进行数据库 CRUD 操作, 使用 PageHelper 组件实现物理分页; 同时, 可以创建 Service 进一步封装上述的数据操作相关方法, 对外仅暴露 Service, 降低系统模块耦合度;

4) 开启事务, 方法级别指定事务管理器, 配置事务传播行为、隔离级别、读写类型等;

检测任务相关信息表 detect_info 数据模型见表 5。

Table 5. Table data model of information related to detection task

表 5. 检测任务相关信息表数据模型

名称	类型	长度	NULL	PK	数据举例
id	int	11	not null	primary key	{1,2,3,4,...}
execStaff	varchar	24	null	no	{甲 乙 丙 ...}
workLevel	string	10	null	no	{M1 M2 ... M8}
operationSpeed	string	10	null	no	{Single Double}
updateTime	date	default	null	unique	{3.24,4.19,5.27,4.28,...}

优化配置措施:

1、使用 Druid Spring Boot Starter 集成 Alibaba Druid 数据库连接池与监控, Druid 数据库连接池, 监控数据库访问性能, 统计 SQL 的执行性能。

2、AOP 切换多数据源数据库, 实现读写分离。

3、@Transactional 注解事务, 实现非侵入式的方法级别的事务回滚。

3.3. 热点 Redis 数据

Redis 是一款支持 String、List、Hash、Set、Sorted Set 数据类型的内存高速缓存数据库, key-value 存储类型使得数据查询性能极高, 支持主从模式, 支持持久化, 支持事务, set/get、add/remove 等都是原子操作。

管理系统使用 Redis 存储检测任务周期的实时数据, 用以图表显示、即时分析、能效计算等, 并设置 key 的过期时间为检测周期的 2 倍, 过期之前不采用 redis 持久化, 而是转存于 MySQL 成为常规历史检测数据; 系统以主备方式保证 Redis 的 High Availability, redis-master 以异步 rehash 的方式保证 redis-slave 数据同步, 实现数据最终一致性。

能测信息管理系统整合 Redis 步骤如下:

1) 基于 Spring Boot 架构, 使用 spring-boot-starter-data-redis 引入 redis;

2) 添加配置文件, 设置 spring.redis.host、spring.redis.port 等 Redis 服务器与连接池信息;

3) 使用 RedisTemplate 进行 set、get、add、remove、hasKey 等操作。

数据模型见表 6。

Table 6. Data type stored by Redis
表 6. Redis 存储的数据类型

Key	类型	操作	子系统下指标常权
execTimeSec	list	lpush	{3.00,15.00,18.00,...}
workLevel	string	set	{M1 M2 ... M8}
operationSpeed	string	set	{Single Double}
height	list	lpush	{3.24,4.19,5.27,4.28,...}

优化配置措施:

- 1、redis-master 不做任何持久化工作，转存至 MySQL 代替 RDB 内存快照、AOF 日志文件。
- 2、主备方式代替主从方式，对外提供一个虚拟 IP(VIP)，keepalived 保证 Redis HA，WEB app 通过 VIP 进行存储操作，正常时主机 redis-master 提供服务，宕机时 VIP 自动漂移到备机 redis-slave 上；另外，主备数据同步使用异步同步，在性能得到保证的同时，最终一致性已足够需求。

4. 测试

4.1. 后台压力测试

后台完整测试一般需涵盖工具类等方法测试、Controller 类等 RESTful API 请求测试以及并发访问压力性能测试。其中基本方法测试基于 Junit4, RESTful API 测试基于 Mock 技术, 性能、效率测试基于 ContiPerf。

能测信息管理系统基于 ContiPerf 进行性能测试步骤如下:

- 1) 在 dev 环境下引入 org.databene.contiperf 组件进行性能测试;
- 2) 使用注解@PerfTest (invocations = 100, threads = 10)设置启动 10 个线程执行任务 100 次;
- 3) 运行测试, 访问: target/contiperf-report/index.html。

得到 ContiPerf 报表如图 4。

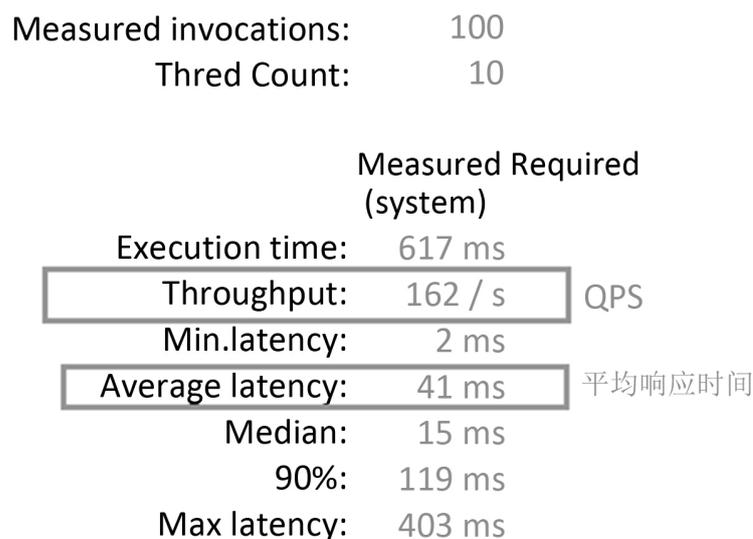


Figure 4. ContiPerf report of performance testing
图 4. ContiPerf 性能测试报告

图 4 中 throughput 显示每秒执行任务次数 162 次, 平均响应时间为 0.041 s, 性能满足需求。后端系统使用内嵌的 Tomcat 作为 Servlet 容器, 启动 ContiPerf 测试后用 VisualVM 查看应用所占的资源情况如图 5。

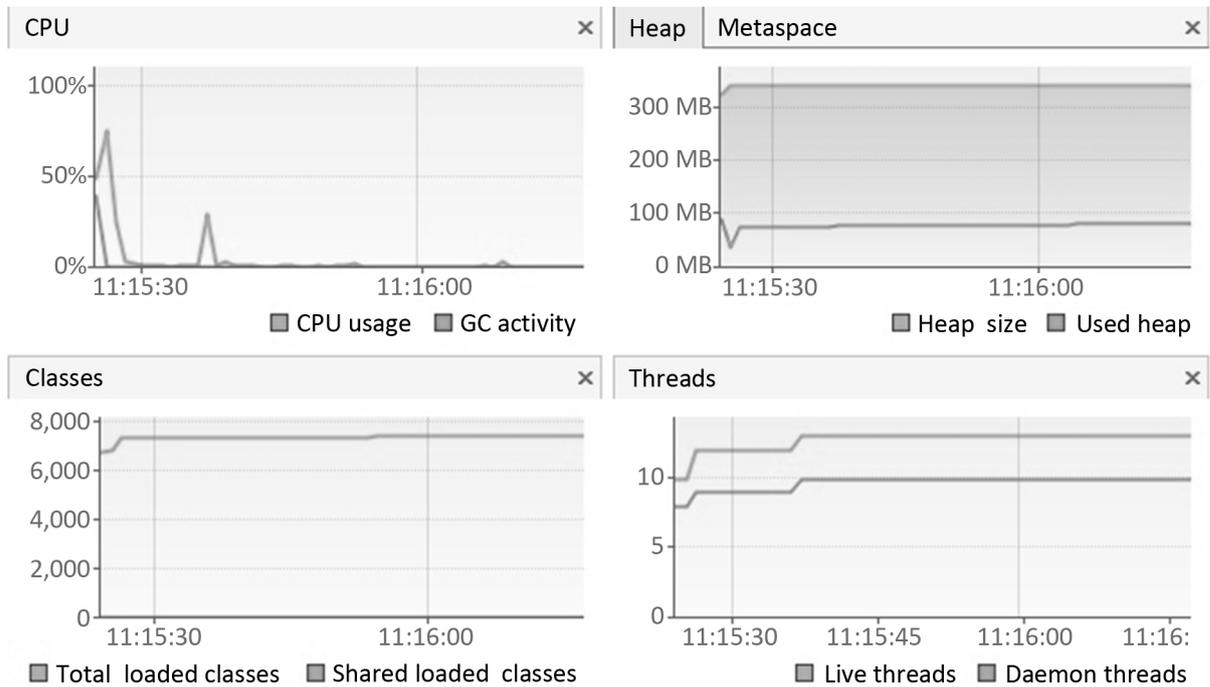


Figure 5. Monitoring information by VisualVM

图 5. VisualVM 监控信息

如上图所示，ContiPerf 开启测试后，Daemon threads 后台线程 10 个，使用的堆内存为约 90MB，对于资源使用情况符合预期。

4.2. 数据库性能测试

Apache JMeter 是一款开源的压力测试工具，支持数据库、http 请求等的压力测试，同时也支持混合应用场景多线程组并发测试，本文使用 JMeter 进行数据库压力测试，步骤如下：

- 1) 新建测试计划，添加线程组，设置线程数 100；
 - 2) 添加 JDBC Connection Configuration，设置待测远程数据库服务器地址以及登录信息；
 - 3) 添加 JDBC Request，设置待测 select 语句，本次测试以单表条件查询为例，select id, workLevel from detect_info where DATE(updateTime) >= DATE('2018-10-15') limit 15；
 - 4) 添加监听器，依次添加察看结果树与聚合报告；
- 配置完成，执行测试后，聚合报告结果如图 6。

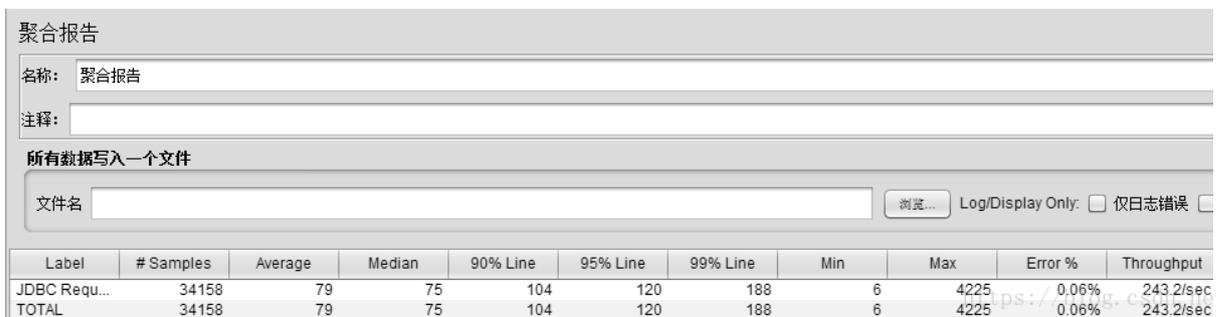


Figure 6. JMeter aggregation report of stress test results

图 6. JMeter 聚合报告压力测试结果

如图 6 所示, 总的 select 语句请求数 Samples 为 34158 条, Average 平均响应时间为 0.079s, Error% 测试出现的错误请求数百分比为 0.06%, Throughput(TPS, 吞吐量)每秒处理的请求数 243.2 条, 能承受较大的并发请求量。

4.3. HBase 性能对比测试

针对既有的检测数据 ER 模型管理系统解决方案[10], 本文参考该方案搭建 MySQL 数据库、表(建 mechanical、electrical、environmental 三类能效因素表, 并将 HBase 已 load 数据完全写入, 保证两者的测试数据量一样), 并使用 Druid 连接池、查询索引优化, 使用 Apache JMeter 压测三表联合查询语句 “select type, motor, friction, speed, power, level, weight, height from (mechanical left join electrical on mechanical.id=electrical.id) left join environmental on mechanical.id=environmental.id” 进行对比测试, 设置线程数 100。

对于本文设计的 HBase 多维因素数据表 multid_factor, 使用 “Yahoo!CloudServing Benchmark(YCSB)” 进行其数据服务性能基础测试, 步骤如下:

1) 配置 workload 类型为 workloadb, 即 Read mostly workload;

2) 设置 load 初始化 Hbase 表数据 “./ycsb load hbase10-P../workloads/workloadb-p threads = 100-cp /etc/hbase/conf-p table = multid_factor-p columnfamily = mechanical-p columnfamily = electrical-p columnfamily = environmental-p recordcount = 10000-s > ./load.report” ;

3) 运行负载测试 “./ycsb run hbase10 ...(如步骤 2)中所示) -s > ./run.report” 。

归纳 JMeter 聚合报告结果与 YCSB 命令行返回测试结果报告 run.report, 得出 MySQL 常规方案与本文设计 HBase 多维数据存储方案性能核心指标对比见表 7。

Table 7. Comparison table of the core indexes of MySQL and HBase performance
表 7. MySQL 与 HBase 性能核心指标对比表

DB	数据量	请求类型/数	吞吐量(ops/s)	AverageLatency (s)
MySQL	10,000	READ/1000	144.48	0.097
HBase	10,000	SELECT/1000	376.79	0.036

如表 7 所示, 关于多维的较大数据查询, 在相同数据量、请求数、线程数的压测情况下, 即使常规方案数据库针对多表的查询数据建立索引, 吞吐量以及平均响应时间仍然逊于 “数据本身即索引” 的 HBase 多维数据表。另外, 在多维因素增加维度的情况下, 常规库数据库多表查询的响应时间也会有所增长, 而 HBase 因为本身列存储的性质则不会因为增加列族而变化。

5. 结束语

本文基于 HBase、MySQL 以及 Redis 分别存储能效检测多维因素数据、常规数据以及热点数据, 以 SSM 架构, 结合配置中心、任务调度、日志文档、权限、消息、图表显示、推送等模块, 根据配置注册电动葫芦型号以及其历史多维因素数据、常规数据, 智能推送合适的检测任务, 任务调度适时采集热点数据并实时图表显示, 使用的消息队列支持多终端设备同时采集, 同时可生成日志报表。该系统设计探讨了多维度数据的存储、索引等优化方案, 具有较好的高可用性、可扩展性、高效率等特点, 可满足不同型号电动葫芦多种测试条件的能效检测的信息管理需求, 为相关人员提供便捷检测与技术支持。

基金项目

KJ(Y)2015018, 江苏省特检院资助项目。

参考文献

- [1] 张志坚. 电动单梁起重机质量安全分析与研究[D]: [硕士学位论文]. 杭州: 浙江工业大学, 2014.
- [2] 陶冶. 电动葫芦单梁桥式起重机的参数化设计及有限元分析[D]: [硕士学位论文]. 太原: 太原科技大学, 2014.
- [3] 贾薛铖, 王松雷, 徐榕, 等. 电动葫芦能效测试方法研究[J]. 起重运输机械, 2014(12): 11-15.
- [4] 秦伟皓. 电动葫芦能源利用效率测算方法研究[D]: [硕士学位论文]. 上海: 上海交通大学, 2014.
- [5] Suryotrisongko, H., Jayanto, D.P. and Tjahyanto, A. (2017) Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot. *Procedia Computer Science*, **124**, 736-743. <https://doi.org/10.1016/j.procs.2017.12.212>
- [6] 王琪, 卢军. 基于 Dubbo 的分布式能效管理系统的设计与实现[J]. 电子设计工程, 2018, 26(11): 174-178.
- [7] 鱼朝伟, 詹舒波. 基于 RabbitMQ 的异步全双工消息总线的实现[J]. 软件, 2016(2): 139-146.
- [8] Liang, S. and Yang, Y. (2016) Towards Performance Evaluation of Hbase Based Multidimensional Cloud Index. *IEEE International Conference on Computer Science & Network Technology*, Harbin, 19-20 December 2015, 629-632.
- [9] Vora, M.N. (2012) Hadoop-Hbase for Large-Scale Data. *IEEE International Conference on Computer Science & Network Technology*, Harbin, 24-26 December 2011, 601-605.
- [10] Satoto, K.I., Isnanto, R.R., Kridalukmana, R., *et al.* (2017) Optimizing MySQL Database System on Information Systems Research, Publications and Community Service. *IEEE International Conference on Information Technology*, Semarang, Indonesia, 19-20 October 2016, 1-5.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org