



携程大数据平台实践



自我介绍

张翼

超过10年的互联网从业经验，超过7年的数据系统相关的经验

浙江大学：本科 & 研究生

Ebay：软件开发工程师

大众点评：资深软件开发工程师 -> 技术经理

携程：大数据平台技术总监



提纲

大数据平台现状

“成长的烦恼”

再谈稳定性

系统“走马观花”

挑战和未来



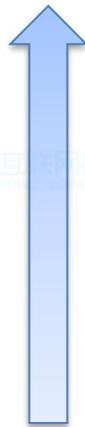
大数据平台现状



平台规模

- 主集群规模 180 → 1100+台 X6
- 数据增量（每天） 250T
- 数据表数量 正式表60000+
- 调度任务数（每天） 50000+ 运行实例120000+
- 底层任务数（每天） 310000+
- 实时集群规模 100+
- 实时作业数 290+ Jstorm ~40+ Spark-streaming

2017



2015

平台架构



OPS/IT/CC

全球互联网技术大会

全球互联网技术大会

全球互联网技术大会

开发平台Zeus

调度

主数据

传输

数据质量

查询平台ART

报表/OLAP
分析

即时查询

机器学习 算法平台

基于Spark图
形化AI平台

GPU云平台

实时数据平台Maise

分布式存储和计算框架

Hive

Spark

HBase

Presto

Kylin

Hadoop

实时框架

JStorm

Spark-
Streaming

Hermes (Kafka)

资源部署和运维监控

自动运维系统

大数据监控系统

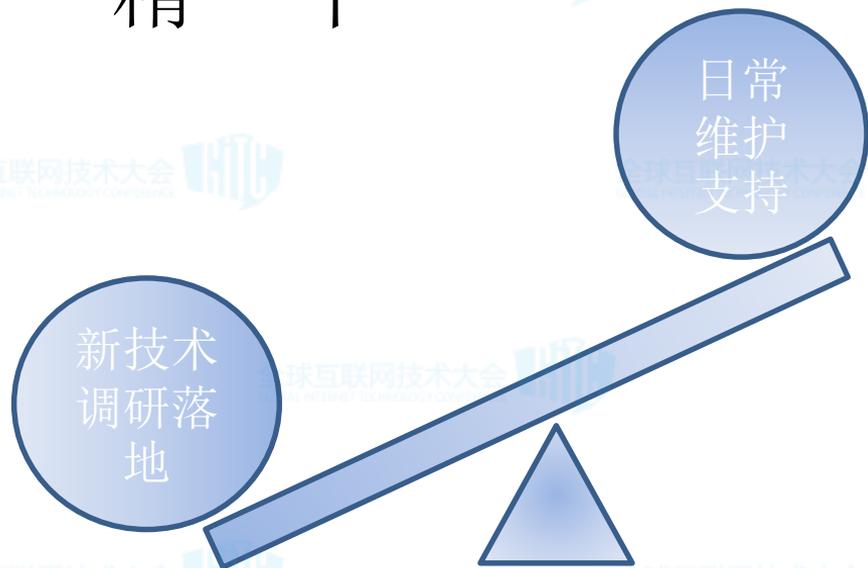
REN SHAO SHI DUO

小 而 精 干

底层数据架构：9 + 1

开发和查询平台：6 + 1

运维数据分析：4 + 1*



“成长的烦恼”

“成长的烦恼”有什么？

运维：

- 系统规模不断扩大
- 系统繁多，复杂性高



开源系统

- 开源是把“双刃剑”
 - 快速构建起相应的系统
 - 随着系统规模的增大，开源系统的问题不断地暴露出来

服务和支持

- 用户不断增长的“物质文化需求”与“短小精悍”团队之间的矛盾
- 临时的支持，问题排查工作变多

总体策略:

- “**自动化**”：节省运维成本，保证环境和配置一致
 - 运维自动化
 - 初始安装 / 变更
 - **覆盖范围尽可能全（特别是客户端）**
 - 监控 + 失败的自动回复
 - 确定的，风险不大的失败点（进程监控 / Thrift Server的可用性监控）
 - 多次自动回复失败需要升级
- 我们的惨痛教训：2015-09 Kerberos升级

思想上做好长期斗争的准备

“深挖洞（加深对现有系统的理解）”，“广积粮（基础知识/新系统调研）”！

总体策略：

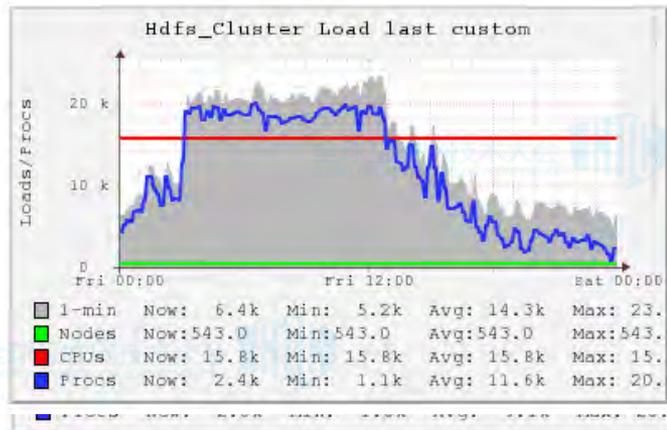
- 建立“代码级”维护能力
 - 招聘时就要选择对技术有浓厚兴趣，能够沉的下心来的同学
 - 在底层团队通过各种层次的分享建立学习，研究的氛围
 - 代码学习小组
- 全员学习，模糊职位的边界
 - 培养方向：一专多能
 - 模糊开发和运维的边界

Hadoop调优是一项长期工程

从2016年10月开始（CDH4.6 -> CDH 5.7.1升级完成, 79个commits）

我们几乎每1-2个月会遇到1个影响集群的稳定性/效率的问题，而且每次问题的Root Cause往往并不相同

实例一：RM调优

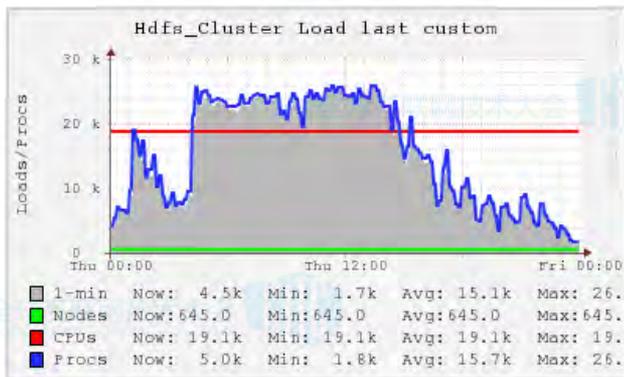


在业务高峰的4点-10点，集群的使用率偏低
通过YARN的主页面我们发现，集群的Used的Vcores只占Vcores Total的70%-80%

通过一段时间的分析，我们发现瓶颈在YARN的Fair Scheduler的效率上

<https://issues.apache.org/jira/browse/YARN-5188>

实例二：NN优化



2017-01底到2017-02初，我们发现在早上6-10点，集群的利用效率有多次较大的下跌

我们分析发现，NN的RPC平均处理时间（RpcProcessingTimeAvgTime）较高
解决方法：给NN减负，增效

[HDFS-9198](#) [HDFS-7964](#) [HADOOP-12483](#)

主节点优化的总结

1. **发现问题**：关注集群总体的利用率；关注NN和RM的关键指标（RPC Process Time / Call Queue Length）和GC指标
2. **分析问题**：分析NN和RM更加细致的指标（GC问题的话分析GC Log）；通过线索在去搜索相应的Jira，筛选Jira，通过Jira查看和分析相关的Code
3. 在保证稳定性的前提下进行**尝试**

总体策略:

- 从使用者的角度去设计产品，关注用户的易用性
- 控制推广的节奏
 - Early Adopter -> 1个BU ->全面铺开
- 完善文档，常见问题FAQ
- “授人以渔”：增强BU数据开发的工程技术能力
- 全员客服 – 短期



再谈稳定性



平台的稳定性是项系统工程

流程

- 合理的流程能够降低人为事故发生的概率，及时发现问题
- 流程无法一簇而就，从失败中学习，举一反三很重要
- 能做“灰度”的一定要做“灰度”
- 提升团队成员的责任心

技术

- 设计尽量简单，复杂的设计修改容易出错
- 合适的重试机制帮助很大
- 木桶效应：关注所有有影响的系统的短板

实例：数据开发平台的稳定性提升

2015年下半年，数据开发平台几乎每个月都会发生1次全局性延迟的大故障

第1阶段：

- 快速止损：通过发布后轮流值班的方式避免发布后的重大事故

第2阶段：

- 解决核心问题：重构调度系统的核心，简化模型

第3阶段：

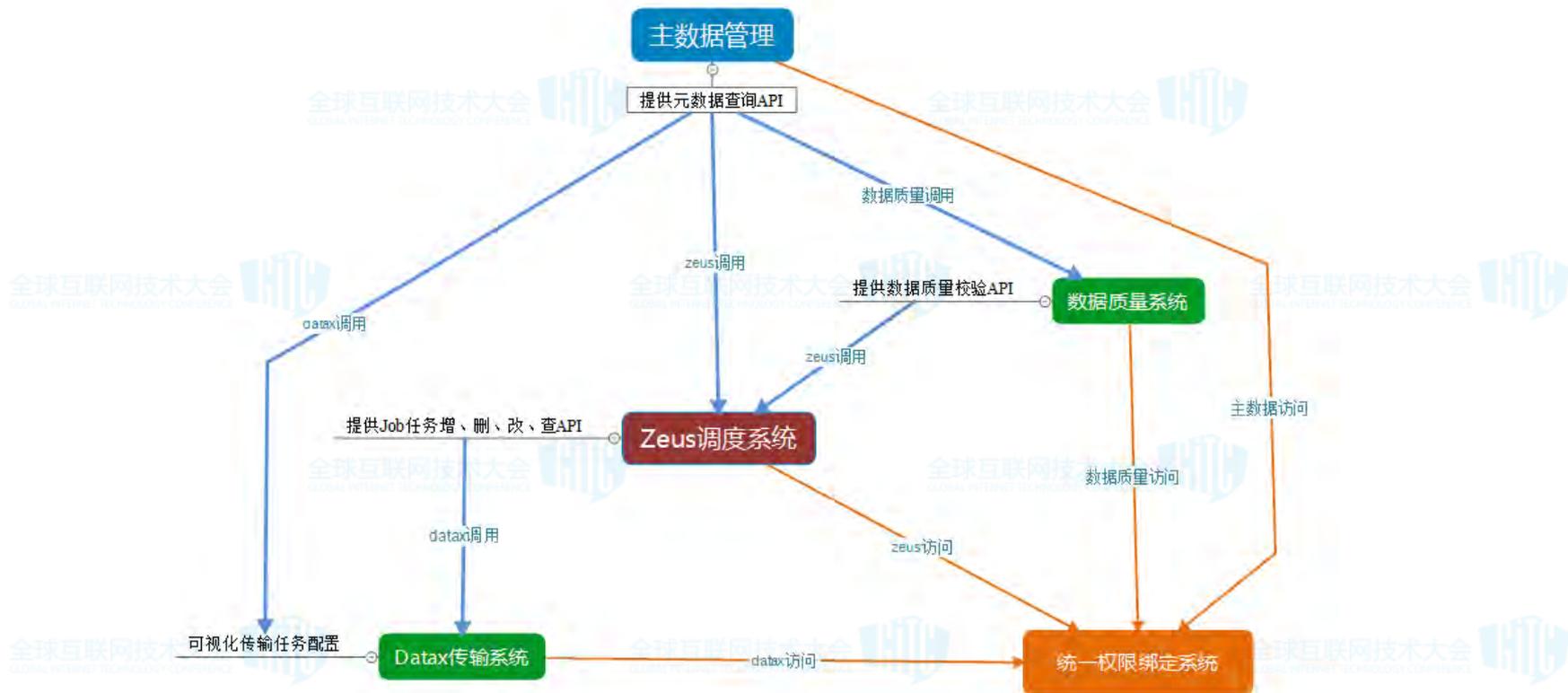
- 降低维护成本：监控 + 自动化恢复策略
- 确定目标，不断迭代
 - 区分系统错误，用户错误；确定系统错误率0.02%的目标（目前是0.01%）
 - 强化各层的重试机制（调度系统 / YARN APP Master ...）
 - 持续的监控和优化



系统 “走马观花”



数据开发平台 “走马观花”



调度和开发

The screenshot displays the Zeus 2.0 Data Development Platform interface. The main area shows a job configuration for a task named '00000127-Lab-y'. The configuration is as follows:

```
1
2
3 set -B
4
5 cat <&EOT = canus_properties
6
7 canus_job_name=canus
8
9 kafka_brokers=
10
11 etl_destination_path=/user/opsapp/canus/core_metrics/ml/data
12 etl_execution_base_path=/user/opsapp/canus/core_metrics/ml/exec
13 etl_execution_history_path=/user/opsapp/canus/core_metrics/ml/history
14
15 canus_message_decoder.class=com.linkedin.canus.etl.kafka.coders.JsonStringMessageDecoder
16 etl_record_writer_provider.class=com.linkedin.canus.etl.kafka.common.KCFileAppenderProvider
17 etl_partitioner.class=com.linkedin.canus.etl.kafka.partitioners.HourlyPartitioner
18
19 kafka_max_poll_msecs=21
20 kafka_max_history_col_days=3
21 kafka_max_poll_msecs_per_task=1
22 kafka_whitelist_topics=core_metrics
23
24 kafka_new_to_earliest_offset=true
25 kafka_client_name=canus
26
27 etl_default_timezone=Asia/Shanghai
28 etl_execution_history_max_of_quits=1
29
30 canus_json_type=ORIGINAL
31 canus_json_keep_origins=true
32
33 canus_work_allocator.class=com.linkedin.canus.workallocator.SingleTopicPartitionsAllocator
34
35 ...
```

调度和开发

The screenshot displays the Zeus 2.0 Data Development Platform interface. The main content area is titled "任务配置" (Job Configuration) and shows details for a job named "dw_data_ots_app(3024)".

任务配置 (Job Configuration):

- 任务名称: dw_data_ots_app(3024)
- 任务ID: 3024
- 任务类型: hive 脚本
- 首次调度: 2014-11-11 10:00:00
- 调度周期: 每天
- 调度时间: 02:00
- 调度失败重试: 3次
- 调度失败重试间隔: 10分钟
- 调度失败重试策略: 立即
- 调度失败重试通知: 否
- 调度失败重试通知邮件: 否
- 调度失败重试通知短信: 否
- 调度失败重试通知电话: 13910000000
- 调度失败重试通知内容: 任务失败, 请及时检查任务配置。

任务描述 (Job Description):

任务ID: 3024
任务名称: dw_data_ots_app(3024)
任务类型: hive 脚本
首次调度: 2014-11-11 10:00:00
调度周期: 每天
调度时间: 02:00
调度失败重试: 3次
调度失败重试间隔: 10分钟
调度失败重试策略: 立即
调度失败重试通知: 否
调度失败重试通知邮件: 否
调度失败重试通知短信: 否
调度失败重试通知电话: 13910000000
调度失败重试通知内容: 任务失败, 请及时检查任务配置。

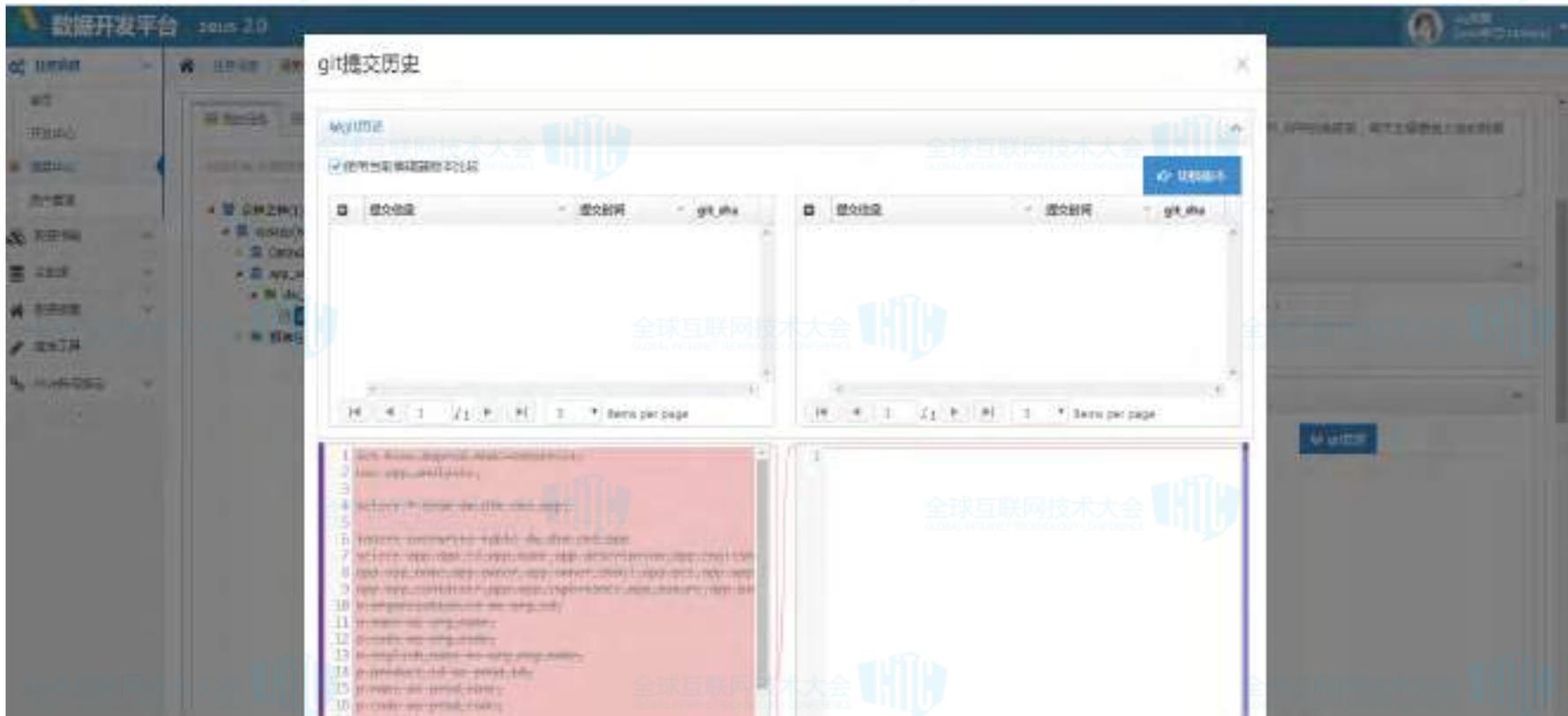
任务配置 (Job Configuration):

调度类型: 定时调度
调度策略: 1
调度表达式: 0 0 5 * * *
调度策略: 默认值

任务脚本 (Job Script):

```
1 set hive.exprs.code=nonstrict;
2 use app_analyst;
3
4 select * from dw_data_ots_app;
5
6 insert overwrite table dw_data_ots_app
7 select app_app_id,app_name,app_description,app_english_name,
8 app_app_name,app_creator,app_owner_email,app_pci,app_app_category,
9 app_app_contact_ser,app_app_importance,app_nature,app_backup_contacts;
```

调度和开发



调度和开发

任务信息

运行日志 8

依赖关系

上游依赖

#	job	id	owner	今日最近运行版本	状态	开始时间	运行时长	查看任务
1	igthotelpagesppageid_d(141783)	141783	igtbi					详情
1.1	h_etl_src_igthotelpage_d(127423)	127423	igtbi	201711010300127423	success	04:47:31	1分钟55秒	详情
1.1.1	igtmvwfactpageview(127222)	127222	igtbi	201711010300127222	success	04:40:21	7分钟9秒	详情
1.1.1.1	factmbpageview(205)	205	bimob	201711010300000205	success	03:54:43	45分钟1秒	详情
1.1.1.1.1	factmbpvlog_hybrid(151)	151	bimob	201711010300000151	success	03:00:00	19分钟29秒	详情
1.1.1.1.2	factmbpvlog_sdk(154)	154	bimob	201711010300000154	success	03:00:00	54分钟39秒	详情

下游依赖

只查询当前owner(opsapp)

#	job	id	owner	今日最近运行版本	状态	开始时间	运行时长	查看任务
1	igthotelpagesppageid_d(141783)	141783	igtbi					详情

数据传输

源库: 目标库:

源表: 目标表:

源数据类型: 目标数据类型:

任务名称	源库名称	目标库名称	源表名称	目标表名称	源数据类型	目标数据类型	操作
任务名称							

源库名称:

目标库名称:

源表名称: 目标表名称:

源数据类型: 目标数据类型:

源表数据类型: 目标表数据类型:

只支持MySQL数据库，其他数据库及其具体材料请参考，请确认query各段中的select字段的数据库和表名与目标表中的数据库和表名一致。

数据传输

数据开发平台 datax2.0

任务列表 (说明: 从datax系统创建的任务, 您只能查看, 不能编辑)

每页显示 10 条记录

jobID	租户	集群	数据源	数据	数据表	目标表	目标库	目标表	创建日期	操作
128058	job	mysql	test	test_job	test	mysql	dataxmysql	test_job_priority_high	2017-09-15	
124568	job	mysql	test	test_job_not_created_job	test	mysql	default	TTTTTTTTTT	2017-09-11	
134568	job	mysql	test	test_job_not_created_job	test	mysql	default	TTTTTT	2017-09-11	
134568	job	mysql	test	test_job_not_created_job	test	mysql	default	TTTTTT	2017-09-11	
128257	job	mysql	@portaguard	hive_account_info	hive	mysql	test	hive_account_info	2017-08-31	
122838	job	mysql	admirenew00	hadooplog	hive	mysql	default	hive	2017-08-30	
122862	job	mysql	test_job	test_job	hive	mysql	default	hive	2017-08-28	
118885	job	mysql	test	datax_admin	hive	mysql	default	hive	2017-08-22	
117728	job	mysql	@portaguard	hive_account_info	hive	mysql	hive	hive_account_info	2017-07-24	
113771	job	mysql	10.25.86.192:1010.0.25.66.3...	hive_object_hierarchy	hive	mysql	hiveobject	hive_object_hierarchy	2017-09-22	

第 1 / 10 页 (共 122 条数据)



连接串管理

支持标准bitan串的登记, 支持用户自定义连接串设置
以保护数据库访问的用户名和密码

Enter

全文搜索

支持基于任务脚本内容的全文关键字搜索

Enter

批量取数

支持同一个关系数据库多张表的hive数据库批量生成datax取数job

Enter

多表联合取数

支持同一个关系数据库多张相同结构的子表联合取数到同一个目标hive表中

Enter

多数据库sharding取数

支持多个关系数据库sharding的相同结构子表的hive数据库批量生成datax取数job

Enter



主数据

The screenshot shows a web-based data management interface. On the left is a navigation sidebar with options like '任意选择', '数据权限', '主数据', '数据变更', '数据工具', '用户与管理', and '日志与审计'. The main area has a search bar with 'idw_app' entered. Below the search bar, there's a table titled 'Results for "idw_app"'. The table has columns for 'Server 名称', 'ID名称', '表名', '表的类型', '表的描述(comment)', '表在数据库中的表空间名', '表的大小', '安全等级', '使用等级', '责任人', '创建时间', and '操作'. Three rows of data are visible, all for 'idw_app' on 'HIVE' server. The first row is for 'idw_app_relation_1', the second for 'idw_app_relation_1_ext_1', and the third for 'idw_app_relation_1_ext_boost1'. At the bottom, there are pagination controls showing '1' of 1 pages.

Server 名称	ID名称	表名	表的类型	表的描述(comment)	表在数据库中的表空间名	表的大小	安全等级	使用等级	责任人	创建时间	操作
HIVE	app_analytic	idw_app_relation_1	内部表	CATIAAPP相关应用表的关联数据		65.8 M	-	-	lywapp	2016-12-15 11:28:21	操作
HIVE	app_analytic	idw_app_relation_1_ext_1	外部表	CATIAAPP相关应用表的关联数据 是idw_app的扩展		202.5 M	-	-	ccwapp	2016-12-16 11:43:42	操作
HIVE	app_analytic	idw_app_relation_1_ext_boost1	外部表	CATIAAPP相关应用表的关联数据 是idw_app的扩展		61.9 M	-	-	lywapp	2016-12-20 17:44:47	操作

主数据

数据开发平台 主数据

表名: app_analyze_dw_app_roleflow_d

描述: 主数据

存储类型: HIVE	Server名称: HIVE	数据库: app_analyze
创建人: opespp	创建时间: -	更新时间: -
数据分区数: 量	大小: 65.0 MB	源类型: 内部表
文件格式: org.apache.hadoop.hive.jdbc.ChRdpFormat	存储路径: [REDACTED]	
北京海融分区保障时间: 永久保障	表结构描述: CATH-APP应用内网关系的层级数据	
创建时间: 2016-12-15 17:29:21	Zou: 周人的修改	操作按钮: <input type="button" value="查看详情"/>

操作按钮:

序号	字段名称	数据类型	字段描述(Comment)
1	appId	string	应用ID
2	appId	string	使用应用ID
3	role_flow	double	应用ID和角色ID的乘积
4	role_flow	double	应用ID和角色ID的乘积
5	role_flow	double	应用ID和角色ID的乘积

主数据

全球互联网技术大会

全球互联网技术大会

全球互联网技术大会

字段信息 18

数据预览

分区信息 817

操作历史

使用热度

表大小趋势

上游血缘

下游血缘

最近7天读取次数

最近7天写入次数

最近1个月读取次数

最近1个月写入次数

最近3个月读取次数

最近3个月写入次数

最近6个月读取次数

最近6个月写入次数

75

7

307

30

1220

88

2197

171

最近7天

最近1个月

最近3个月

最近6个月

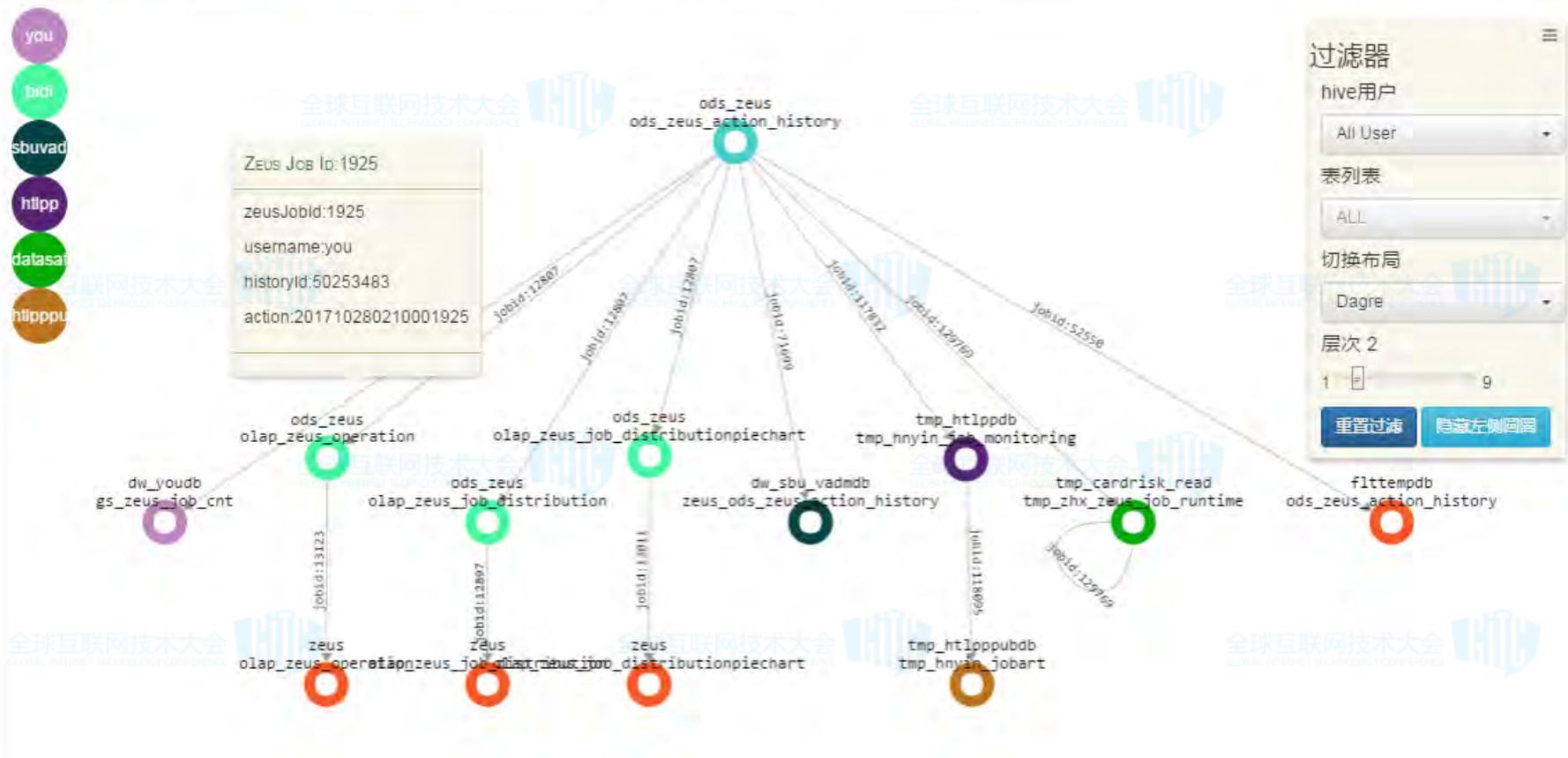
每日使用次数分析



2017-10-28
● 读取次数: 10
● 写入次数: 1

主数据

字段信息 18 数据预览 分区信息 817 操作历史 使用热度 表大小趋势 上游血缘 下游血缘



数据质量

编辑规则

规则名称 规则对象

指标配置

指标序号	类型	名称	计算SQL	描述	操作
1	A1	全局	记录数	count(1)	记录数
2	T1	自定义	前3天的平均数	select round(count(*)/3,0) from dw_ubtdl	前3天的平均数
3	T3	自定义	支持小时分区	select count(1) from :tablename where :pdt>:date-2 and :pthour=:hour-2	支持小时分区

增加 Page 1 of 1 20 View 1 - 3 of 3

计算规则

表达式 表达式

提交

关闭

数据质量



OPS/IT/CC

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

Dataquality

数据质量管理 > 规则管理 > 表校验历史记录

表(dw_ubtdb.factpageview) 手动校验历史

任务类型	表名	规则名称	规则类型	字段名	表达式	校验结果	结束时间	校验执行状态	
1	宙斯	dw_ubtdb.factpageview	访问者的记录数	字段	\	$abs(9433526+8457)>10000$	通过	2017-10-29 16:08:32.0	successed
2	宙斯	dw_ubtdb.factpageview	自定义记录数	表	\	$76415172>50000000$	通过	2017-10-29 09:10:25.0	successed
3	宙斯	dw_ubtdb.factpageview					2017-10-28 16:11:33.0	successed	
4	宙斯	dw_ubtdb.factpageview					2017-10-28 09:14:47.0	successed	
5	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-26'	biubt			2017-10-27 16:05:59	2017-10-27 16:12:07.0	successed
6	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-26'	biubt			2017-10-27 09:07:33	2017-10-27 09:15:53.0	successed
7	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-25'	biubt			2017-10-26 16:05:46	2017-10-26 16:11:50.0	successed
8	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-25'	biubt			2017-10-26 09:05:48	2017-10-26 09:13:09.0	successed
9	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-24'	biubt			2017-10-25 16:05:30	2017-10-25 16:13:08.0	successed
10	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-24'	biubt			2017-10-25 09:05:48	2017-10-25 09:12:48.0	successed
11	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-23'	biubt			2017-10-24 16:05:48	2017-10-24 16:11:32.0	successed
12	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-23'	biubt			2017-10-24 09:05:51	2017-10-24 09:13:02.0	successed
13	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-22'	biubt			2017-10-23 16:06:16	2017-10-23 16:13:09.0	successed
14	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-22'	biubt			2017-10-23 09:05:54	2017-10-23 09:12:57.0	successed
15	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-21'	biubt			2017-10-22 16:05:49	2017-10-22 16:08:28.0	successed
16	宙斯	dw_ubtdb.factpageview	WHERE d = '2017-10-21'	biubt			2017-10-22 09:05:44	2017-10-22 09:11:02.0	successed

表(dw_ubtdb.factpageview)校验通过

规则名称	规则类型	字段名	表达式	校验结果
访问者的记录数	字段	\	$abs(9433526+8457)>10000$	通过
自定义记录数	表	\	$76415172>50000000$	通过

关闭

报表平台 “走马观花”

云报表平台并设置报表

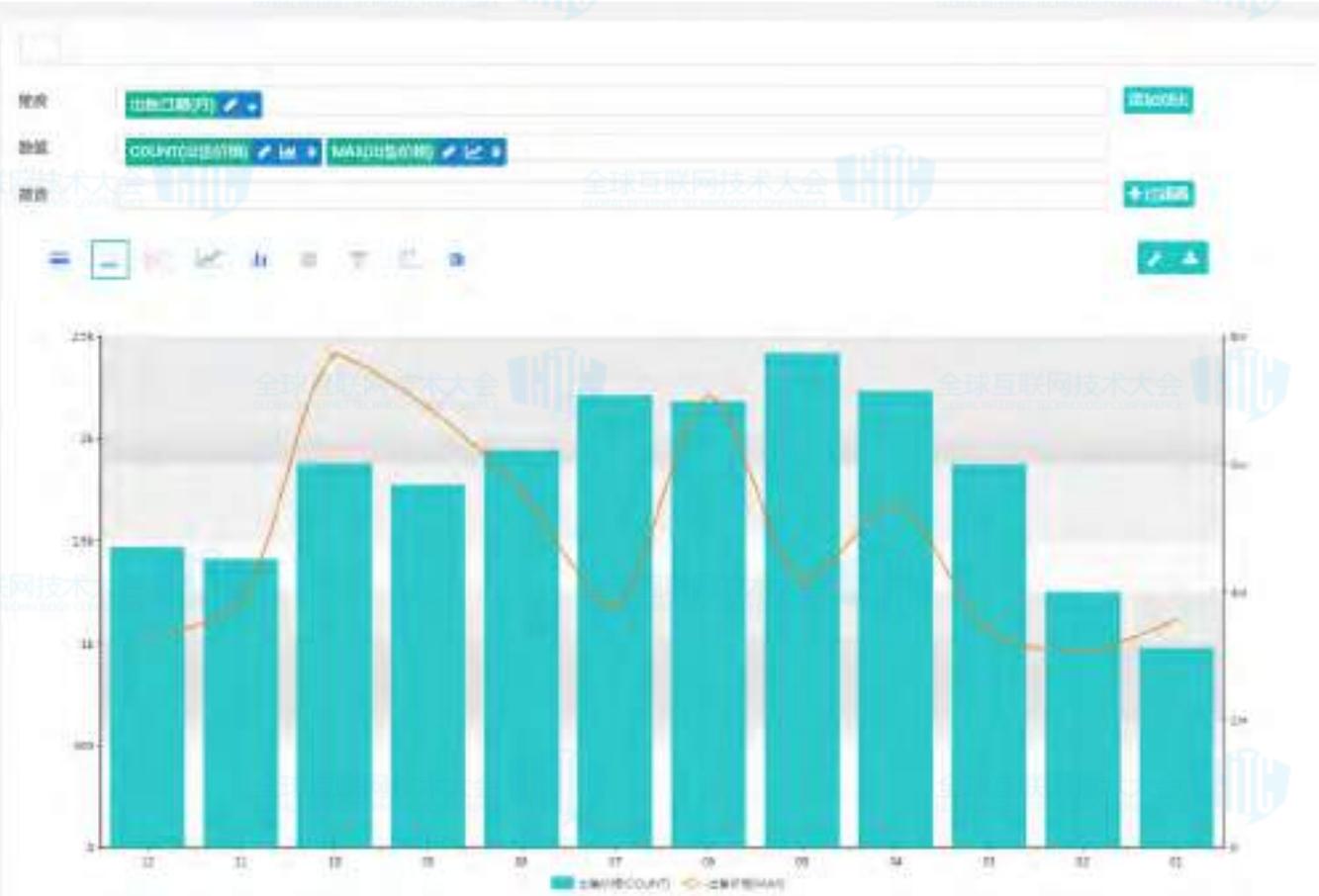
数据源名称: King County/美国华盛顿州

操作按钮: 刷新, 删除, 新增, 分享

报表目录: 自定义目录

- 100 401 机票预订
- 101 机票预订-基础数据
- 102 机票预订-预订数据
- 103 机票预订-预订数据-预订成功
- 104 机票预订-预订数据-预订失败
- 105 机票预订-预订数据-预订取消
- 106 机票预订-预订数据-预订变更
- 107 机票预订-预订数据-预订退款
- 108 机票预订-预订数据-预订其他
- 109 机票预订-预订数据-预订其他-其他
- 110 机票预订-预订数据-预订其他-其他
- 111 机票预订-预订数据-预订其他-其他
- 112 机票预订-预订数据-预订其他-其他
- 113 机票预订-预订数据-预订其他-其他
- 114 机票预订-预订数据-预订其他-其他
- 115 机票预订-预订数据-预订其他-其他
- 116 机票预订-预订数据-预订其他-其他
- 117 机票预订-预订数据-预订其他-其他
- 118 机票预订-预订数据-预订其他-其他
- 119 机票预订-预订数据-预订其他-其他
- 120 机票预订-预订数据-预订其他-其他

报表名称: 机票预订



报表平台 “走马观花”

该界面展示了报表平台的导航菜单。左侧为树状目录，包含以下主要分类：

- 2017Q3数据展示
 - 运营分布
 - 运营分布
 - 运营分布
 - 运营分布
 - 运营分布
 - 运营分布
 - 运营分布
- 运营运营
- 网站运营中心
 - 网站运营
 - Top20国家 酒店预订统计
 - Top20国家 酒店预订统计
 - Top20国家 酒店预订统计(酒店、品牌)
 - 北京酒店预订
 - 2. ART系统数据
 - 3. Zentao系统数据
 - 4. Hadoop相关数据
 - 5. Job运行时长统计 (最近一周)
 - 7. Job运行时长统计 (最近一周平均)
 - 1. 数据仓库数据
 - 3. Hadoop任务运行概况
 - 5. Hadoop任务运行概况
 - hadoop_jobw01

Hadoop监控Dashboard

该仪表盘展示了Hadoop系统的运行指标。上方图表显示了应用运行概况，下方图表显示了Hadoop任务运行概况。右侧弹出了一个任务列表窗口。

3. Hadoop应用运行概况

该折线图显示了应用运行概况，横轴为时间（从09-23到11-01），纵轴为数值（从0到300）。图中包含两条主要数据系列，分别代表不同的应用运行指标。

4. Hadoop任务运行概况

该折线图显示了Hadoop任务运行概况，横轴为时间（从09/11-09/17到10/28-10/28），纵轴为数值（从0到100）。图中包含多条数据系列，分别代表不同的任务运行指标。

提交 pending@mapred-31-07 日有一个失败	8
提交 pending@mapred-31-07 日有一个失败	9
提交 pending@mapred-31-07 日有一个失败	10
提交 pending@mapred-31-07 日有一个失败	11
提交 pending@mapred-31-07 日有一个失败	12
提交 pending@mapred-31-07 日有一个失败	14
提交 pending@mapred-31-07 日有一个失败	16
提交 pending@mapred-31-07 日有一个失败	24
提交 pending@mapred-31-07 日有一个失败	26

机器学习平台 - 基于Spark的图形化AI平台

The screenshot displays the Ctrip ML interface with a workflow for movie recommendation. The workflow consists of the following steps:

- 读取电影打分数据 (Read movie rating data)
- 执行训练 (Execute training)
- 训练测试分离: 5 (Train and test split: 5)
- ALS (ALS)
- 测试数据替换 (Replace test data)
- RMSE评估 (RMSE evaluation)

The interface includes a top navigation bar with the Ctrip ML logo, a breadcrumb trail (训练名: 电影推荐-zy | HiveID: MJ_JB), and a toolbar with various icons. A right-hand sidebar contains a search bar and a list of filters. The bottom section provides configuration details for the Spark job, including a 'Spark:' section with 'edit' and 'cancel' buttons, and a 'Tip:' section.

机器学习平台 - 基于Spark的图形化AI平台

The interface displays a workflow for training and evaluating an ALS model. The steps are:

- 读取电影打分数据 (Load movie rating data)
- 执行优化 (Execute optimization)
- 训练测试分块: 5 (Train/evaluate in chunks of 5)
- 测试数据转换 (Convert test data)
- RMSE评估 (RMSE evaluation)

The ALS component is central to the process, receiving input from the training step and outputting to the conversion step.

Configuration parameters for ALS:

- Alpha: 1
- Checkpoint Interval: 10
- implicit Pref:
- Max Iteration: 10
- Nonnegative:
- Num Item Blocks: 10
- Num User Block: 10

On the right side, there is a sidebar with a search bar and a list of components:

- 推荐模块 (Recommendation Module)
- Type to search
- ALS
- Error/Guest
- Count Vertices
- Class
- Decision Tree
- PMML Columns
- RMSE
- Broadcast
- Convert To N-Grams
- Custom Read CF
- DCI
- DFSH
- Filter Rows
- GBT Classifier

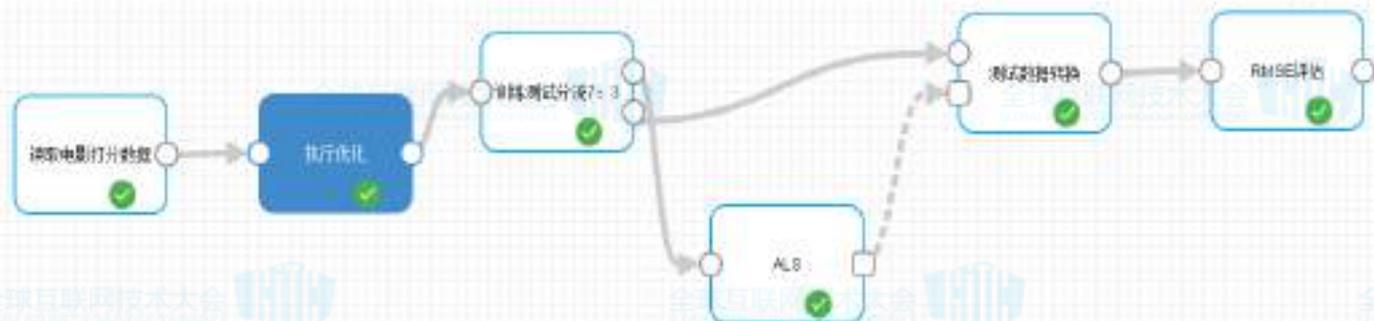
机器学习平台 - 基于Spark的图形化AI平台

The screenshot displays the Ctrip ML machine learning platform interface. At the top, the logo 'Ctrip ML' and the text '训练名: 电影推荐-zy' are visible. The main area features a workflow diagram with nodes: '读取电影打分数据', '执行优化', '训练测试分选', 'ALS', '测试数据转换', and 'RMSE评估'. The '训练测试分选' node is highlighted in blue with a green checkmark. Below the diagram, the '训练测试分选' configuration panel is shown, with the '单步执行' button highlighted in red. The configuration includes '训练数据比例' (0.7), '测试数据比例' (0.3), and '随机种子' (1). The '执行信息' panel on the right shows execution details: '开始时间: 2016-11-21 15:35:23', '结束时间: 2016-11-21 15:35:24', and '执行耗时: 1 seconds'. The '执行输出' section contains the following text:

```
customTransformedDF_01: org.apache.spark.sql.DataFrame = [userid: int, movieid: int, rating: double, timestamp: int]
splits: Array[org.apache.spark.sql.DataFrame] = Array([userid: int, movieid: int, rating: double, timestamp: int], [userid: int, movieid: int, rating: double, timestamp: int])
trainData_01: org.apache.spark.sql.DataFrame = [userid: int, movieid: int, rating: double, timestamp: int]
testData_01: org.apache.spark.sql.DataFrame = [userid: int, movieid: int, rating: double, timestamp: int]
```

机器学习平台 - 基于Spark的图形化AI平台

评测名: 电影推荐-xy



执行代码

基本信息

参数信息

执行信息

定制输出

▶ 单击执行 ▶ 双击执行 ▶ 鼠标执行 ▶ 高级操作 ▶

开始时间: 2016-11-21 15:34:01

结束时间: 2016-11-21 15:35:21

执行耗时: 77 seconds

执行输出

```
1 def custom_transform(inputDF: DataFrame): DataFrame = {
2   // custom code ...
3
4
5   val scf = inputDF.repartition(4)
6   scf.collect()
7   scf
8 }
```

```
import org.apache.spark.sql.DataFrame
custom_transform(inputDF: org.apache.spark.sql.DataFrame)org.apache.spark.sql.DataFrame
input_csv_data_01: org.apache.spark.sql.DataFrame = [userid: int, movieid: int, rating: double, timestamp: int]
customTransformedDF_01: org.apache.spark.sql.DataFrame = [userid: int, movieid: int, rating: double, timestamp: int]
```

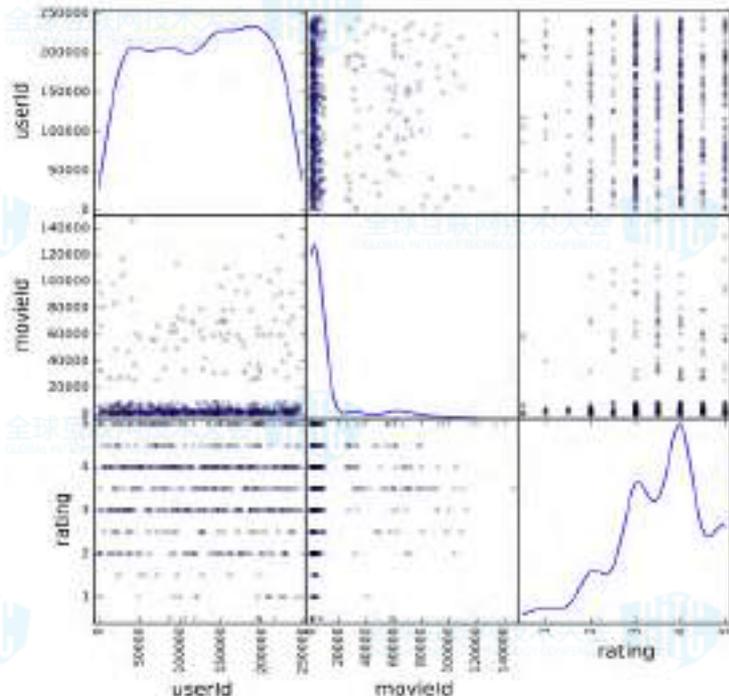
机器学习平台 - 基于Spark的图形化AI平台

直接注释

图形 数据

执行/取消

```
1 %spark
2 %python 数据探索主要是对spark dataFrame的数据collect到driver端,
3 %所以进行计算,工具可以使用python的pandas, spark可以将采集到的数据
4
5 import matplotlib
6 import pandas as pd
7 matplotlib.use('Agg')
8
9 import numpy as np
10 import matplotlib.pyplot as plt
11 import matplotlib.pyplot as plt
12 import StringIO
13 from pandas.tools.plotting import scatter_matrix
14
15 #这是必须的,否则图表不显示
16 def showDF():
17     img = StringIO.StringIO()
18     p.savefig(img, format='svg')
19     img.seek(0)
20     print "html <div> - img.buf + "</div>"
21     plt.close()
22
23 #从控制台可获取DataFrame
24 defName = sqlContext.table(tableName)
25 customTransformedDF_01 = sqlContext.table("customTransformedDF_01").select("userId","movieId","rating")
26
27 #print customTransformedDF_01.count()
28
29 #matplotlib = customTransformedDF_01.sample(4112, 0.00001, True)
30
31 #spark DataFrame转换为Pandas DataFrame, 然后通过pandas api进行科学统计或可视化
32 pdDF = sampleData.toPandas()
33 #print pdDF
34 # python 计算与画图code
35 #pdDF["rating"].hist.bins(1)
36 plt.figure()
37 #pdDF.hist(bins=100, color='k', alpha=0.5, bins=50)
38 scatter_matrix(pdDF, s=10, figsize=(7, 7), diagonal='hist')
39
40 showDF()
```



机器学习平台 - 基于Spark的图形化AI平台

基础信息

模块名称

模块描述

模块归属 共享级别

代码模板

```
1 import org.apache.spark.ml.recommendation.ALS
2 val $inputDFS = sqlContext.table("$inputDFS")
3 val als = new ALS()
4   .setAlpha($alpha$)
5   .setCheckpointInterval($checkpointInterval$)
6   .setImplicitPrefs($implicitPrefs$)
7   .setMaxIter($maxIter$)
8   .setNonnegative($nonnegative$)
9   .setNumItemBlocks($numItemBlocks$)
10  .setNumUserBlocks($numUserBlocks$)
11  .setRank($rank$)
12  .setRegParam($regParam$)
13  .setSeed($seed$)
14  .setRatingCol("$ratingCol$")
15  .setItemCol("$itemCol$")
16  .setUserCol("$userCol$")
17  .setPredictionCol("$predictionCol$")
18 val $outputMD$ = als.fit($inputDFS)
```

配置信息

Key	Key显示名	描述信息	参数类型	默认参数值	可选项	参数归属
inputDF	<input type="text" value="输入训练数据"/>	<input type="text"/>	<input type="text" value="STRING"/>	<input type="text"/>	<input type="text" value="格式: v1,v2.."/>	<input type="text" value="输入!"/>
alpha	<input type="text" value="Alpha"/>	<input type="text"/>	<input type="text" value="NUMBER"/>	<input type="text" value="1"/>	<input type="text" value="格式: v1,v2.."/>	<input type="text" value="模块!"/>
checkpointInterval	<input type="text" value="Checkpoint Interva"/>	<input type="text"/>	<input type="text" value="NUMBER"/>	<input type="text" value="10"/>	<input type="text" value="格式: v1,v2.."/>	<input type="text" value="模块!"/>
implicitPref	<input type="text" value="Implicit Pref"/>	<input type="text"/>	<input type="text" value="BOOLEAN"/>	<input type="text" value="false"/>	<input type="text" value="格式: v1,v2.."/>	<input type="text" value="模块!"/>

机器学习平台 - GPU的云平台(开发中)

The screenshot displays the Ctrip DI Service web interface. The top navigation bar includes the Ctrip logo, the text 'Ctrip DI Service', and a user profile icon for 'MuChang'. A left sidebar contains navigation options: '资源地址', '平台服务列表', and '我的服务实例'. The main content area, titled '平台服务列表', shows two Jupyter notebook instances. Each instance has a 'Name' and a 'Desc' field. The first instance is named 'jupyter with tensorflow cpu' and its description is '通过jupyter进行tensorflow深度学习开发, 支持 python/c/ c++, 计算资源: Gpu node: 4个CPU核'. The second instance is named 'jupyter with tensorflow gpu' and its description is '通过jupyter进行tensorflow深度学习开发, 支持 python/c/ c++, 计算资源: Gpu'. The interface also features a pagination control at the bottom of the list, showing '1' of 10 items.

机器学习平台 - 基于Spark的图形化AI平台





挑战和展望



目前最大的挑战：通过平台推动公司的知识和数据的共享
不能只建设数据工具！

实例1：主数据平台

- 目前只是表的元数据及其衍生的信息
- 希望能够建成数据知识分享的平台，包括数仓架构，指标体系等更多信息

实例2：AI平台

- 不仅仅是提供模型训练工具，需要的是全链路的解决方案
- 能够覆盖 数据获取，特征抽取，模型实验，模型训练，线上服务 等AI应用开发的各个生命周期的需求

目前是大数据技术正经历着一个新旧更替

- MR -> Spark
- Hive -> SparkSQL
- Storm -> Flink（建议新项目）

AI平台不像大数据平台，有比较成熟的底层平台和解决方案

- 更好用的工具助力“AI民主化”
- 落地各个生命周期的工具，逐步整合成全链路的解决方案

系统不应该是仅仅只包含通用的平台和工具，应该更加贴近数据业务，为用户提供一站式服务



Q & A

Thanks 😊

