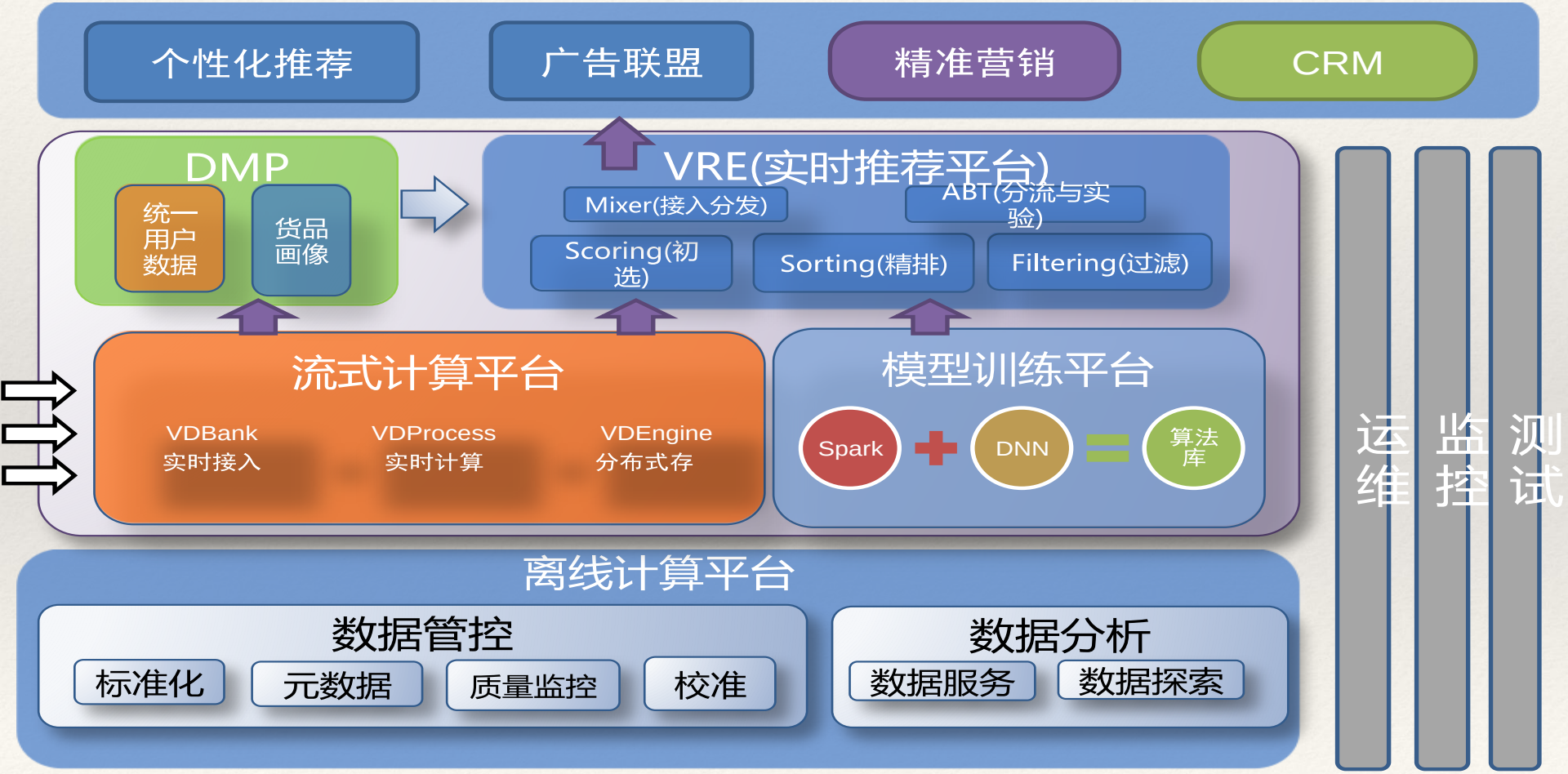
唯品会大数据平台

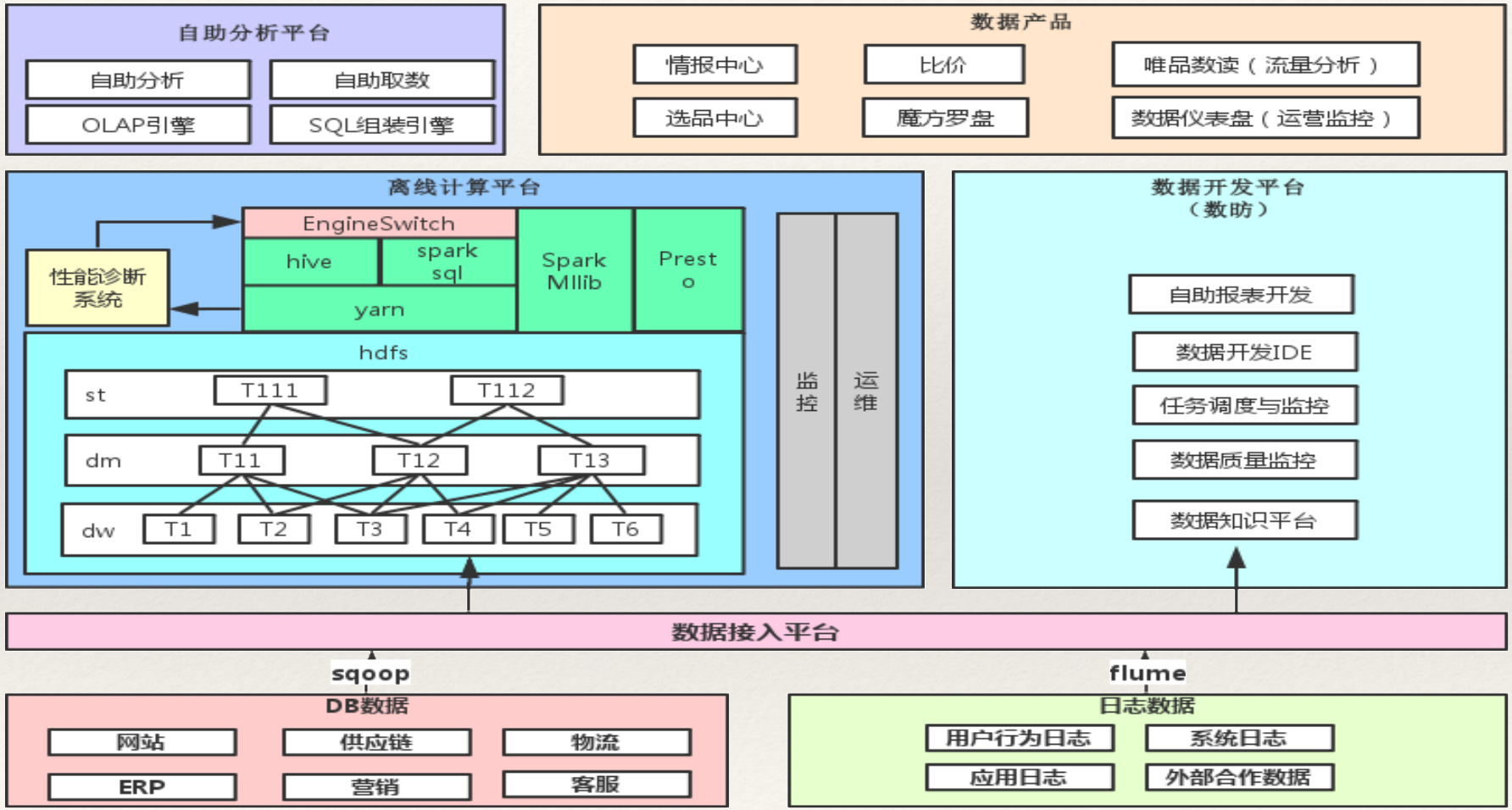
大数据存储和计算资源管理

邮箱:
eric.shan@vipshop.com
微信: shanchaoeric

唯品会大数据平台规划


个性化推荐
广告联盟
精准营销
CRM
DMP
统一用户数据
货品画像
VRE(实时推荐平台)
Mixer(接入分发)
ABT(分流与实验)
Scoring(初选)
Sorting(精排)
Filtering(过滤)
流式计算平台
VDBank
实时接入
VDProcess
实时计算
VDEngine
分布式存
模型训练平台
Spark
DNN
算法库
运维
监控
测试
离线计算平台
数据管控
标准化
元数据
质量监控
校准
数据分析
数据服务
数据探索

唯品会大数据平台现状


自助分析平台
自助分析
自助取数
OLAP引擎
SQL组装引擎
数据产品
情报中心
比价
唯品数读（流量分析）
选品中心
魔方罗盘
数据仪表盘（运营监控）
离线计算平台
EngineSwitch
hive
spark sql
yarn
Spark Mllib
Prest o
性能诊断系统
hdfs
st
T111
T112
dm
T11
T12
T13
dw
T1
T2
T3
T4
T5
T6
监控
运维
数据开发平台（数昉）
自助报表开发
数据开发IDE
任务调度与监控
数据质量监控
数据知识平台
数据接入平台
sqoop
flume
DB数据
网站
供应链
物流
ERP
营销
客服
日志数据
用户行为日志
系统日志
应用日志
外部合作数据

大数据管理工作范畴

- 业务系统
- 调度系统
- ETL
- 数据模型
- 元数据/主数据管理
- 数据质量
- 开发流程
- 运维流程
- 数据审计和安全
- 资源管理

“数据平台使用申请”

- 用户提交：
- 资源类型
- hdfs存储/hive数据库/hive计算资源/mr计算资源…
- 资源数目
- 100T存储/1T内存/1000颗CPU…
- 访问方式
- hive/presto/spark/webhdfs…

- 管理员处理：
- hdfs分配：
- path/name quota/space quota
- hive分配： 数据库/授权
- yarn分配：
- 队列最小资源/最大资源/weight

理想很丰满，现实很骨感

理想

- 系统强大
- 数据规范
- 流程规范
- 技术成熟
- 业务成熟

现实

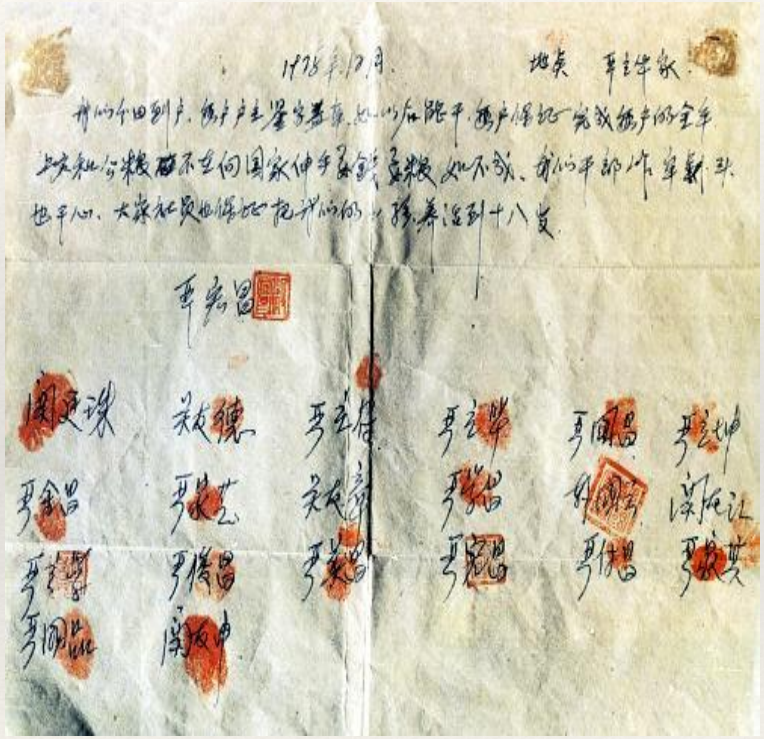
- 模型变更迅速，开发周期短
- 用户能力参差不齐
- 大量的历史包袱
- 大量的技术包袱
- 平台不稳定，掌控力差
- 分层不明确

各种问题

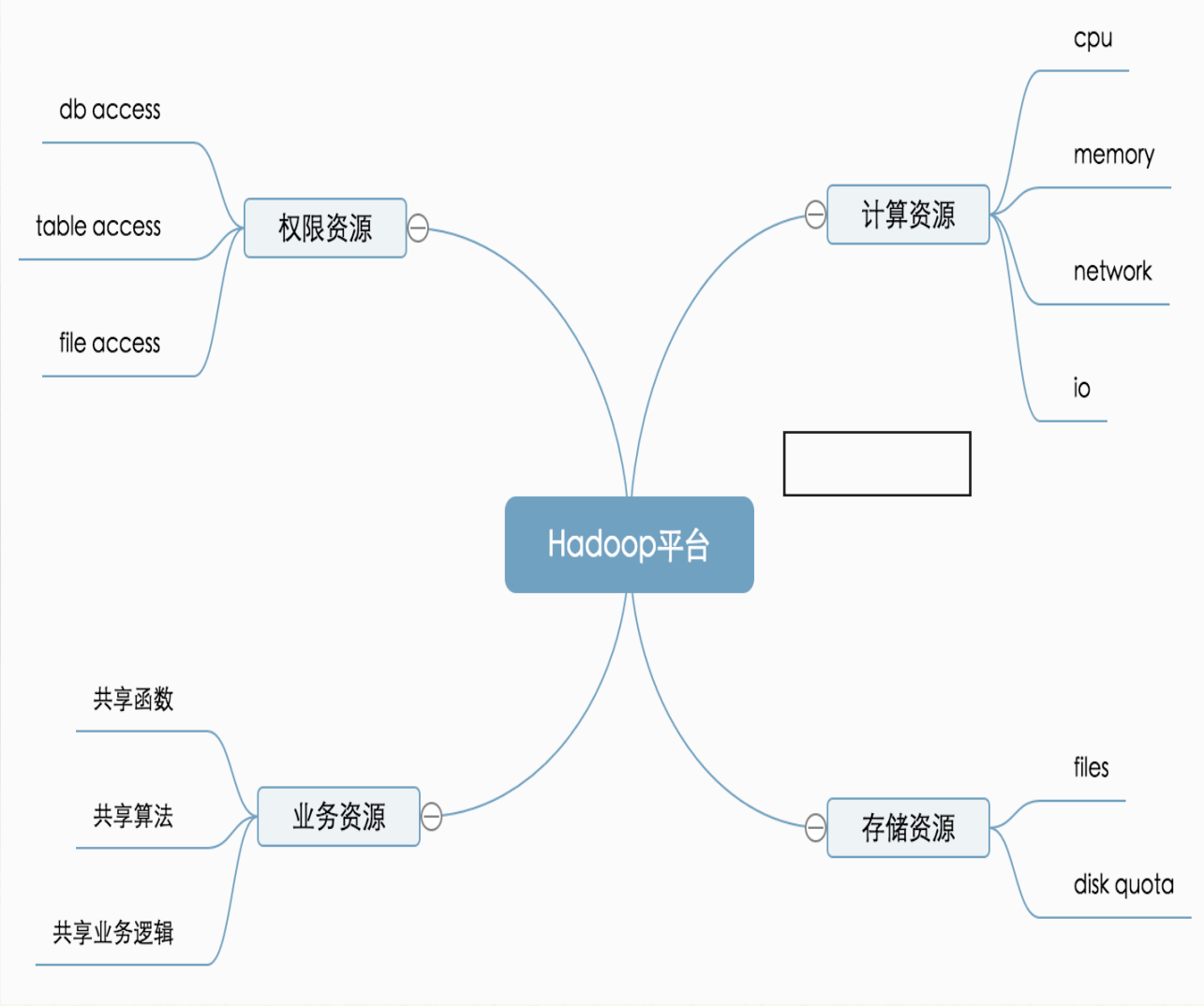
- 这个任务昨天还好好的，为什么今天跑不出来了？
- 2-10倍的数据量，能撑得住吗？
- 怎么几千个任务都慢了？
- 最近磁盘使用急剧增加，谁在用？
- 这个表好像不用了，我能删除掉吗？
- 集群要扩容吗？扩多少？

核心：资源管控

- 分田到户
- 目的：
  - 从乱序到有序
  - 申请和分配有据可查
  - 规则公开透明
  - 数据公开透明
  - 有多少资源，干多少事
  - 合理的KPI和惩罚机制
  - ROI，资源倾斜给回报率高的项目



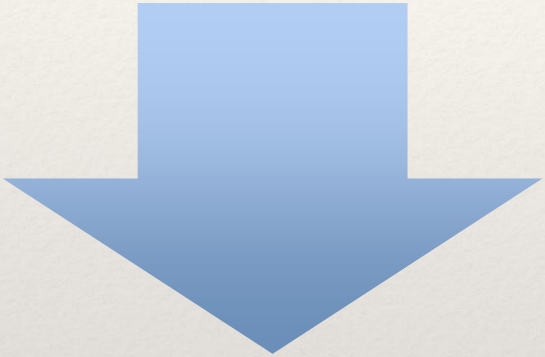
资源有什么？


Hadoop平台
权限资源
db access
table access
file access
计算资源
cpu
memory
network
io
业务资源
共享函数
共享算法
共享业务逻辑
存储资源
files
disk quota

为什么存储和计算需要关注？

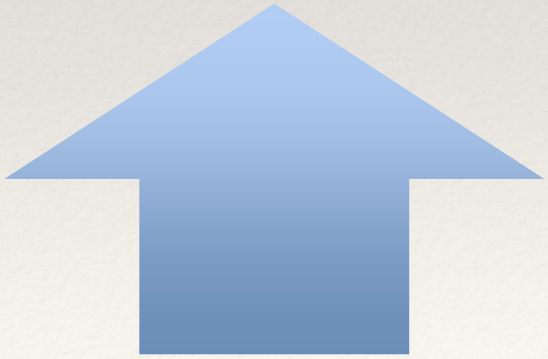
- Scale Up > Scale Out
- Namenode - 存储(2亿blocks/2亿files)
  - standby namenode updateCountForQuota缓慢影响主从一致性，进而影响切换（HDFS-6763）
  - standby checkpoint缓慢导致增量blockreport汇报被skip，影响主从一致性，进而影响切换（HDFS-7097）
  - standby checkpoint GC导致transfer Fsimage超时失败
  - 集群启动期间，blockreport需要错开，导致启动缓慢，namenode压力增加
- ResourceManager - 计算(1k+并行job/40w+ job每天)
  - 大量任务运行期间，resource manager分配能力不足
    - https://issues.apache.org/jira/browse/YARN-3547 部分解决问题
    - https://issues.apache.org/jira/browse/YARN-5188 our patch for fairscheduler
  - 队列分配过粗，互相影响严重

开源节流



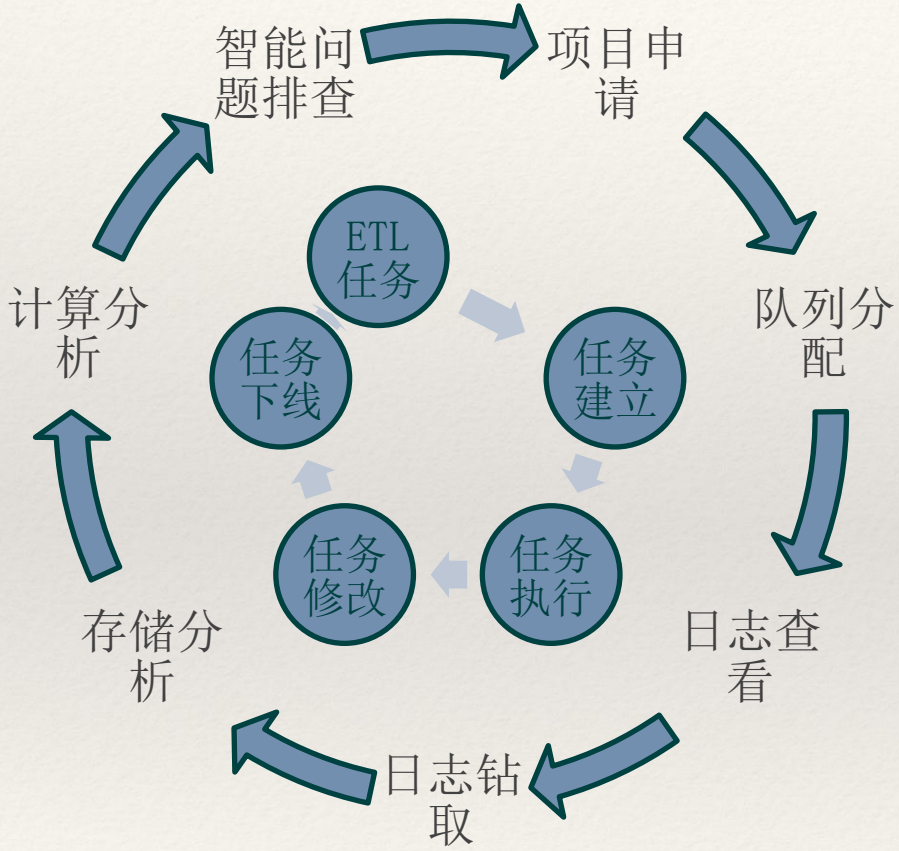
Federation
存储优化管理
计算优化管理

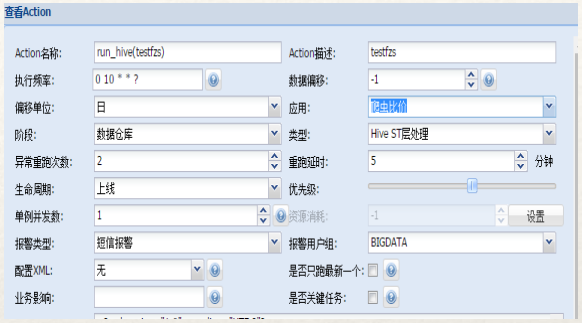
提升namenode rpc性能
提升yarn的containaer assign性能
增加机器

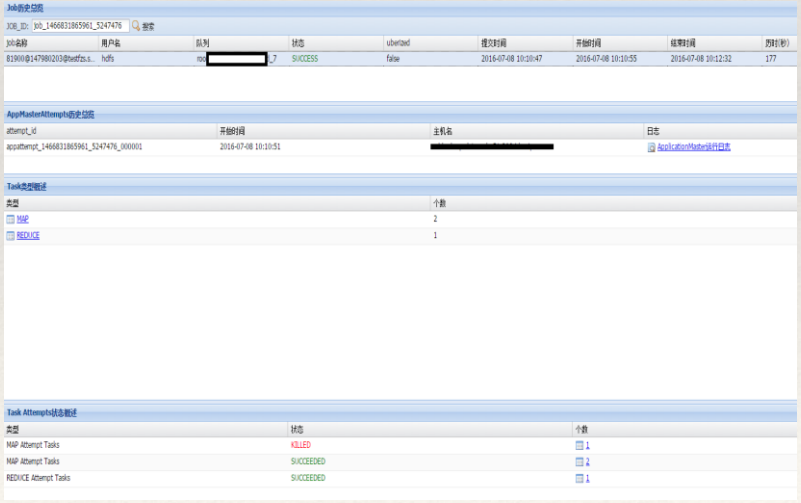


管理数据化成果

数据仓库-Hadoop闭环







仓库使用数据化


Demeter
资源使用TOP10
资源增长率TOP10
新增任务TOP10
应用MR应用趋势图
新增任务趋势图

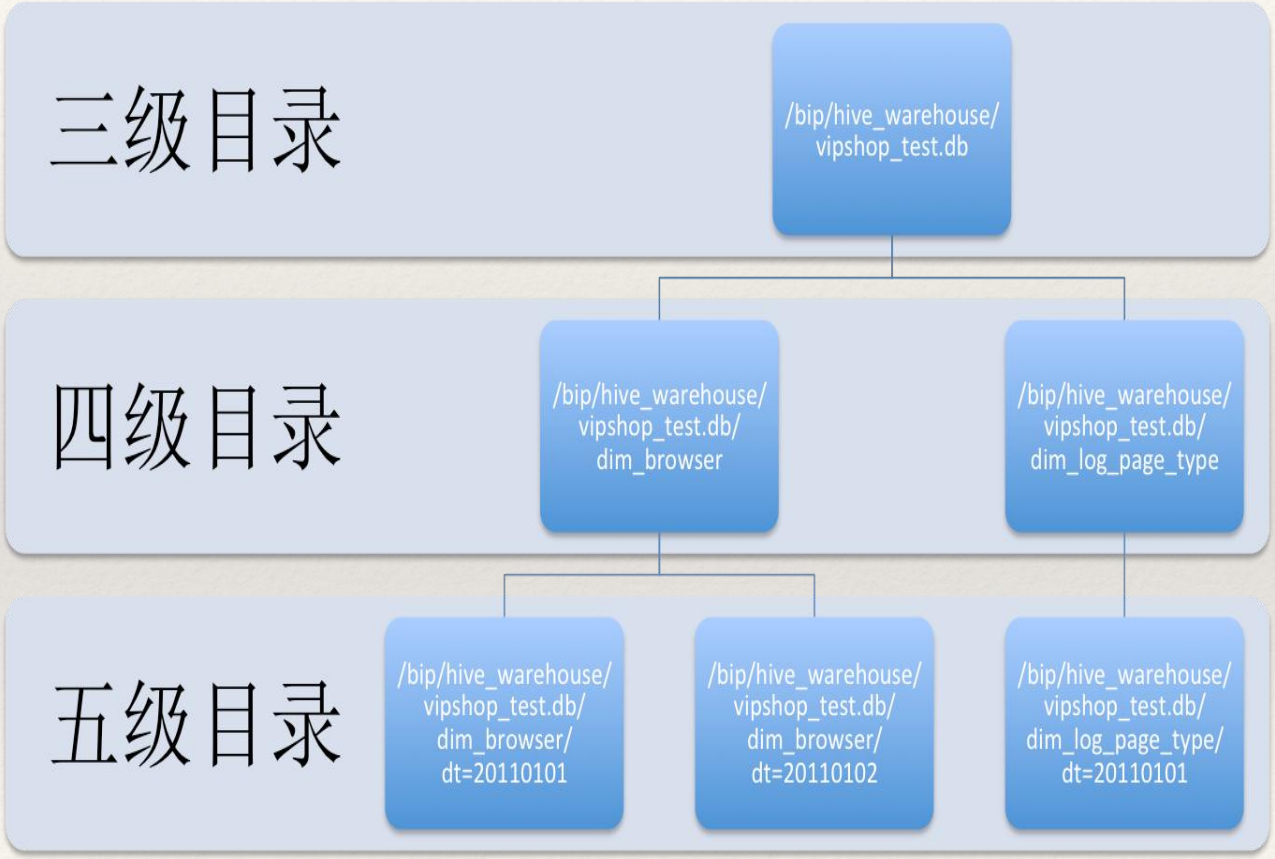

Demeter
资源使用 (TOP10)
资源使用增长率 (TOP10)
新增表
资源使用趋势图
新增表趋势


Demeter
表数量变化趋势
表存储变化趋势
说明:
表数量：目前在hive中存在的表的数量总和。
说明:
表存储变化趋势：是指在对应的日期内hive中所有表占用的存储空间总和。

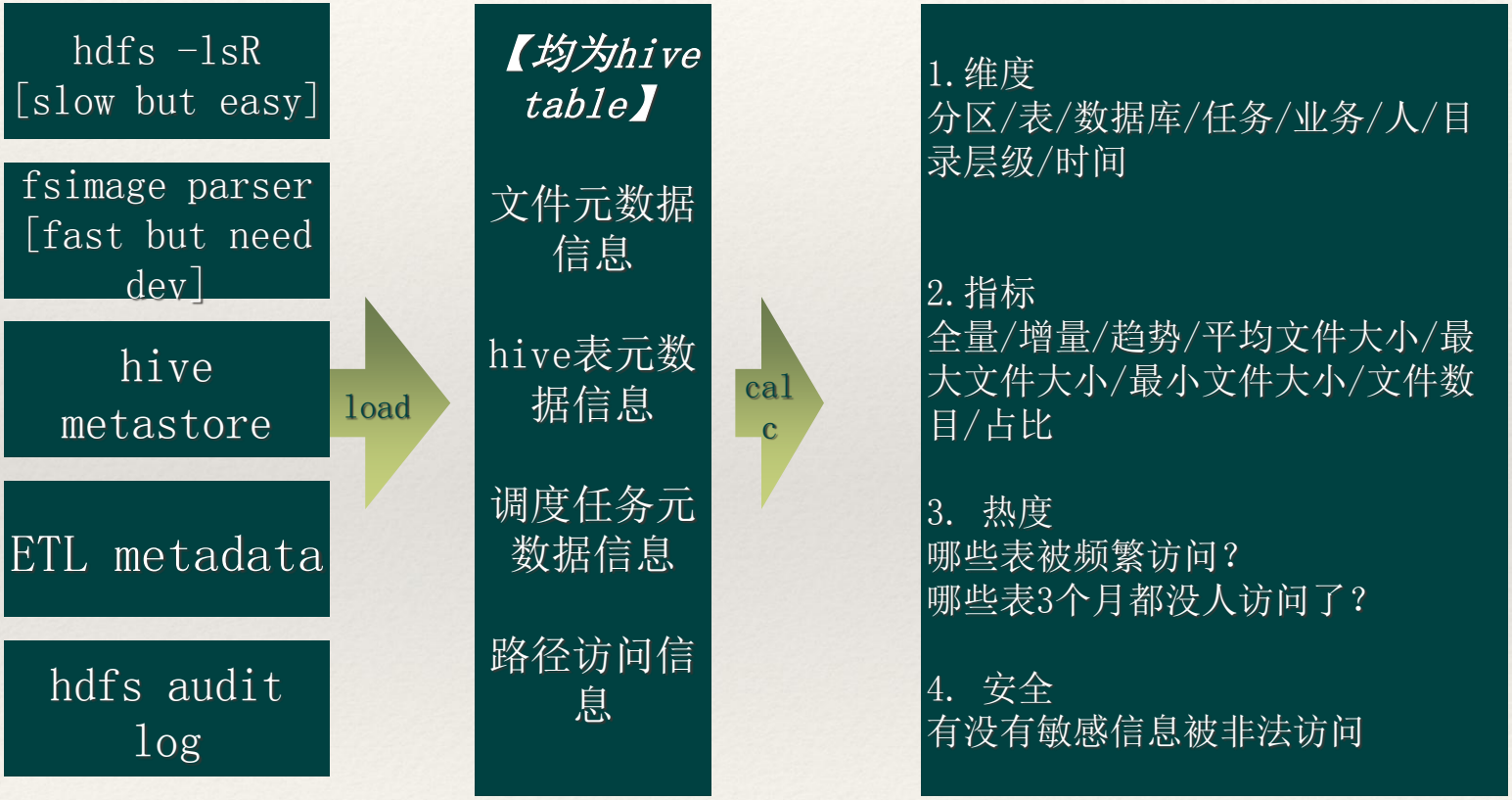

Demeter
数据库使用量 (TOP10)
数据库增长率 (TOP10)
数据库表数量 (TOP10)
数据库存储使用趋势
数据库表数量趋势

存储资源管理

存储资源管理
- hdfs存储资源



存储资源管理
– 如何获取存储数据



存储资源管理
- 如何使用存储数据

- 容量计费
  - 通过计费来控制资源
  - 存储数据完整透明
  - 消费预警，提前知会用户
- 空间管理
  - 自动配置生命周期管理规则
  - 存储格式，压缩格式选择（orc+gzip）
- 文件管理
  - 自动配置生命周期管理规则
  - 小文件har归档

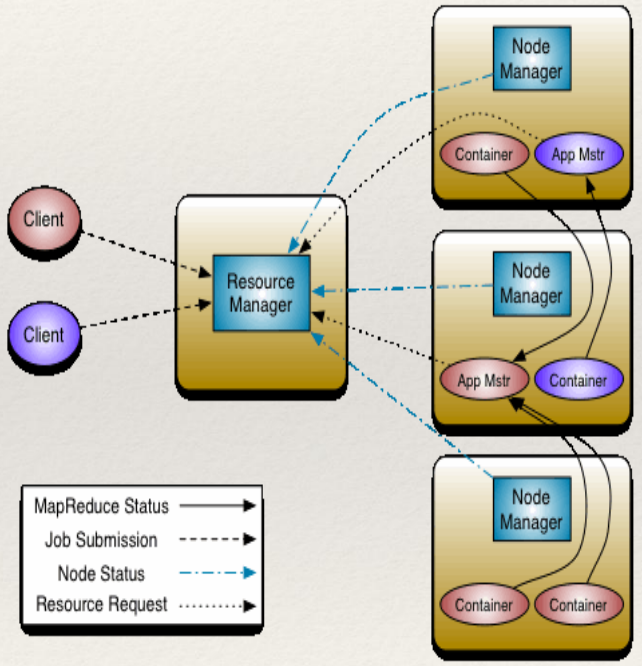
存储资源管理
– 控制存储的价值

- 解决NN“单点”瓶颈
- 控制服务器数量，降低成本
- 规范数据生命周期管理
- 统计冷热数据使用，反馈给ETL生命周期管理

计算资源管理

计算资源管理
yarn - 统一调度管理

- yarn，好像搞定了资源管理，我们还需要管理什么？



计算资源管理
- beyond yarn

- 队列管理，共享还是独享？队列分到多细合适？如何确保关键队列的资源？
- 每个队列的使用情况如何？
- 这个部门的新同事总是写错sql，占用大量资源，怎么办？
- 晚上3点多A队列资源紧张，在干什么？
- B任务，最近消耗资源情况怎么样？
- B任务，C sql，为什么step1的application突然跑慢了？
- 今天最消耗资源的application是哪个？能优化吗？
- 有没有数据倾斜造成的任务延迟？我们要解决一下
- 这么多机器，分配的任务数均衡吗？
- 有没有一些机器任务失败率特别高？

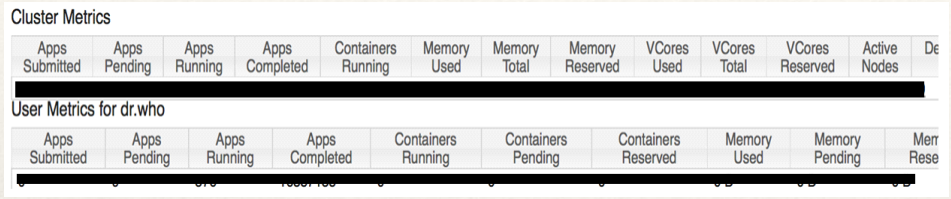
计算资源管理
– 如何使用计算资源

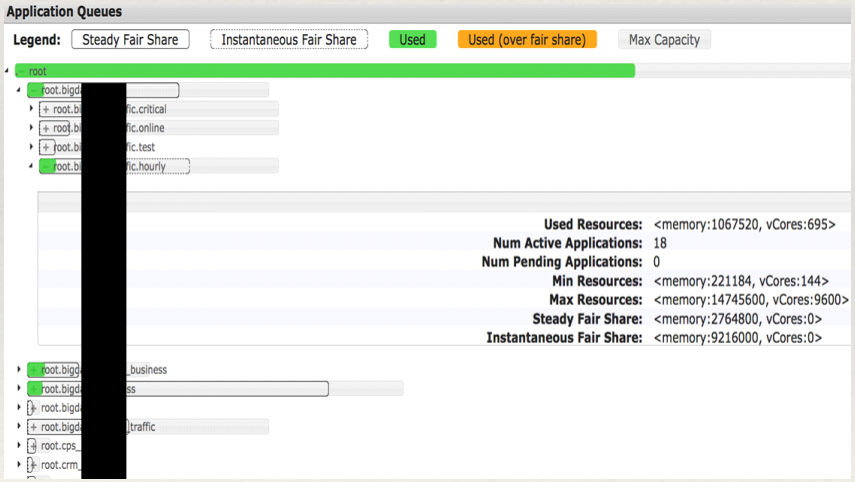
- 容量计费
  - 通过计费来控制资源
  - 存储数据完整透明
  - 消费预警，提前知会用户
- 实时告警和自动处理
  - 根据队列设置不同的规则，如运行时长，使用资源，自动发现和触发停止动作
  - 通过业务注码，自动展示运行中的业务细节
- 数据倾斜自动识别
- 队列数据化运营

计算资源管理
- 公平调度

- 我们的管理原则：
  - 尽量细化，单个业务分配单独队列
  - 队列分配的min/max/weight由实际业务来评估，上线初期会不断调整
  - min是保证的最小资源，确保优先获得
  - max是业务的最大资源限制，确保不会超过
  - 每个队列由多个不同级别的子队列组成，子队列业务可灵活调整
  - 子队列大小可以基于时间动态调整
    - 白天，天任务队列缩小，小时任务队列放大
    - 夜晚，天任务队列放大，小时任务队列缩小
    - 关键任务确保队列内的最小队列保证

计算资源管理
- Yarn实时运行情况监控





- 优点
  - 数据完全实时
- 缺点
  - 展现不够直观
  - 无历史时序数据

计算资源管理（秒级）
– 数据获取

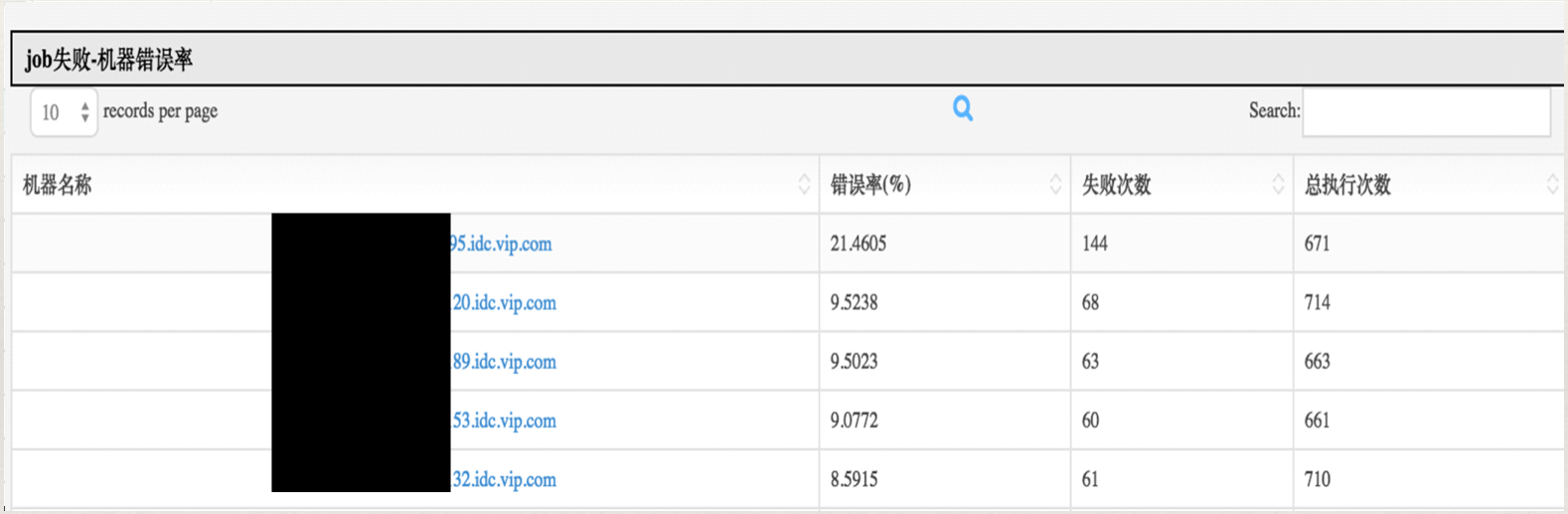
- historylog
  - 通过实时计算框架，获取每个application的明细执行结果
  - 缺点：任务完成后才能获取到完整信息
- job api
  - 通过api实时获取到所有job的基础信息
  - 比默认rm的api提供更多字段信息，如sql信息
  - 缺点：不是100%完整的数据，定期获取必然会丢失数据

计算资源管理（秒级）
- 用户查询识别示例

资源使用(运行中资源/总资源)	query	提交时间	map进展(已完成的map/总map)	reduceNeeded	reduce进展(已完成的reduce/总re	用户名
0.01% (2/16000)	SELECT [redacted] olution, c	2016-04-21 20:16:35	100.00% (1619/1619)	yes	100.00% (959/961)	sam.a
0.31% (50/16000)	---店[redacted] 楼层 select	2016-04-21 20:23:19	84.00% (59/70)	yes	0.00% (0/39)	yanli
0.29% (47/16000)	create[redacted] _jrd.temp_	2016-04-21 20:29:45	0.00% (0/47)	yes	0.00% (0/27)	huich
0.28% (45/16000)	select[redacted] evel_new ,	2016-04-21 20:28:23	2.00% (1/46)	yes	0.00% (0/25)	cindy
0.28% (45/16000)	select[redacted] evel_new ,	2016-04-21 20:28:08	2.00% (1/46)	yes	0.00% (0/25)	cindy
0.22% (36/16000)	create[redacted] _bigdata.g	2016-04-21 20:30:10	0.00% (0/36)	yes	0.00% (0/21)	anita
0.21% (33/16000)	create[redacted] _bigdata.w	2016-04-21 20:28:52	0.00% (0/33)	yes	0.00% (0/18)	charl
0.09% (14/16000)	select[redacted] name,a.riq	2016-04-21 20:29:05	100.00% (24/24)	yes	0.00% (0/14)	huich
0.00% (0/16000)	create[redacted] _bigdata.w	2016-04-21 20:26:49	100.00% (2/2)	yes	100.00% (1/1)	charl
0.01% (1/16000)	select[redacted] act_fav_pe	2016-04-21 20:28:51	100.00% (2/2)	yes	0.00% (0/1)	borui
0.01% (1/16000)	select[redacted] rom vipods.	2016-04-21 20:29:59	0.00% (0/1)	no		yanva

Thu Apr 21 18:48:01 CST 2016 jobname=--xxx.chen-qid:152011-...100(Stage-2) user=xxx.chen job_id=job_1459656116710_7806076 starttime=1461232053 exceed 3600 seconds,killing...

计算资源管理（秒级）
-实时监控task kill ratio



job失败-机器错误率

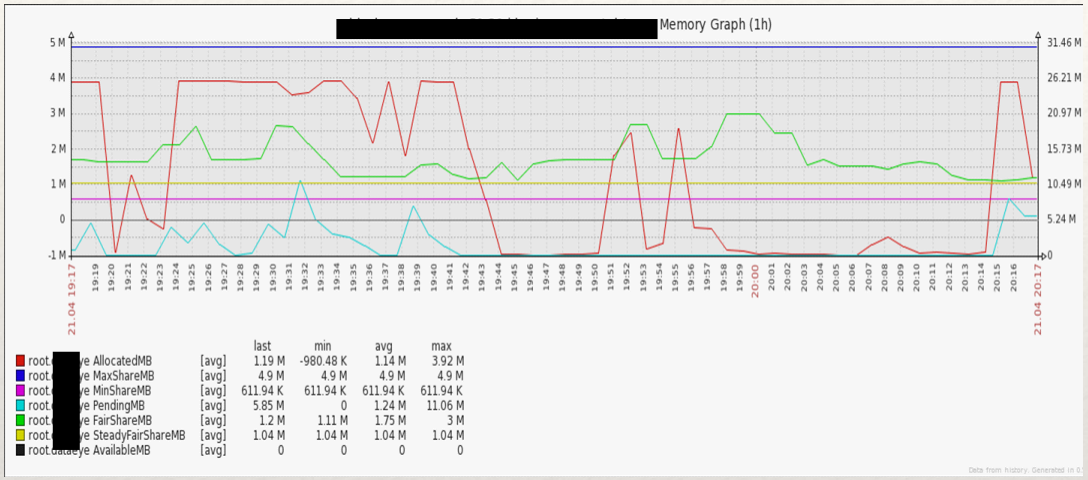
10 records per page　　Search:

机器名称	错误率(%)	失败次数	总执行次数
[redacted]95.idc.vip.com	21.4605	144	671
[redacted]20.idc.vip.com	9.5238	68	714
[redacted]89.idc.vip.com	9.5023	63	663
[redacted]53.idc.vip.com	9.0772	60	661
[redacted]32.idc.vip.com	8.5915	61	710

计算资源管理（分钟级）
- jmx数据来补充

- jmx: "http://%s:8088/jmx" % (IP)
- 返回格式:
- #"name" : "Hadoop:service=ResourceManager,name=QueueMetrics,q0=root,q1=mapreduce,q2=xxx,q3=panda",
- #"modelerType" : "QueueMetrics,q0=root,q1=mapreduce,q2=xxx,q3=panda",
- #"tag.Queue" : "root.mapreduce.xxx.panda",
- #"tag.Context" : "yarn",
- #"tag.Hostname" : "xxxx",
- #"running_0" : 0,
- #"running_60" : 0,
- #"running_300" : 0,
- #"running_1440" : 0,
- #"FairShareMB" : 0,
- #"FairShareVCores" : 0,
- #"SteadyFairShareMB" : 1228800,
- #"SteadyFairShareVCores" : 0,

计算资源管理（分钟级）
- 单个队列监控实例



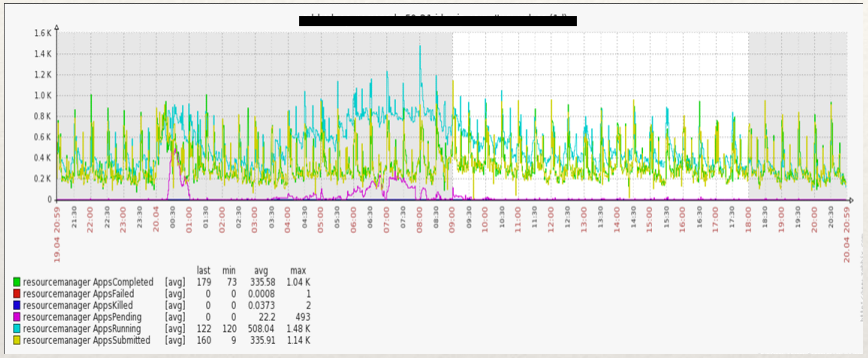
队列分配红线跑平

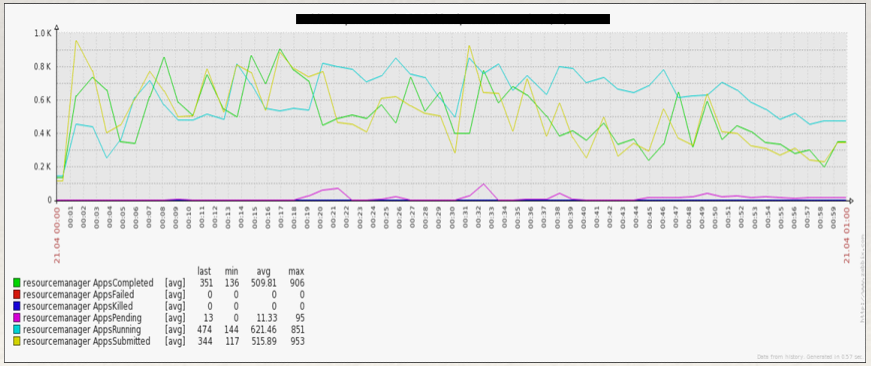
队列等待蓝线升高

-结论，单个业务资源吃紧

-需要增加最大可分配资源

计算资源管理（分钟级）
- resourcemanager metric监控示例

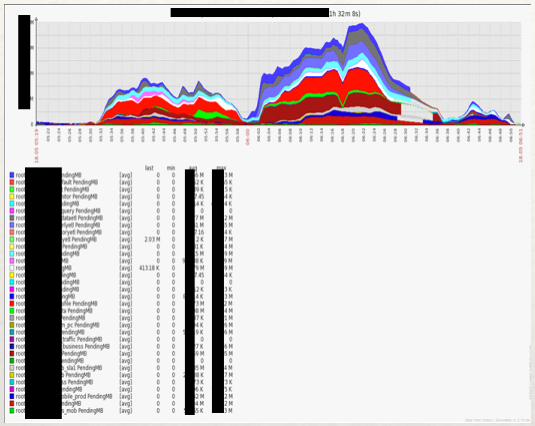


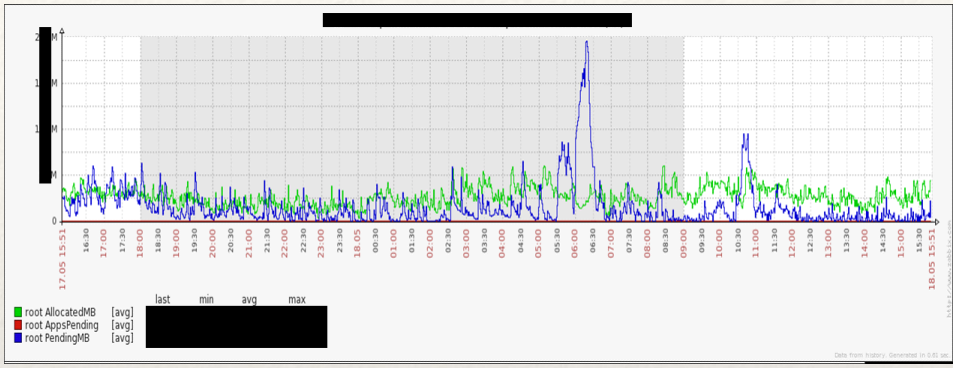


调整前：
高峰期app pending增加
凌晨任务1个小时任务延迟

调整min后：
最大pending不超过100
pending很快下降

计算资源管理（分钟级）
- resourcemanager metric监控示例



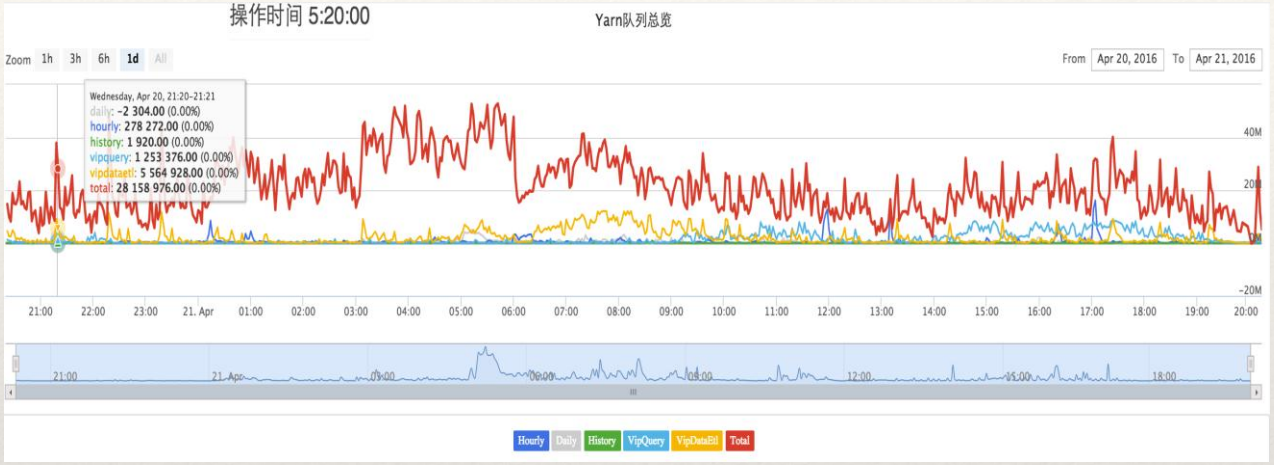


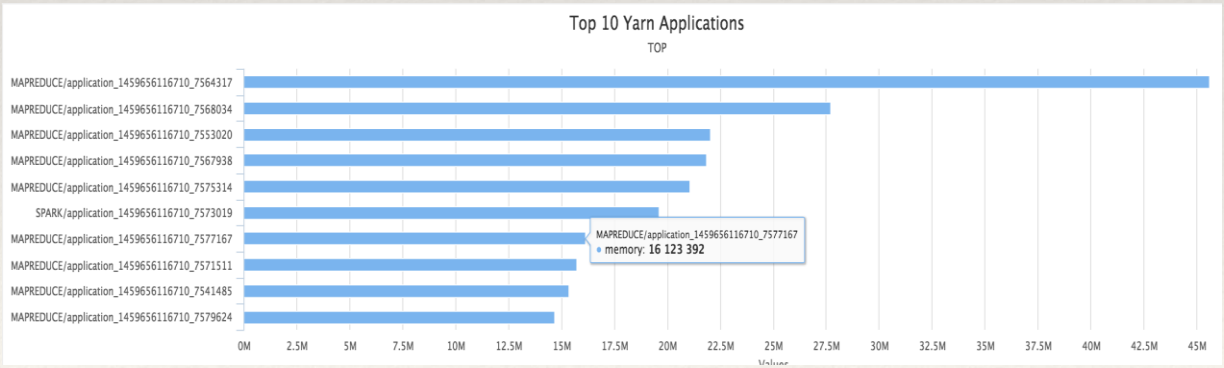
高峰期资源需求增加，但是分配能力下降
yarn分配能力受到影响，将问题加剧

计算资源管理（分钟级）
– 优化展现

- 集群总体资源分布情况
- 最消耗资源的是什么任务
- 实时/历史的数据查看

计算资源管理（分钟级）
- 队列总览展现





计算资源管理（分钟级）
– 队列总览展现

20:18:52

Name	AllocatedMem
application_1459656116710_7706807 Kill	4524 GB
application_1459656116710_7706796 Kill	1502 GB
application_1459656116710_7706558 Kill	1232 GB
application_1459656116710_7706789 Kill	795 GB
application_1459656116710_7706735 Kill	752 GB
application_1459656116710_7706932 Kill	691.5 GB
application_1459656116710_7707079 Kill	618 GB
application_1459656116710_7706706 Kill	612 GB
application_1459656116710_7706596 Kill	612 GB
application_1459656116710_7706563 Kill	612 GB

20:17:52

Name	AllocatedMem
application_1459656116710_7706802 Kill	4417.5 GB
application_1459656116710_7706558 Kill	2087 GB
application_1459656116710_7706796 Kill	1614.5 GB
application_1459656116710_7706401 Kill	1503 GB
application_1459656116710_7706208 Kill	1179 GB
application_1459656116710_7706806 Kill	1078.5 GB
application_1459656116710_7706605 Kill	1056 GB
application_1459656116710_7706562 Kill	1022 GB
application_1459656116710_7706697 Kill	849 GB
application_1459656116710_7705867 Kill	819.5 GB

20:16:52

Name	AllocatedMem
application_1459656116710_7706401 Kill	1951.5 GB
application_1459656116710_7706208 Kill	1270.5 GB
application_1459656116710_7705867 Kill	962 GB
application_1459656116710_7706458 Kill	618 GB
application_1459656116710_7706132 Kill	478.5 GB
application_1459656116710_7706057 Kill	355.5 GB
application_1459656116710_7705797 Kill	334.5 GB
application_1459656116710_7706540 Kill	325.5 GB
application_1459656116710_7706063 Kill	297 GB
application_1459656116710_7706535 Kill	268.5 GB

计算资源管理（天级）
- 离线资源使用

- 查询集群的资源使用场景
- 时间/应用/队列维度的资源使用情况
- 核心ETL任务近期map/reduce使用情况
- 单个attempt的metrics指标查看，如读取超过1kw行数据的map任务
- 等等

计算资源管理（天级）- 数据倾斜识别示例

job_id	reduce数量	最大的reduce记录	平均的reduce数量	执行时间（分钟）
job_1459656116710_6658192	222	2822637983	15337298.81	102
job_1459656116710_6664132	23	37513841	2546765.435	96
job_1459656116710_6647201	365	2008016597	12100798.89	92
job_1459656116710_6649269	195	1299715244	40103824.76	92
job_1459656116710_6678373	222	357911334	3676790.941	87
job_1459656116710_6644657	65	127639703	2927121.677	77
job_1459656116710_6655736	35	179517393	5129068.371	75
job_1459656116710_6649261	196	1173068814	33226867.51	69
job_1459656116710_6672118	56	72236233	3390145.625	69
job_1459656116710_6586793	999	1053934083	16646491.87	68
job_1459656116710_6648192	517	881368212	12514695.39	62
job_1459656116710_6743462	222	357911334	3676790.941	61
job_1459656116710_6586097	41	2061710884	50285631.32	58

```
2016-04-17 00:27:46,128 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 2667.83 sec
2016-04-17 00:28:46,615 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 2734.54 sec
2016-04-17 00:29:47,118 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 2801.9 sec
2016-04-17 00:30:47,598 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 2864.16 sec
2016-04-17 00:31:48,065 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 2923.87 sec
2016-04-17 00:32:48,407 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 2988.7 sec
2016-04-17 00:33:48,757 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3051.42 sec
2016-04-17 00:34:49,170 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3116.62 sec
2016-04-17 00:35:49,513 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3182.49 sec
2016-04-17 00:36:49,923 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3246.46 sec
2016-04-17 00:37:50,298 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3308.14 sec
2016-04-17 00:38:50,648 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3369.24 sec
2016-04-17 00:39:50,960 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3435.01 sec
2016-04-17 00:40:51,313 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3500.37 sec
2016-04-17 00:41:51,656 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3563.65 sec
2016-04-17 00:42:51,975 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3628.23 sec
2016-04-17 00:43:52,294 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3694.49 sec
2016-04-17 00:44:52,624 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3756.79 sec
2016-04-17 00:45:52,951 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3822.47 sec
2016-04-17 00:46:53,240 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3891.12 sec
2016-04-17 00:47:53,559 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 3953.68 sec
2016-04-17 00:48:53,823 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 4019.17 sec
2016-04-17 00:49:54,099 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 4088.16 sec
2016-04-17 00:50:54,420 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 4150.72 sec
2016-04-17 00:51:54,771 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 4220.34 sec
2016-04-17 00:52:55,037 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 4280.66 sec
2016-04-17 00:53:55,270 Stage-1 map = 100%, reduce = 97%, Cumulative CPU 4346.13 sec
```

TaskAttempts列表

Attempt ID	type	status	任务状态	日志	开始时间	结束时间	历时(秒)
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:43	2016-04-17 01:02:05	7462
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:04	2016-04-17 00:27:42	38
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:04	2016-04-17 00:27:42	38
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:04	2016-04-17 00:27:42	38
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:04	2016-04-17 00:27:42	38
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:04	2016-04-17 00:27:42	38
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:04	2016-04-17 00:27:42	38
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:04	2016-04-17 00:27:42	38
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:04	2016-04-17 00:27:42	38
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:04	2016-04-17 00:27:42	38
attempt_1459656116710...	REDUCE	SUCCEEDED	reduce > reduce	TaskAttempt运行日志	2016-04-17 00:27:04	2016-04-17 00:27:42	38

计算资源管理
-计算资源优化实例

- 用更少的资源计算
  - orcfile，压缩率更高，列式存储降低资源消耗
  - 权衡资源和性能，基于record而不是size调整reduce数量
  - 基于hll的uv估算函数，提供可增量的uv计算

计算资源管理
-计算资源优化实例

- 用更多的资源计算，更快的释放
  - sparksql，内存需求高，复杂计算快
  - presto/impala，利用mpp框架提高计算性能

计算资源管理
-计算资源优化实例

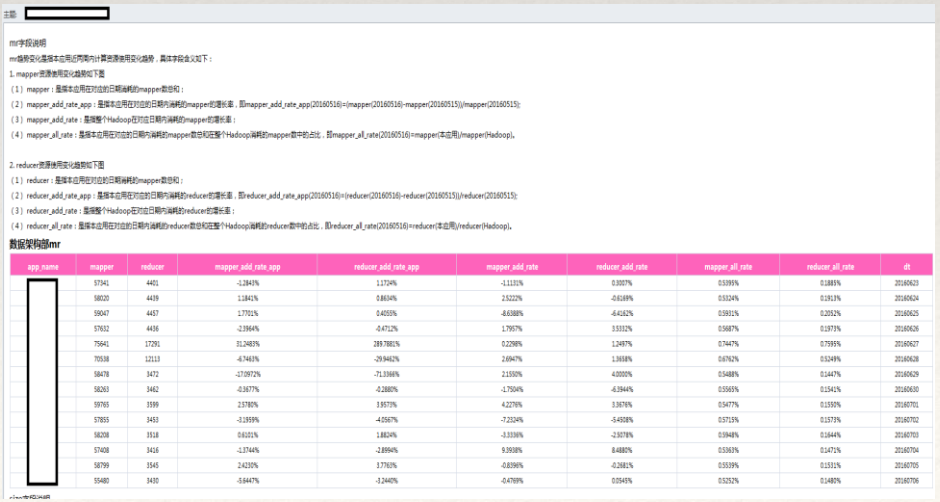
- 不同队列的资源使用上限限制
  - 基于项目粒度的队列资源把控，个性化控制最大可提交资源
  - 项目队列的最大值限制，避免单个项目失控

计算资源管理
-计算资源优化实例

- Reduce Slow Start
- 基于实际m/r的比值设置该参数
- 资源更快的获得和释放，整体集群受益

资源使用报告

- 1年：1T=3000 RMB, 1 CPU/1.5G Mem = 800 RMB
- 部门磁盘使用费用
  - 磁盘使用是没有时间范围的概念，最大值计算
- 部门CPU/Mem使用费用
  - cpu/mem使用要考虑task执行时间，面积计算



mr字段说明

mr趋势变化是指本应用近两周内计算资源使用变化趋势，具体字段含义如下：

1. mapper资源使用变化趋势如下图

（1）mapper：是指本应用在对应的日期消耗的mapper数总和；

（2）mapper_add_rate_app：是指本应用在对应的日期内消耗的mapper的增长率，即(mapper_add_rate_app(20160516)=(mapper(20160516)-mapper(20160515))/mapper(20160515);

（3）mapper_add_rate：是指整个Hadoop在对应日期内消耗的mapper的增长率；

（4）mapper_all_rate：是指本应用在对应的日期内消耗的mapper数总和在整个Hadoop消耗的mapper数中的占比，即mapper_all_rate(20160516)=mapper(本应用)/mapper(Hadoop)。

2. reducer资源使用变化趋势如下图

（1）reducer：是指本应用在对应的日期消耗的mapper数总和；

（2）reducer_add_rate_app：是指本应用在对应的日期内消耗的reducer的增长率，即reducer_add_rate_app(20160516)=(reducer(20160516)-reducer(20160515))/reducer(20160515);

（3）reducer_add_rate：是指整个Hadoop在对应日期内消耗的reducer的增长率；

（4）reducer_all_rate：是指本应用在对应的日期内消耗的reducer数总和在整个Hadoop消耗的reducer数中的占比，即reducer_all_rate(20160516)=reducer(本应用)/reducer(Hadoop)。

数据架构部mr

app_name	mapper	reducer	mapper_add_rate_app	reducer_add_rate_app	mapper_add_rate	reducer_add_rate	mapper_all_rate	reducer_all_rate	dt
[redacted]	57341	4401	-1.2843%	1.1724%	-1.1131%	0.3007%	0.5395%	0.1885%	20160623
	58020	4439	1.1841%	0.8634%	2.5222%	-0.6189%	0.5324%	0.1913%	20160624
	59047	4457	1.7701%	0.4059%	-8.6388%	-6.4162%	0.5931%	0.2052%	20160625
	57632	4436	-2.3964%	-0.4712%	1.7957%	3.5312%	0.5687%	0.1973%	20160626
	75641	17291	312.483%	289.7881%	0.2298%	1.2497%	0.7447%	0.7595%	20160627
	70538	12113	-6.7463%	-29.9462%	2.6947%	1.3658%	0.6762%	0.5249%	20160628
	58478	3472	-17.0972%	-71.3366%	2.1550%	4.0000%	0.5488%	0.1447%	20160629
	58263	3462	-0.3677%	-0.2880%	-1.7504%	-6.3944%	0.5565%	0.1541%	20160630
	59765	3599	2.5780%	3.9573%	4.2276%	3.3676%	0.5477%	0.1550%	20160701
	57855	3453	-3.1959%	-4.0567%	-7.2324%	-5.4508%	0.5715%	0.1573%	20160702
	58208	3518	0.6101%	1.8824%	-3.3336%	-2.3078%	0.5948%	0.1644%	20160703
	57408	3416	-1.3744%	-2.8964%	9.3938%	8.4880%	0.5363%	0.1472%	20160704
	58799	3545	2.4230%	3.7763%	-0.8396%	-0.2681%	0.5539%	0.1531%	20160705
	55480	3430	-5.6447%	-3.2440%	-0.4789%	0.0545%	0.5252%	0.1480%	20160706

唯品会
vip.com
一家专门做特卖的网站

Thanks !

www.vip.com