**天润智能互联网平台DaaS测试报告**

**编 写 人： 李振兴**

**审 核 人：** **杨锐**

**北京工业大数据创新中心有限公司**

**2018年 9 月6日**

**文档记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **修订** | **日期** | **作者** | **描述** |
| 0.1 | 2018-08-21 | 李振兴 | 大纲 |
| 0.2 | 2018-08-27 | 李振兴 | 修改性能测试需求 |
| 0.3 | 2018-09-03 | 李振兴 | 添加实时接入测试结果 |
| 0.4 | 2018-09-03 | 王川 | 添加实时查询测试结果 |
| 0.5 | 2018-09-04 | 郑春喜 | 添加批量分析测试结果 |
| 0.7 | 2018-09-05 | 李振兴 | 添加功能测试结果 |
| 1.0 | 2018-09-06 | 杨锐 | 审核文档，发布 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 测试环境 4](#_Toc524105211)

[1.1. 系统拓扑图 4](#_Toc524105212)

[1.2. 系统信息 4](#_Toc524105213)

[2. 功能测试 5](#_Toc524105214)

[3. 性能测试 10](#_Toc524105215)

[3.1. 数据接入 10](#_Toc524105216)

[3.1.1. 数据准备 10](#_Toc524105217)

[3.1.2. 测试结果 10](#_Toc524105218)

[3.2. 实时查询 10](#_Toc524105219)

[3.2.1. 数据准备 10](#_Toc524105220)

[3.2.2. 测试结果 11](#_Toc524105221)

[3.3. 批量分析 12](#_Toc524105222)

[3.3.1. 前置条件 12](#_Toc524105223)

[3.3.2. 数据准备 12](#_Toc524105224)

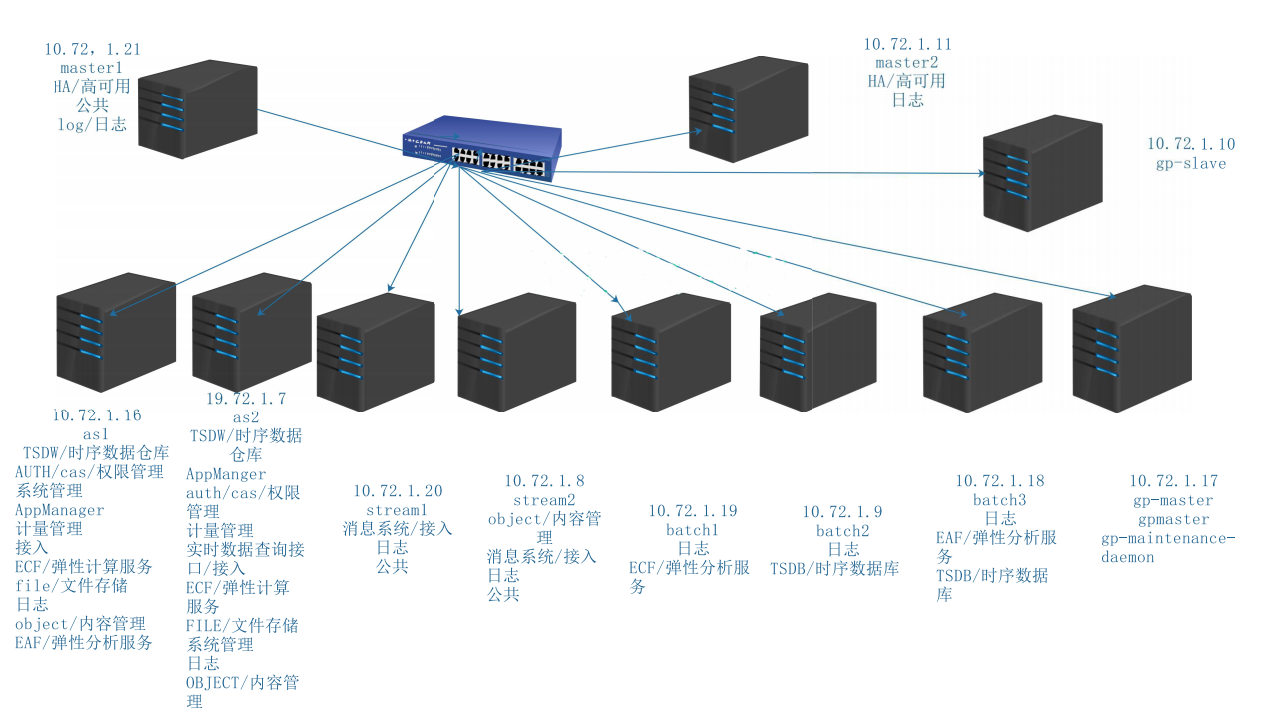
[3.3.3. 测试结果 12](#_Toc524105225)

[4. 附录 14](#_Toc524105226)

[4.1. Python脚本 14](#_Toc524105227)

[4.2. R脚本 14](#_Toc524105228)

1. 测试环境
   1. 系统拓扑图



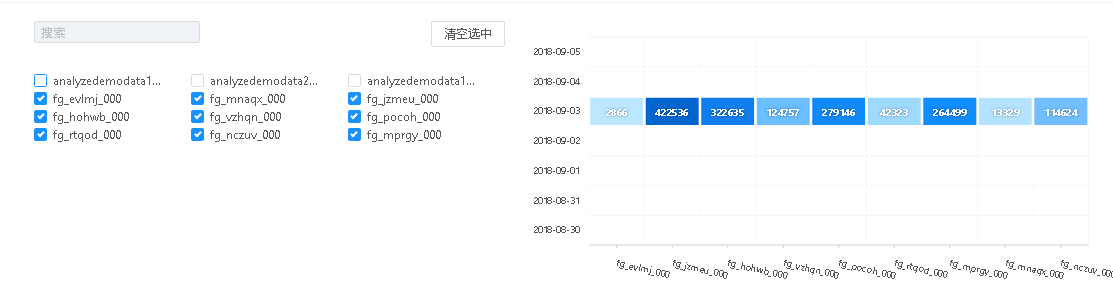
* 1. 系统信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **节点名称** | **IP** | **操作系统** | **CPU线程数** | **内存** |
| [kstone-7.rt.com](http://kstone-7.rt.com) | 10.72.1.7 | Ubuntu 14.04.2 | 48 | 256G |
| [kstone-8.rt.com](http://kstone-8.rt.com) | 10.72.1.8 | Ubuntu 14.04.2 | 48 | 128G |
| [kstone-9.rt.com](http://kstone-9.rt.com) | 10.72.1.9 | Ubuntu 14.04.2 | 48 | 256G |
| [kstone-10.rt.com](http://kstone-10.rt.com) | 10.72.1.10 | Ubuntu 14.04.2 | 48 | 128G |
| [kstone-11.rt.com](http://kstone-10.rt.com) | 10.72.1.11 | Ubuntu 14.04.2 | 48 | 128G |
| [kstone-16.rt.com](http://kstone-16.rt.com) | 10.72.1.16 | Ubuntu 14.04.2 | 48 | 256G |
| [kstone-17.rt.com](http://kstone-17.rt.com) | 10.72.1.17 | Ubuntu 14.04.2 | 48 | 128G |
| [kstone-18.rt.com](http://kstone-18.rt.com) | 10.72.1.18 | Ubuntu 14.04.2 | 48 | 256G |
| [kstone-19.rt.com](http://kstone-19.rt.com) | 10.72.1.19 | Ubuntu 14.04.2 | 48 | 256G |
| [kstone-20.rt.com](http://kstone-20.rt.com) | 10.72.1.20 | Ubuntu 14.04.2 | 48 | 128G |
| [kstone-21.rt.com](http://kstone-21.rt.com) | 10.72.1.21 | Ubuntu 14.04.2 | 48 | 128G |

1. 功能测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类型 | 功能 | 子功能 | 功能描述（基于硬件设备条件可支撑的功能） | 测试方法 | 测试结果 |
| DaaS | 数据接入 | 时序数据实时接入 | 提供HTTP，Kafka等实时接入接口适配功能 | 模拟实时数据通过不同接口接入数据平台,并检查数据是否存储成功. | 数据接入正常，测试通过 |
| 提供实时接入Java SDK功能 | 通过java sdk 编写程序模拟数据接入数据平台,并检查数据是否存储成功 | Sdk工作正常，数据正常接入，测试通过 |
| 提供故障容错功能，单点故障不影响数据接入 | 模拟数据介入,并手工停掉相应的数据接入节点,检查数据接入不受影响 | 停止服务后数据正常接入，恢复服务后，系统正常，测试通过 |
| 提供实时自动去重功能 | 模拟发送重复数据进入数据平台,最终在数据平台中查询是否去重 | 去重组件工作正常，重复数据被剔除，测试通过 |
| 时序数据批量导入 | 以界面形式提供批量导入功能 | 通过页面导入数据,测试批量导入数据功能 | 页面导入功能正常，测试通过 |
| 以界面形式提供定时自动导入功能 | 通过页面创建定时导入功能,并提供数据,测试定时导入 | 定时导入功能正常，测试通过 |
| 非结构化数据上传 | 以界面和接口形式提供非结构化数据上传功能 | 1,通过页面上传非结构化测试数据  2通过接口上传非结构化数据 | 非结构化数据上传正常，测试通过 |
| 数据质量核查 | 提供对于实时接入和批量导入的数据质量核查功能 | 1. 实时接入异常数据,通过运维页面的日志功能查看异常数据日志 2. 批量导入异常数据，通过页面的任务报告页面查看异常数据报告 | 异常数据核查功能正常，测试通过 |
| 数据管理 | 数据模型管理 | 提供元模型驱动的数据建模功能页面 | 通过页面创建时序数据表，验证该功能 | 时序数据表创建成功，测试通过 |
| 多类型数据存储 | 统一数据模型下管理的多引擎统，支持时序数据、对象数据、关系数据等的统一存储管理 | 在控制台页面进行统一模型管理验证 | 测试通过 |
| 时序数据管理 | 提供为工业传感数据优化的时间序列数据仓库功能 | 页面验证提供时序数据仓库功能,支持数据建模,数据查看/抽取功能 | 数据仓库页面显示正常，功能可用，测试通过 |
| 工业内容管理 | 提供为工业领域非结构化数据管理优化的管理功能 | 页面验证提供非结构化数据管理功能,支持数据建模,上传,下载功能 | 内容管理页面显示正常，功能可用，测试通过 |
| 数据分区和碎片整理 | 提供智能分区和碎片整理功能。 | 通过后台验证平台支持数据分区及碎片文件合并功能 | 分区及碎片整理功能工作正常，测试通过 |
| 数据分析 | 分析算法库服务 | 内置通用数据分析算法库 | 页面验证通用算法库 | 通用算法库页面显示正常，功能可用，测试通过 |
| 分析模型管理服务 | 提供分析模型管理功能 | 页面验证提供分析模型的新建/编辑/删除功能 | 分析模型功能验证通过 |
| 工业数据并行分析框架服务 | 提供R、Python、Matlab非侵入式并行自动并行化分析功能 | 执行分析脚本，通过添加分组信息验证并行分析功能 | 分析脚本并行化执行功能测试通过 |
| 提供Spark、MapReduce编程框架、数据访问SDK和执行框架功能 | 通过文档提供的编程框架编写demo程序,验证编程框架及执行功能 | 功能测试通过 |
| 可视化分析服务 | 提供可视化的异构数据关联分析和并行分析框架支持 | 通过弹性分析服务页面新建分析项目，并使用多种数据源验证可视化的异构数据关联分析和并行分析框架支持 | 功能测试通过 |
| 分析模型运行支持 | 支持分析模型的周期性运行 | 通过页面将分析项目发布为服务并创建定时任务执行该服务以验证此功能 | 服务可定时执行，测试通过 |
| 提供任务运行资源分配和隔离功能 | 通过页面配置资源队列以验证此功能 | 资源分配与隔离功能正常，测试通过 |
| 分析结果管理 | 提供多种形式的分析结果访问功能 | 验证可通过文件或接口形式访问 | 分析结果管理测试通过 |
| 数据访问 | 时序数据即席查询服务 | 提供包括原始数据查询、切片查询、采样查询、趋势查询等近线时序数据查询服务 | 页面执行相应查询验证该功能 | 即席查询功能测试通过 |
| 时序数据历史数据抽取服务 | 对时序数据仓库存储历史数据的抽取服务 | 通过页面执行抽取任务,并查看抽取结果以验证此功能 | 数据抽取功能测试通过 |
| 工业内容查询访问服务 | 以界面和接口方式提供工业内容查询、工业内容访问功能 | 1.通过页面访问工业内容管理功能  2通过接口访问工业内容管理功能 | 内容管理功能页面功能正常，  内容管理接口功能正常，测试通过 |
| 平台管理 | 平台组件运维 | 以界面形式提供数据平台组件一体化运维功能 | 通过运维管理员界面进行运维功能验证 | 运维管理员界面功能正常，测试通过 |
| 资源管理服务 | 以界面形式提供平台计算资源的管理和计算任务监控功能 | 在页面上查看任务列表和资源监控来验证该功能 | 资源、任务监控服务工作正常，测试通过 |

1. 性能测试
   1. 数据接入
      1. 数据准备

****

* + 1. 测试结果

**关键词:**

设备数量:并行发送数据的设备数

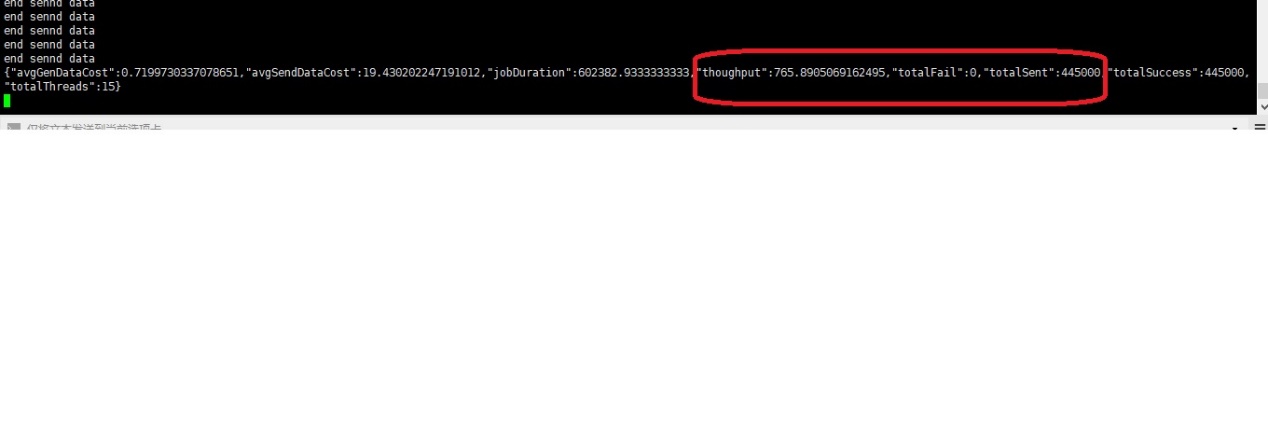
测点数:单条数据的列数

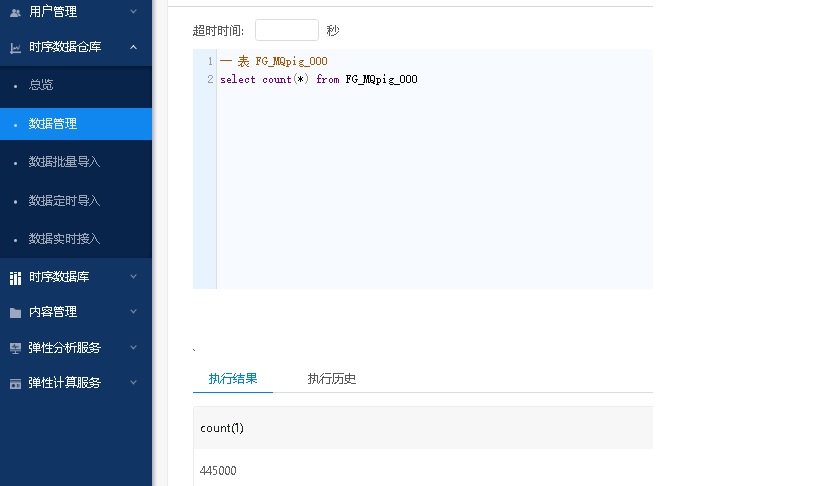
频率:设备数据发送的频率

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **数据特点** | | | **测试结果** | |
|  | **设备数量** | **测点数** | **频率** | **平均带宽** | **平均吞吐量** |
| 1 | 5000 | 1000 | 7s | 70M/s | 71万点/秒 |
| 2 | 10 | 1000 | 20ms | 40M/s | 50万点/秒 |
| 3 | 1000 | 1000 | 10mins | 1M/s | 1667点/秒 |
| 4 | 5000 | 1000 | 10mins | 5M/s | 8000点/秒 |

* + 1. 测试截图

仅以测试1为例：

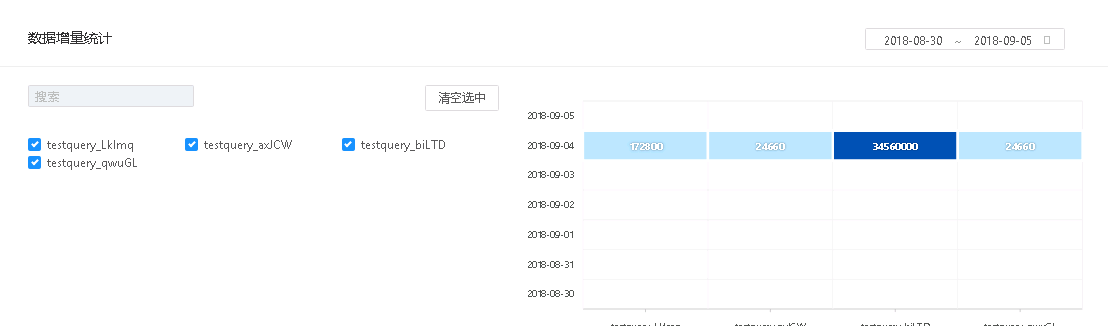








* 1. 实时查询
     1. 数据准备

****

* + 1. 测试结果
       1. 原始数据查询

**关键词:**

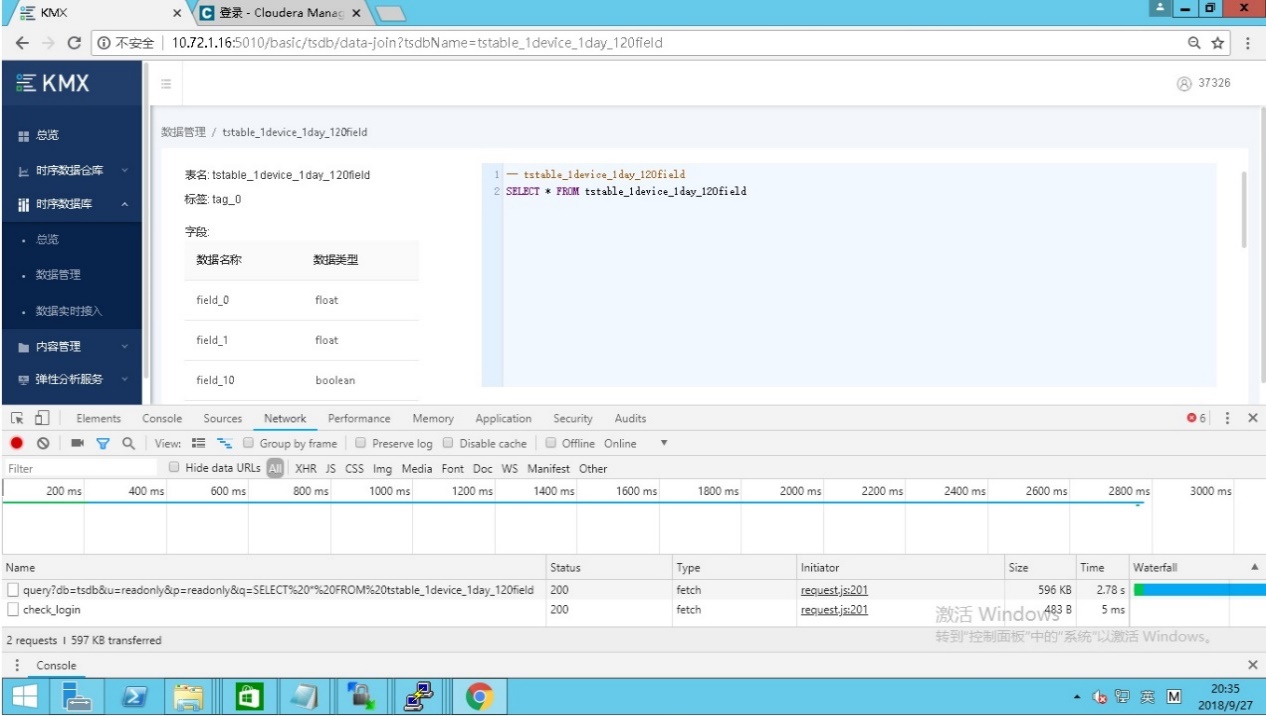
查询特点: 设备量,并发量,时间跨度,结果集大小

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **原始数据查询** | **描述** | **查询特点** | | | **测试结果** |
| **设备数量** | **返回行数** | **返回列数** | **响应时间** |
| 1 | 查询1个7s设备1天的120列数据 | 1 | 12342 | 120 | 2.8s |
| 2 | 查询1个7s设备1天的30列数据 | 1 | 12342 | 30 | 1.05s |
| 3 | 查询1个7s设备7天的300列数据 | 1 | 86400 | 300 | 36.75 |

* + - 1. 测试截图

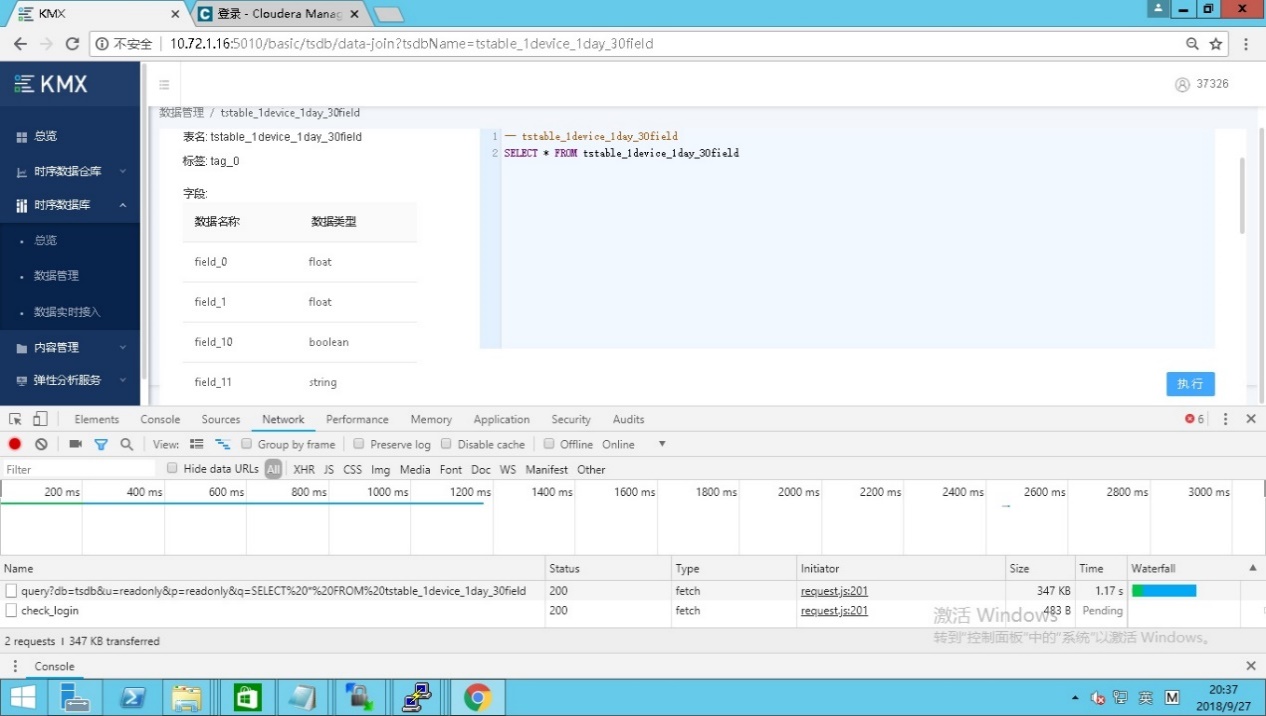
##### Q1

SELECT \* FROM tstable\_1device\_1day\_120field



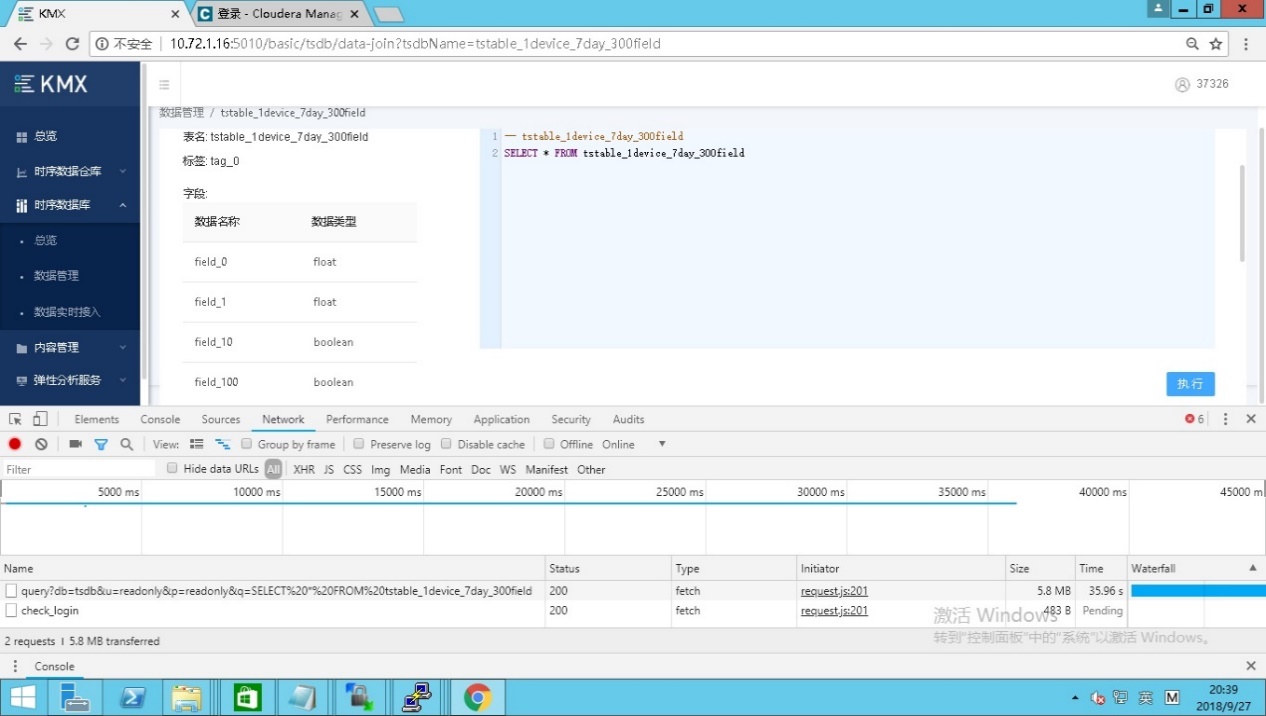
##### Q2

SELECT \* FROM tstable\_1device\_1day\_30field



##### Q3

SELECT \* FROM tstable\_1device\_7day\_300field



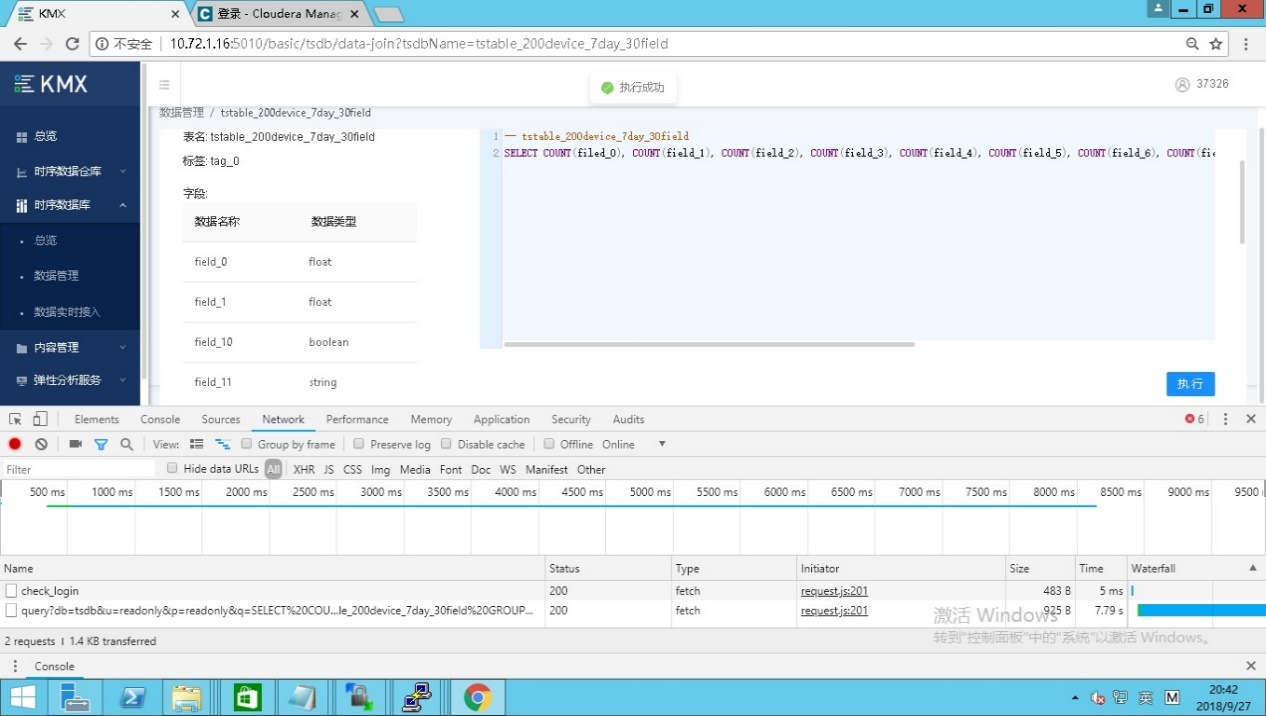
* + - 1. 聚合查询

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **聚合查询** | **数据规模** | **查询特点** | | | **测试结果** |
| **聚合方法** | **聚合粒度** | **聚合列数** | **响应时间** |
| 1 | 查询200个7s设备7天30列数据 | count | 1天 | 10 | 1.59s |
| 2 | 查询200个7s设备7天30列数据 | min | 1天 | 10 | 1.3s |
| 3 | 查询200个7s设备7天30列数据 | max | 1天 | 10 | 1.29s |
| 4 | 查询200个7s设备7天30列数据 | sum | 1天 | 10 | 2.58s |
| 5 | 查询1个7s设备7天300列数据 | sum | 1天 | 300 | 3.0s |
| 6 | 查询1个7s设备7天300列数据 | min | 1天 | 300 | 3.05s |
| 7 | 查询1个7s设备7天300列数据 | max | 1天 | 300 | 2.89s |
| 8 | 查询1个7s设备7天300列数据 | count | 1天 | 300 | 4.28s |

* + - 1. 测试截图

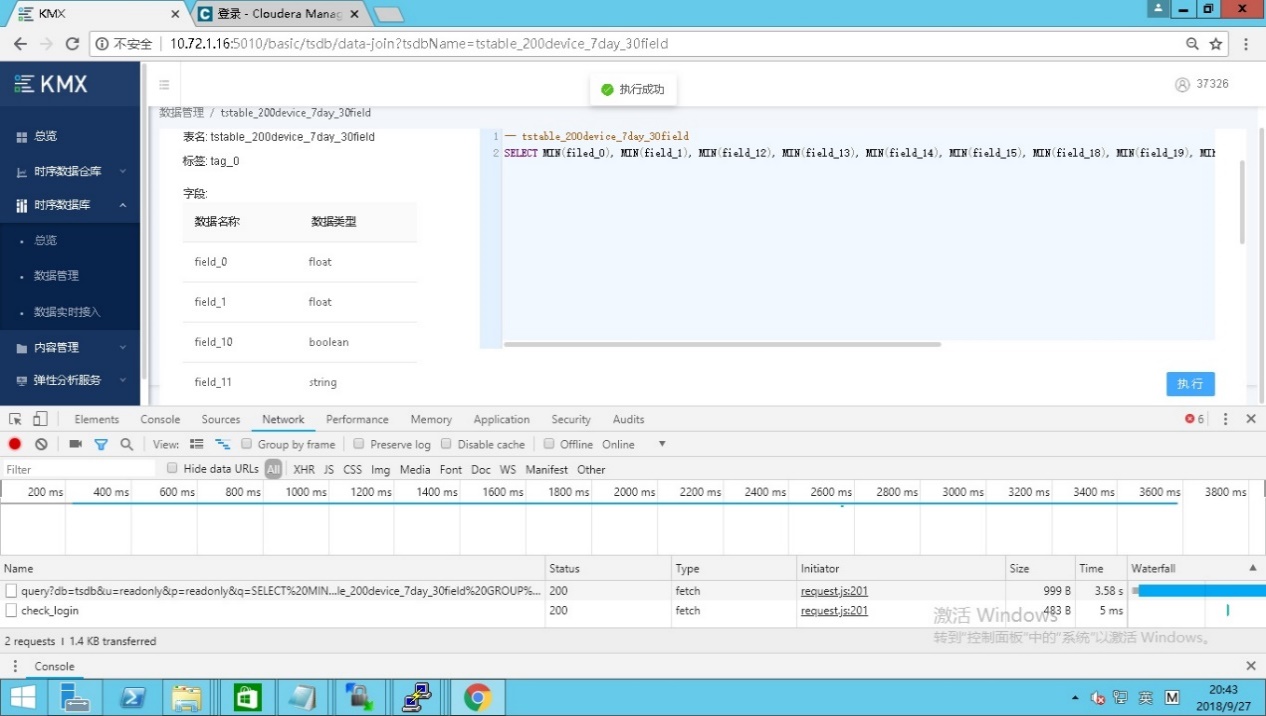
##### Q1

SELECT COUNT(filed\_0), COUNT(field\_1), COUNT(field\_2), COUNT(field\_3), COUNT(field\_4), COUNT(field\_5), COUNT(field\_6), COUNT(field\_7), COUNT(field\_8), COUNT(field\_9) FROM tstable\_200device\_7day\_30field GROUP BY time(1d)



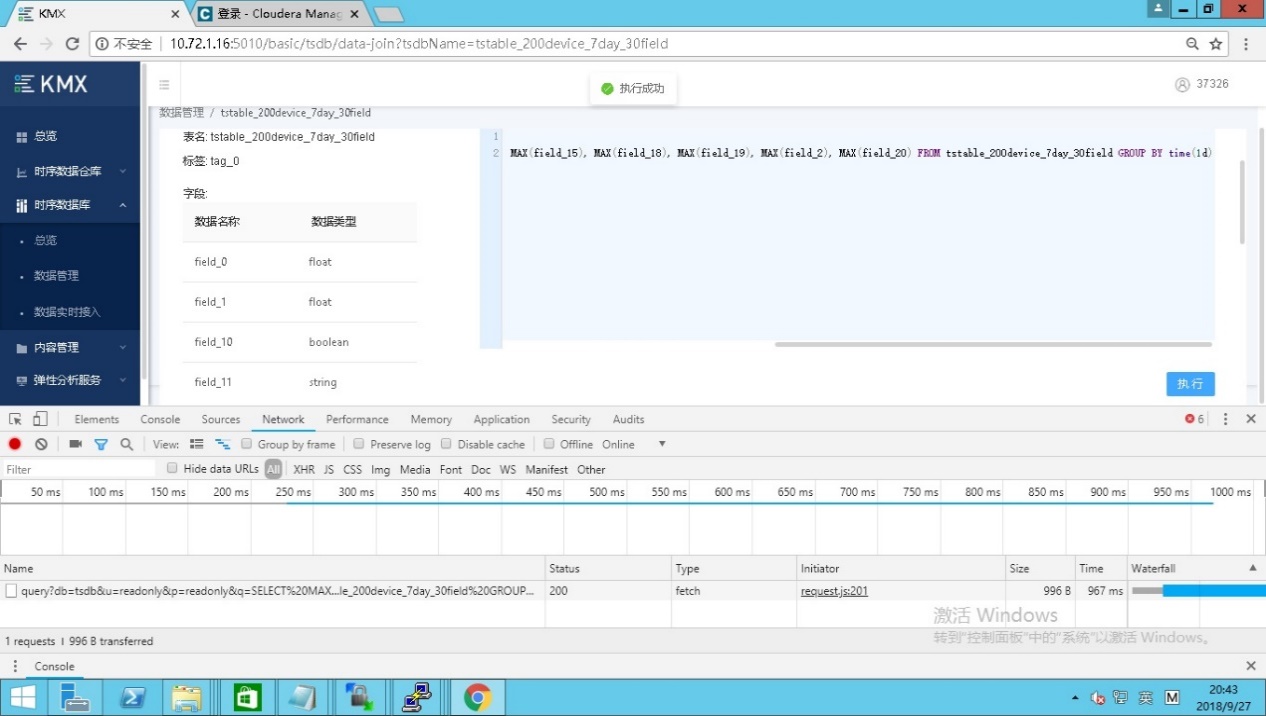
##### Q2

SELECT MIN(filed\_0), MIN(field\_1), MIN(field\_12), MIN(field\_13), MIN(field\_14), MIN(field\_15), MIN(field\_18), MIN(field\_19), MIN(field\_2), MIN(field\_20) FROM tstable\_200device\_7day\_30field GROUP BY time(1d)



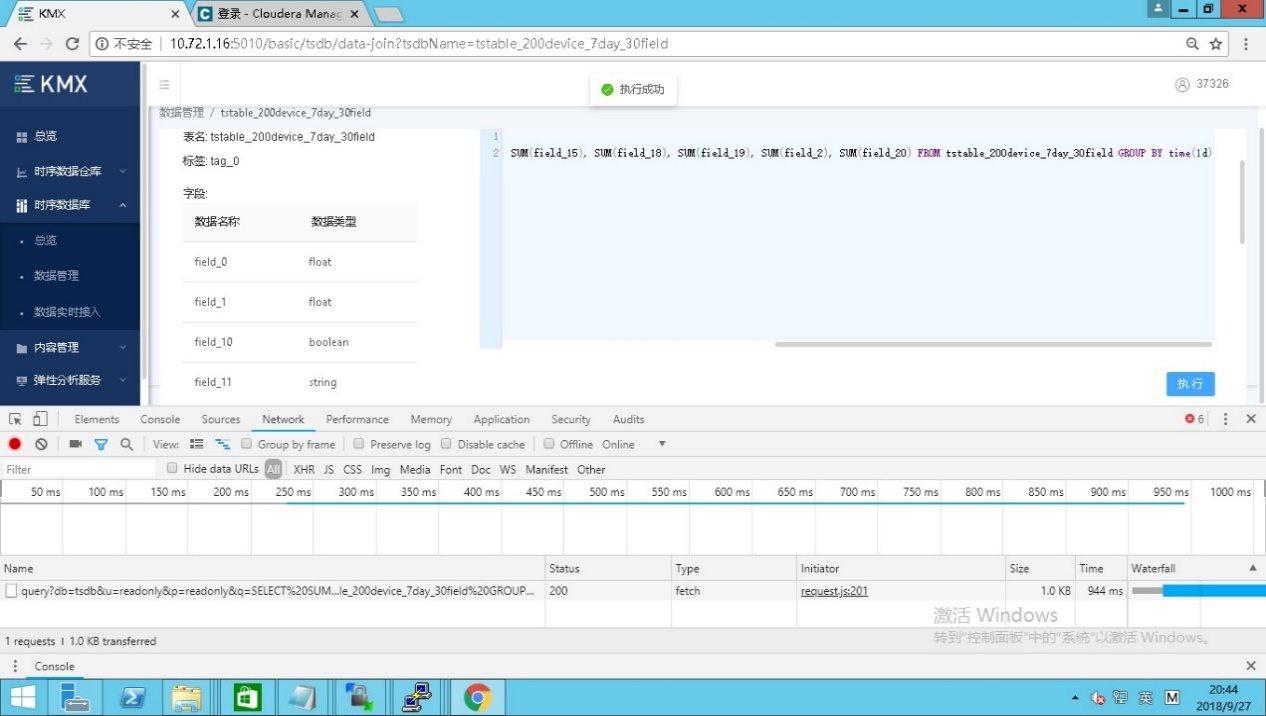
##### Q3

SELECT MAX(filed\_0), MAX(field\_1), MAX(field\_12), MAX(field\_13), MAX(field\_14), MAX(field\_15), MAX(field\_18), MAX(field\_19), MAX(field\_2), MAX(field\_20) FROM tstable\_200device\_7day\_30field GROUP BY time(1d)



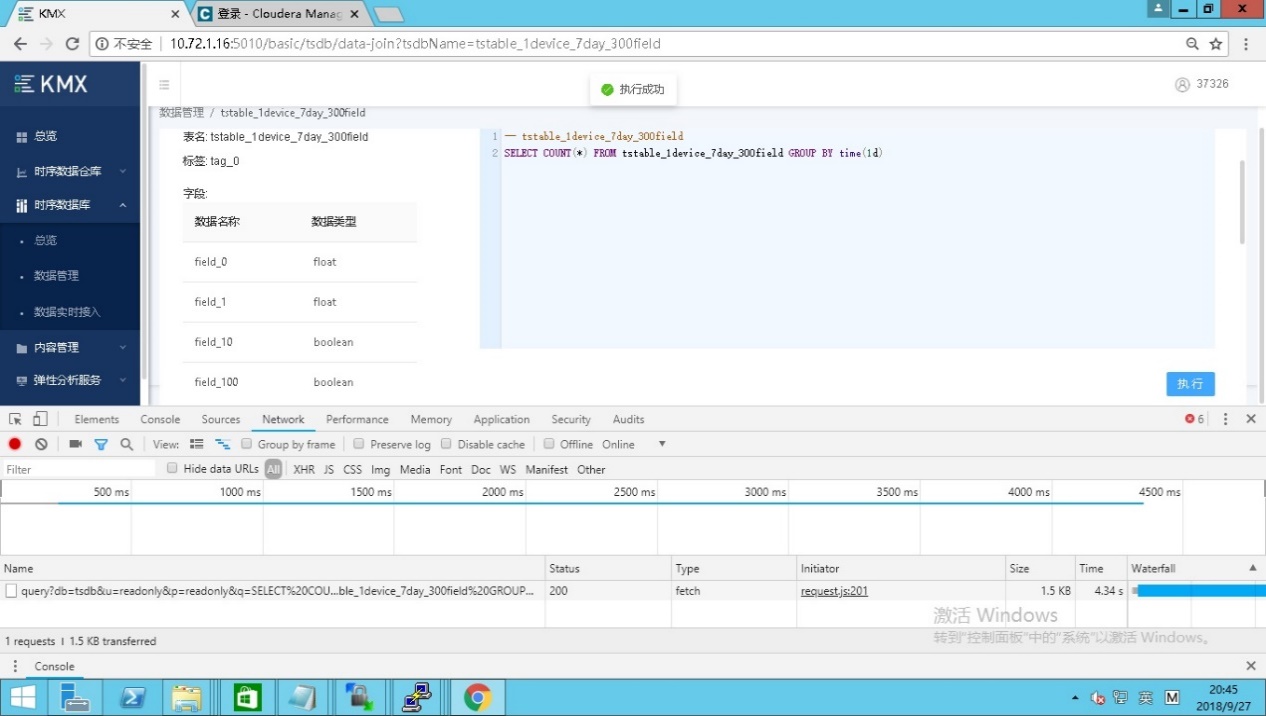
##### Q4

SELECT SUM(filed\_0), SUM(field\_1), SUM(field\_12), SUM(field\_13), SUM(field\_14), SUM(field\_15), SUM(field\_18), SUM(field\_19), SUM(field\_2), SUM(field\_20) FROM tstable\_200device\_7day\_30field GROUP BY time(1d)



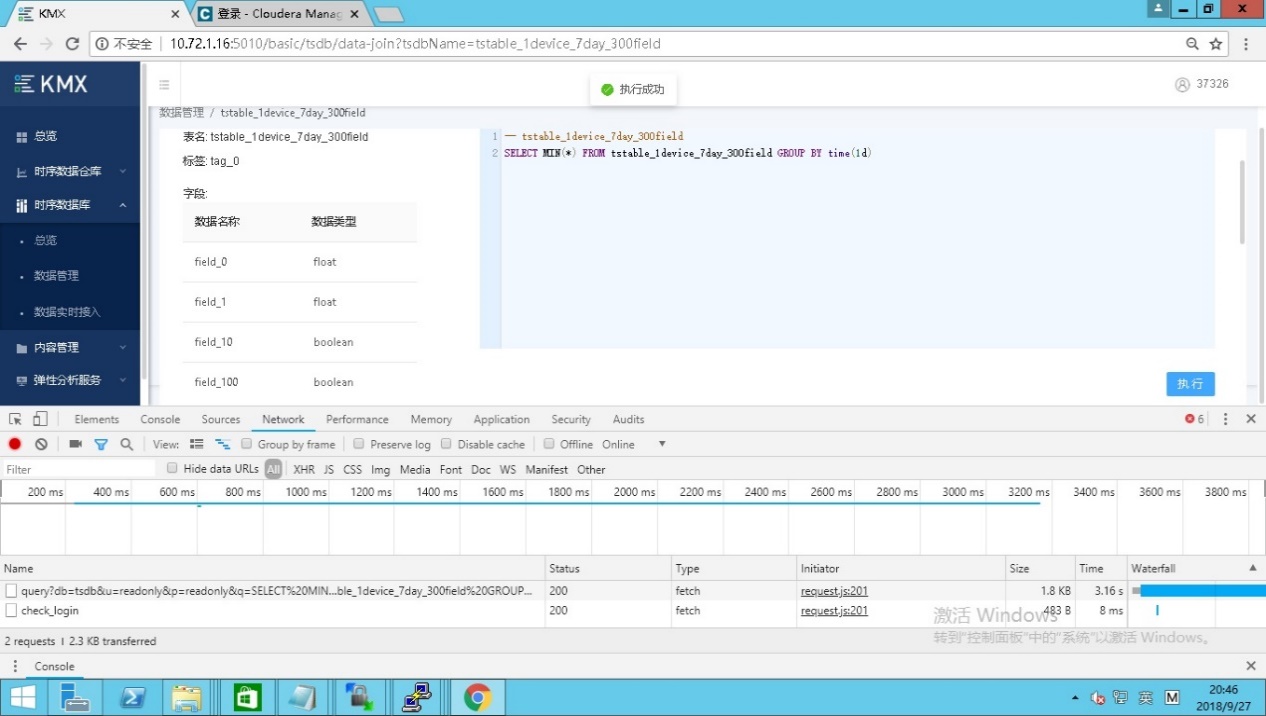
##### Q5

SELECT COUNT(\*) FROM tstable\_1device\_7day\_300field GROUP BY time(1d)



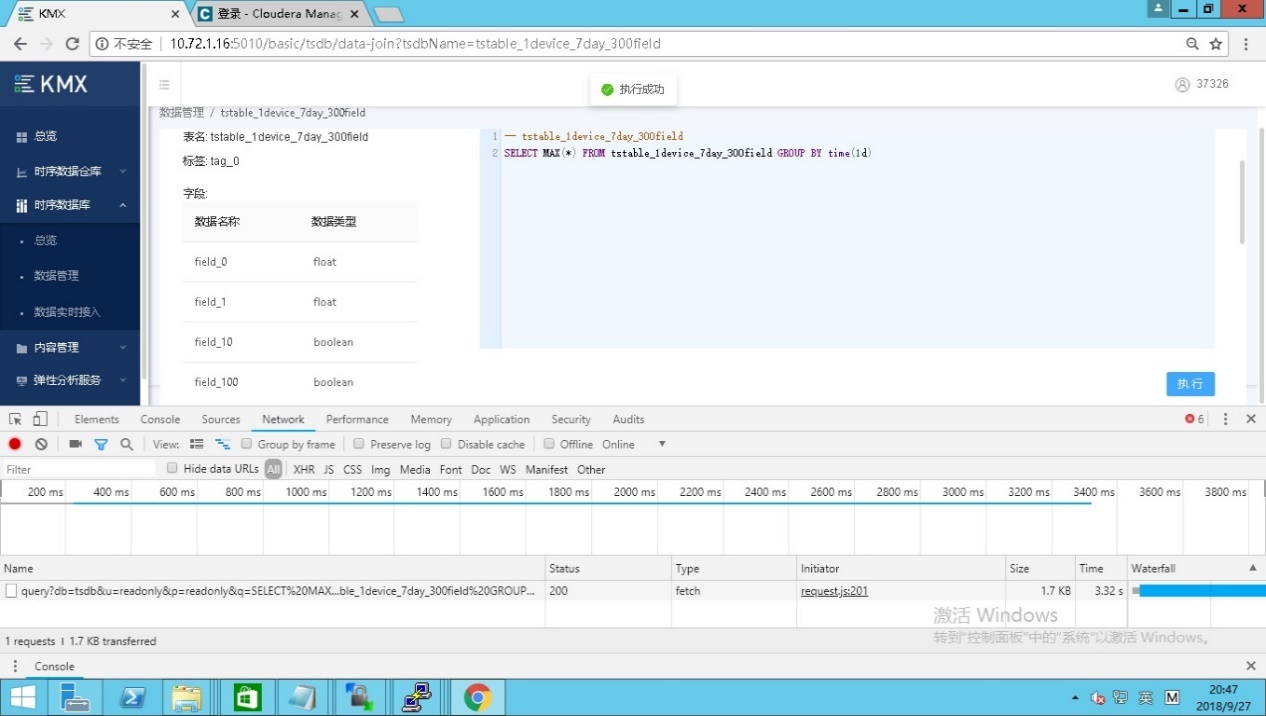
##### Q6

SELECT MIN(\*) FROM tstable\_1device\_7day\_300field GROUP BY time(1d)



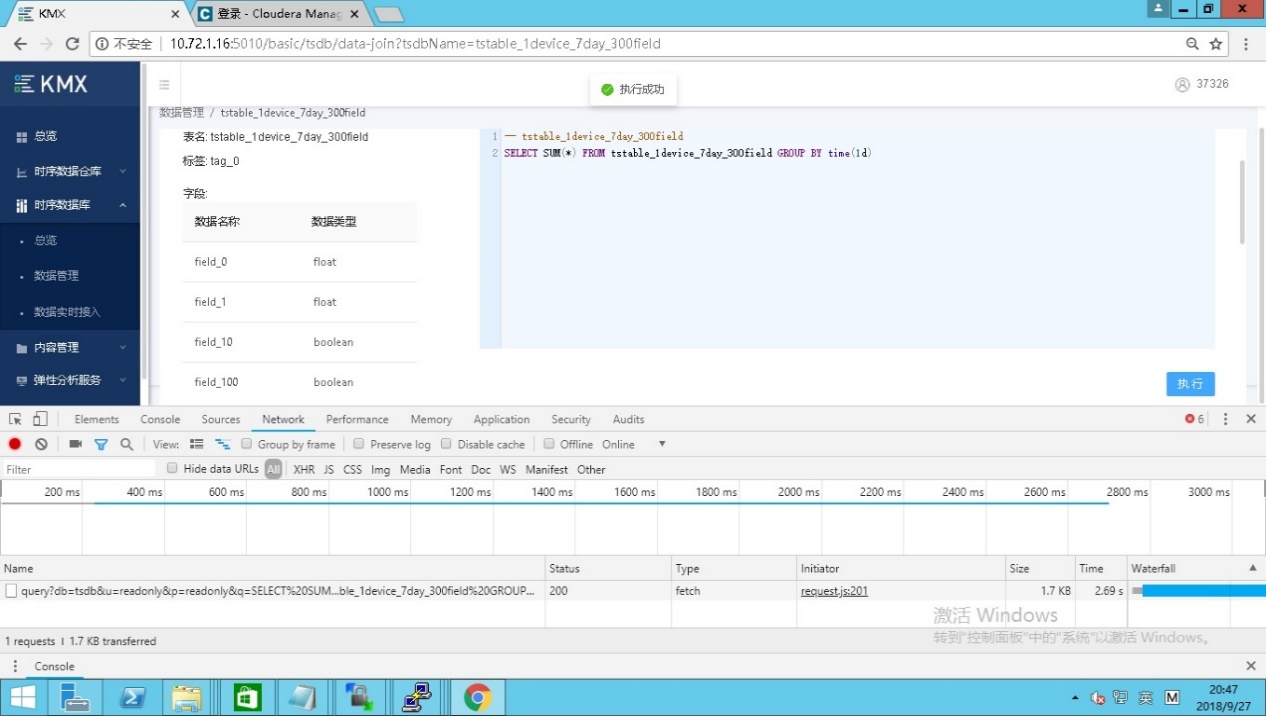
##### Q7

SELECT MAX(\*) FROM tstable\_1device\_7day\_300field GROUP BY time(1d)



##### Q8

SELECT SUM(\*) FROM tstable\_1device\_7day\_300field GROUP BY time(1d)



* 1. 批量分析
     1. 前置条件

1.使用客户提供的分析脚本

2.使用客户提供的数据模拟程序

* + 1. 数据准备



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **数据表** | **数据行数** | **数据文件大小（csv）** | **描述** |
| **1** | **Analyzedemodata100** | **37028500** | **3.2G** | **100个7秒设备30天数据** |
| **2** | **Analyzedemodata1000** | **370285000** | **32G** | **1000个7秒设备30天数据** |
| **3** | **Analyzedemodata200** | **444342799** | **38G** | **200个7秒设备180天数据** |

* + 1. 测试结果

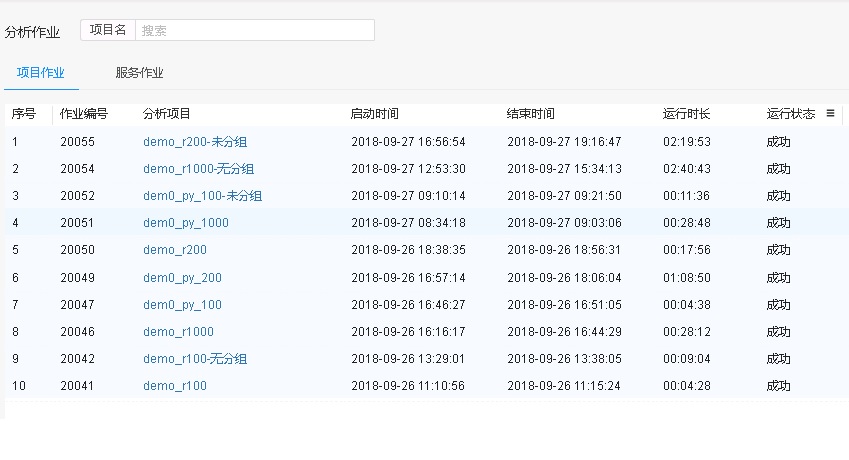
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **描述** | **数据集特点** | | | **测试结果** | |
|  | **设备数量** | **时间跨度** | **分组条件** | **执行结果** | **分析耗时** |
| 1 | 使用python脚本分析7s数据 | 100 | 30天 | 无分组 | 成功 | 11分31秒 |
| 2 | 使用python脚本分析7s数据 | 100 | 30天 | 按风机分组 | 成功 | 4分34秒 |
| 3 | 使用python脚本分析7s数据 | 1000 | 30天 | 无分组 | 抽取成功，分析失败 | n/a |
| 4 | 使用python脚本分析7s数据 | 1000 | 30天 | 按风机分组 | 成功 | 28分35秒 |
| 5 | 使用python脚本分析7s数据 | 200 | 180天 | 无分组 | 数据抽取成功，分析失败 | n/a |
| 6 | 使用python脚本分析7s数据 | 200 | 180天 | 按风机分组 | 成功 | 1小时18分03秒 |
|  |  | | | | | |
| 7 | 使用R脚本分析7s数据 | 100 | 30天 | 无分组 | 成功 | 9分7秒 |
| 8 | 使用R脚本分析7s数据 | 100 | 30天 | 按风机分组 | 成功 | 4分28秒 |
| 9 | 使用R脚本分析7s数据 | 1000 | 30天 | 无分组 | 成功 | 1小时27分21秒 |
| 10 | 使用R脚本分析7s数据 | 1000 | 30天 | 按风机分组 | 成功 | 28分37秒 |
| 11 | 使用R脚本分析7s数据 | 200 | 180天 | 无分组 | 成功 | 1小时47分钟20秒 |
| 12 | 使用R脚本分析7s数据 | 200 | 180天 | 按风机分组 | 成功 | 38分45秒 |

* + 1. 测试截图

原始测试结果：

****

补充测试结果：



1. 附录
   1. Python脚本

# -\*- coding: utf-8 -\*-

import os

import sys

def main(input\_file, output\_file):

results=[]

title0=['date','mean']

output\_file = output\_file + "result"

results.append(','.join(title0)+'\n')

data\_lines=[each.strip().split(',') for each in open(input\_file[0])][1:]

TimeStamp\_date=[eachline[0].split(' ')[0] for eachline in data\_lines]

speed=[eachline[4] for eachline in data\_lines]

UniDate=list(set(TimeStamp\_date))

for eachday in UniDate:

temp\_da=[float(each[4]) for each in data\_lines if each[0].startswith(eachday) ]

temp\_mean=round(sum(temp\_da)/len(temp\_da),3)

results.append(','.join([eachday,str(temp\_mean)])+'\n')

open(output\_file, 'w').writelines(results)

#if \_\_name\_\_=='\_\_main\_\_':

# main(sys.argv[1], sys.argv[2])

* 1. R脚本

library(plyr)

mainfunction<-function(input,output){

data0 <- read.csv(input,stringsAsFactors=FALSE,header=TRUE)

wt\_ts<-data0[[1]]

ts\_sl<-strsplit(as.character(wt\_ts), " ", fixed=TRUE)

ts\_date<-matrix(unlist(ts\_sl), ncol=2, byrow=TRUE)[,1]

uni\_date<-levels(as.factor(ts\_date))

data1<-data0[,4:5]

data1$dates<-ts\_date

record\_mean<-function(dataf){

return(data.frame(sp\_mean=round(mean(dataf[[2]]),4)))

}

results<-ddply(data1, .(dates), record\_mean)

dir\_path<- dirname(output)

file\_name<-paste0(dir\_path,"/xch\_wt07\_sp\_avg.csv")

write.csv(results, file=file\_name, quote=FALSE, row.names=FALSE )

return(file\_name)

}