DTCC

2016中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2016

数 据 定 义 未 来

SequeMedia 盛拓传媒　ITPUB.com　ChinaUnix.net　ITPUB

携程实时大数据平台实践分享

@张翼

公司简介

携程旅游网成立于1999年，总部设在上海，目前有员工30000余人

2003年12月9日在美国纳斯达克成功上市

携程拥有超过2.5亿的注册会员

酒店预订：在全球200多个国家和地区拥有超过120万家酒店的会员酒店

机票预订：产品覆盖全球六大洲5000多大中城市

旅游度假：线路产品覆盖超过100多个目的地国家和地区；2015年大陆地区度假产品的服务人次超过2000万

我的介绍

浙江大学本科，硕士毕业

9年工作经验，5年大数据架构的经验

之前在eBay中国研发中心和大众点评工作过，从0开始组件团队，搭建起大众点评数据平台的基础架构

目前是携程的大数据平台负责人

关注大数据架构领域的发展，对Hadoop，HIVE，HBASE，Spark，Storm等有所研究，致力于大数据架构和业务场景的结合和落地，通过数据产生业务价值

目录

缘起

携程数据业务的特点：

- 业务部门多，形态差别大：酒店 / 机票两大BU，超过15个SBU和公共部门
- 业务复杂，变化快

之前，各个业务部门也有一些实时数据应用，但存在着诸多问题：

- 技术上五花八门
- 力量薄弱，应用的稳定性无法保证
- 缺少周边的配套设施
- 数据和信息共享不顺畅

平台需求

稳定可靠的平台：业务只需要关心业务逻辑的实现，平台维护交给专业同学

完整的配套设施：测试环境，上线，监控，告警

信息共享：数据共享，应用场景共享，互相启发

及时的服务：解决从开发，上线，维护整个过程中遇到的问题

目录

技术选型

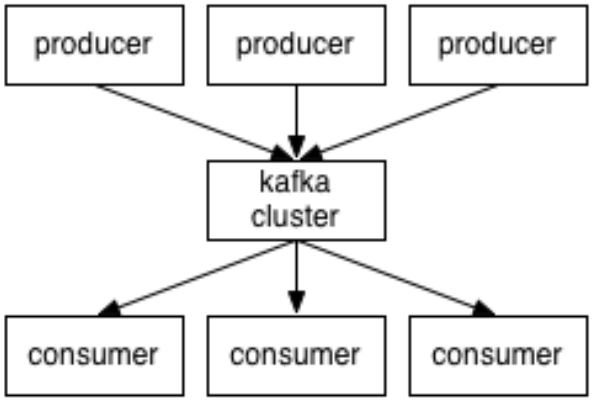
消息队列：

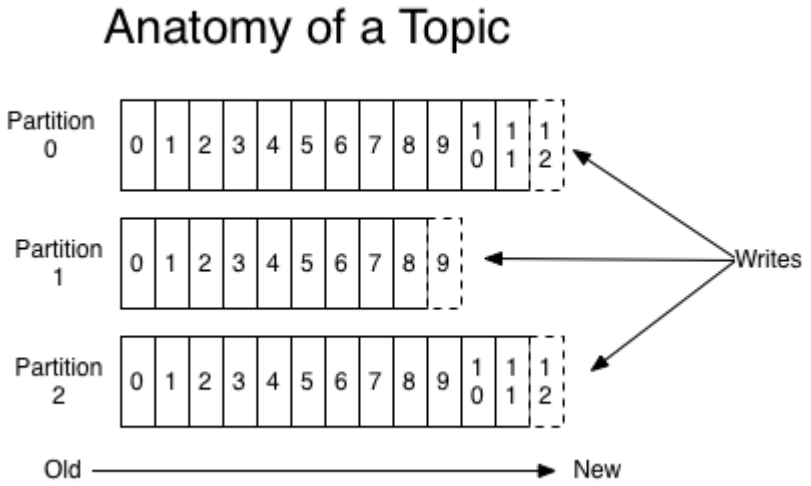
实时处理平台：

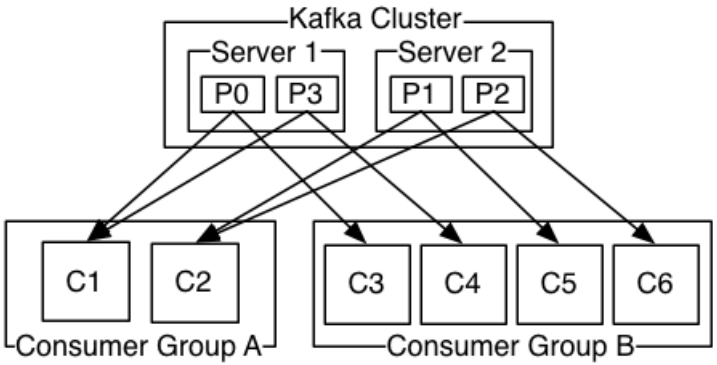
PK

主要出于稳定性的考虑，我们最后选择Storm作为数据处理的平台

Kafka简介



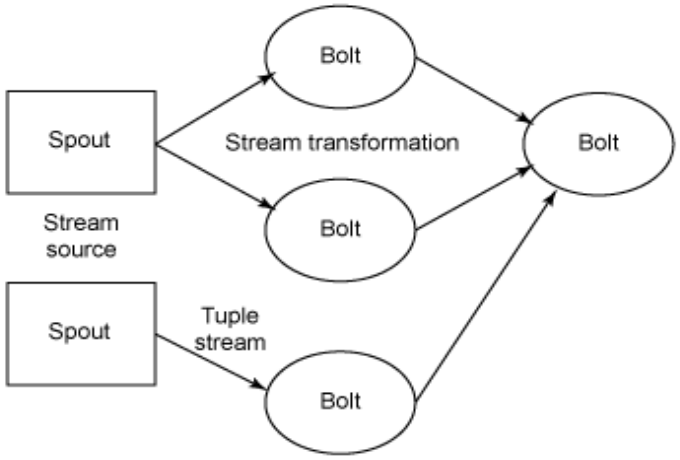


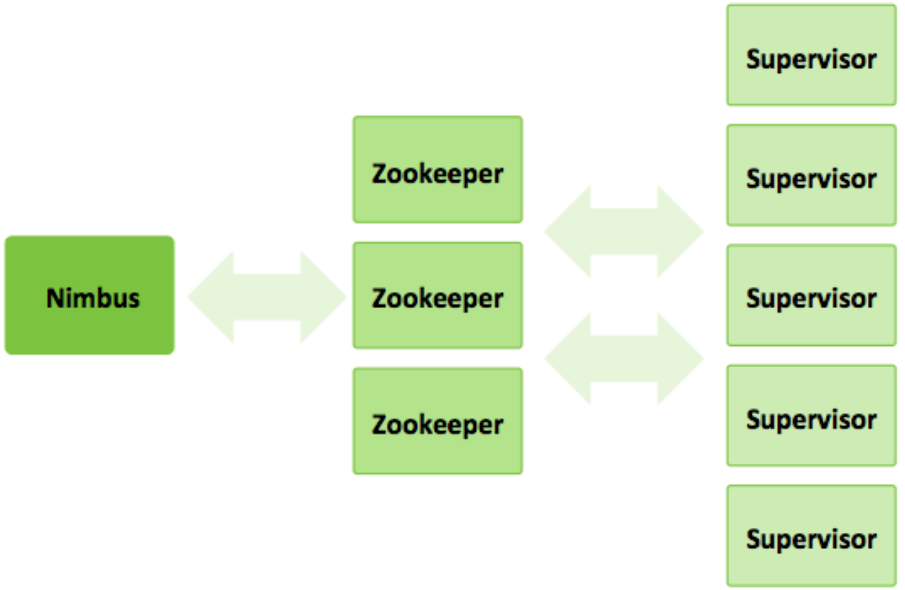


A two server Kafka cluster hosting four partitions (P0-P3) with two consumer groups. Consumer group A has two consumer instances and group B has four.

- 消息在一个Topic Partition中会按照它发送的顺序
- 每个partition分布在集群的每台服务器上，可以为每个partition来设置多个Replication （Leader / Follower）
- 1个topic的replication factor是N，能容忍N-1台机器Failed而没有数据损失

Storm简介





Spout：“水龙头”数据接入单元

Bolt：数据处理单元

Storm的并发的三个层次：

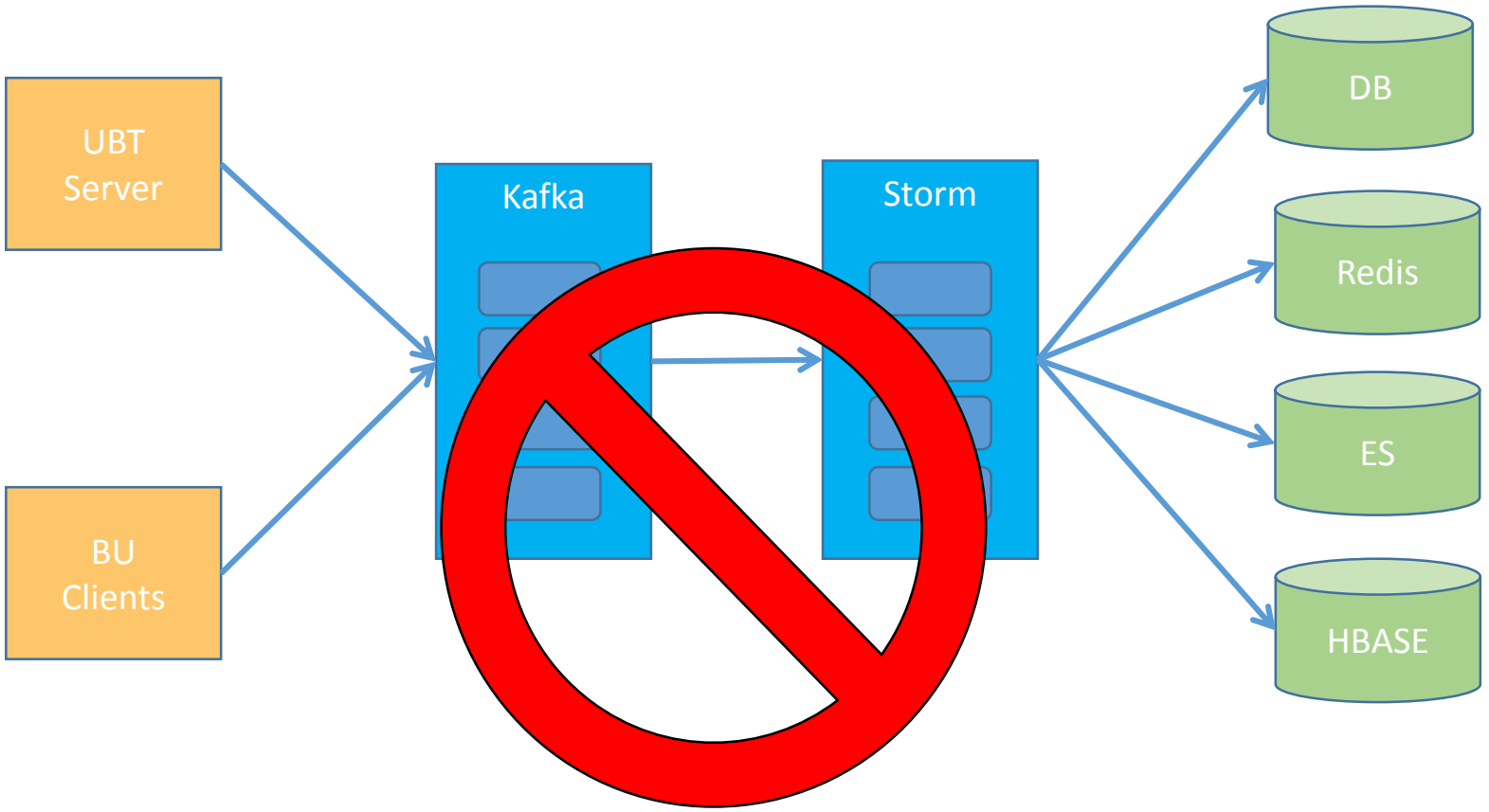
- Worker
- Executor
- Task

Nimbus：Master节点

Supervisor：Worker节点，用来管理worker

两者之间通过ZK来做通讯

系统架构示意图

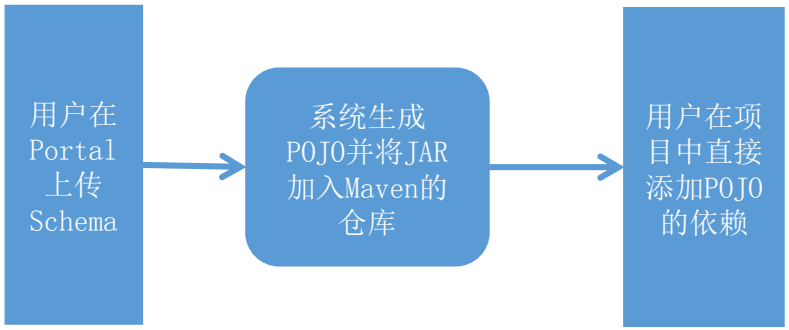


这样远远不够！

平台治理 – 数据共享

数据共享：数据共享的前提是用户能够清楚地知道可以使用的数据源的业务的含义以及其中**数据的Schema**

我们的解决方法是统一的**Portal的站点**和使用AVRO来定义数据的Schema；我们在Storm之上**封装了自己的API**，来自动完成数据的反序列化

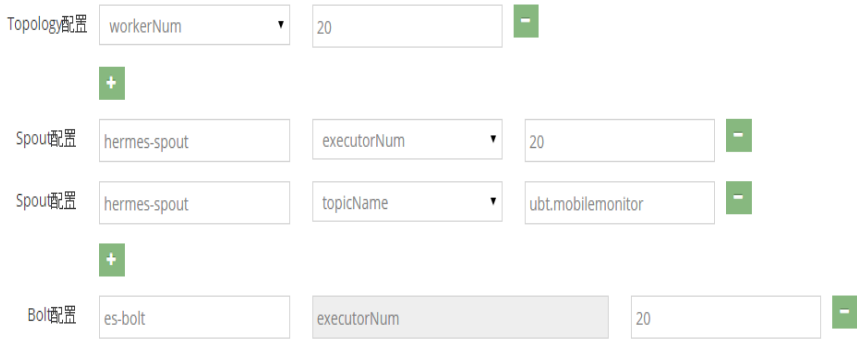



```
private static class ExtractBolt extends AbstractMuiseHermesBoltAutoAcked<UserAction> {
        public ExtractBolt(String topic){
            super(topic, UserAction.class);
        }
        @Override
        public void process(UserAction message, BasicOutputCollector collector) {
            //业务逻辑
            ......
            collector.emit(new Values(vid, target, page, type));
        }
        @Override
        public void specifyOutputFields(OutputFieldsDeclarer declarer) {
            declarer.declare(new Fields("vid", "target", "page", "type"));
        }
    }
```

平台治理 – 资源的控制

Portal允许用户对于作业设置，对每个Spout和Bolt设置并发相关的参数，通过审核后才能生效

Storm之上封装自己的API，屏蔽这些参数的设置




```
public static void main(String[] args) throws AlreadyAliveException, InvalidTopologyException {
        String topologyName = "UBT-Demo-New";
        String topicName = "UBT_TOPIC_Action";
        CtripKafkaSpout hermesSpout = new CtripKafkaSpout(topicName);
        CtripTopologyBuilder builder = new CtripTopologyBuilder(topologyName);
        builder.setSpout("hermes-spout", hermesSpout);
        builder.setBolt("extractbolt", new ExtractBolt(topicName)).localOrShuffleGrouping("hermes-spout");
        builder.setBolt("analysisbolt", new AnalyseBolt(5)).fieldsGrouping("extractbolt", new Fields("vid", "page"));
        builder.setBolt("dashboardbolt", new DashBoardBolt()).localOrShuffleGrouping("analysisbolt");
        Config conf = new Config();
        if(args != null && args.length > 0 && "local".equalsIgnoreCase(args[0])){
            CtripStormSubmitter.submitToLocal(conf, builder);
        }else{
            CtripStormSubmitter.submitToCluster(conf, builder);
        }
    }
}
```

平台治理 – 统一的管理Portal

用户对于作业的管理都能通过Portal上提供的功能完成

实时作业列表

显示自己的job

每页显示 10 条记录　　过滤:

序号	Job名称	Job类型	状态	创建人	创建时间	当前使用版本	集群	操作
0	user_behavior_abtesting	普通作业	审核通过	zw	2015-12-18 16:20:53	4.1.3		
1	search_antibot_count	普通作业	审核通过	p_li	2015-07-17 11:27:44	0.6.6		
2	ubt_entity_broker	普通作业	审核通过	zytu	2015-12-07 15:55:40	2.0.0	prod_tech	
3	ubt_pagevisit_statistics	普通作业	审核通过	zytu	2015-12-07 15:58:29	4.0.0	prod_tech	
4	user_behavior_cmatrix	普通作业	审核通过	jiangzhu	2016-02-13 09:33:45	0.4.3	prod_prerelease	
5	user_behavior_cmatrix_usermetric	普通作业	审核通过	jiangzhu	2016-01-06 17:37:32	0.4.7	prod_tech	
6	ubt_elasticsearch	普通作业	审核通过	jia.liu	2016-03-11 15:27:36	2.2.0	prod_tech	
7	ubt_demo_avro	补数据	审核通过	zhouhao	2015-09-11 15:07:01	0.4	prod_tech	
8	user_behavior_cmatrix_mobilemonitor	普通作业	审核通过	jiangzhu	2016-02-17 11:28:24	0.4.7	prod_prerelease	
9	ops_dbcenter_mobilemonitor	普通作业	审核通过	ftyue	2015-08-17 11:13:36	0.1.27		

展示第1条记录至第10条记录（总计134条记录）　　« ‹ 1 2 3 4 5 › »

+增加新的作业

初期的业务接入

在平台搭建的同时，我们积极推进数据源和相关业务应用的接入

数据源：

- UBT - 携程所有用户的行为日志
- Pprobe - 应用的访问日志

相关应用：

- 基于UBT日志分析的实时报表
- 基于Pprobe日志的实时反爬虫分析程序

平台搭建初期的经验

最初尽可能地做好平台治理的规划：重要的设计和规划都需要提前做好，后续调整时间越晚，付出的成本越大

系统只实现核心的功能：集中力量

尽量早接入业务

- 前提是核心功能基本稳定
- 系统只有真正被用起来才会得到不断的进化
- 低优先级

接入业务需要有一定的量：

- 能够帮助整个平台更快地稳定下来
- 积累技术和运维上的经验

目录

日志相关系统的完善

Storm UI：

worker-6728.log

Prev First Last Next

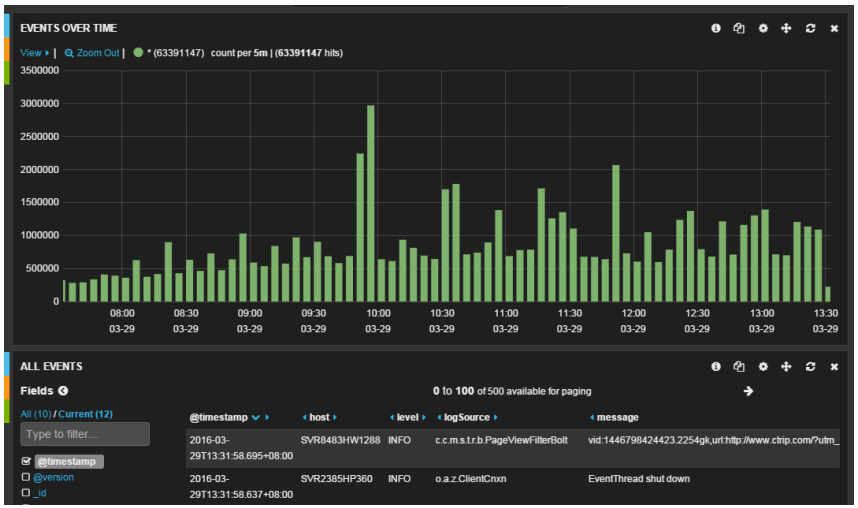
Download Full Log

```
[mail.smtp.auth]=[true]
2016-03-29T12:58:09.279+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [driver.maxWaiting4Calculate]=[-1]
2016-03-29T12:58:09.279+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [app.driver.sendSeqScoreChangedHermes]=
[true]
2016-03-29T12:58:09.279+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [driver.fileTransaction.storePath]=
[./transactions]
2016-03-29T12:58:09.279+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [log4j.appender.CLOG.serverPort]=[63100]
2016-03-29T12:58:09.280+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [app.calculators]=[4]
2016-03-29T12:58:09.280+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [□□□□□□□]=[]
2016-03-29T12:58:09.280+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [cat.server.ip]=[cat.ctripcorp.com]
2016-03-29T12:58:09.280+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [cat.server.httpPort]=[80]
2016-03-29T12:58:09.280+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [app.runtime]=[prd_gray]
2016-03-29T12:58:09.280+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [mail.smtp.user]=[appmail078]
2016-03-29T12:58:09.281+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [app.grayMode]=[true]
2016-03-29T12:58:09.281+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config:
[app.reader.schema.SeqScoreUpdateForAppoint.hermes.maxWaitTime]=[20000]
2016-03-29T12:58:09.281+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [driver.daemonMode]=[true]
2016-03-29T12:58:09.281+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config:
[app.reader.schema.SeqScoreUpdateForAppoint.hermes.groupid.storm]=[hotel.seqrules.appointrank.processer]
2016-03-29T12:58:09.281+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [app.driver.sendSeqScoreChangedRedis]=
[true]
2016-03-29T12:58:09.281+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [writer.AntRedisWriter.cRedisServiceUrl]=
[http://ws.config.framework.ctripcorp.com/configws/]
2016-03-29T12:58:09.281+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [log4j.appender.CLOG.serverIp]=
[collector.logging.sh.ctriptravel.com]
2016-03-29T12:58:09.282+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [app.mertic.name]=
[hotel.data.seq.appointer.gray]
2016-03-29T12:58:09.282+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [app.writer.bufferSize]=[20000]
2016-03-29T12:58:09.282+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [cat.server.port]=[2280]
2016-03-29T12:58:09.282+0800 c.h.d.f.d.c.DataAntsConfig [INFO] Config: [cat.env]=[pro]
```

ES：

Logstash -> Kanban

方便用户进行查询

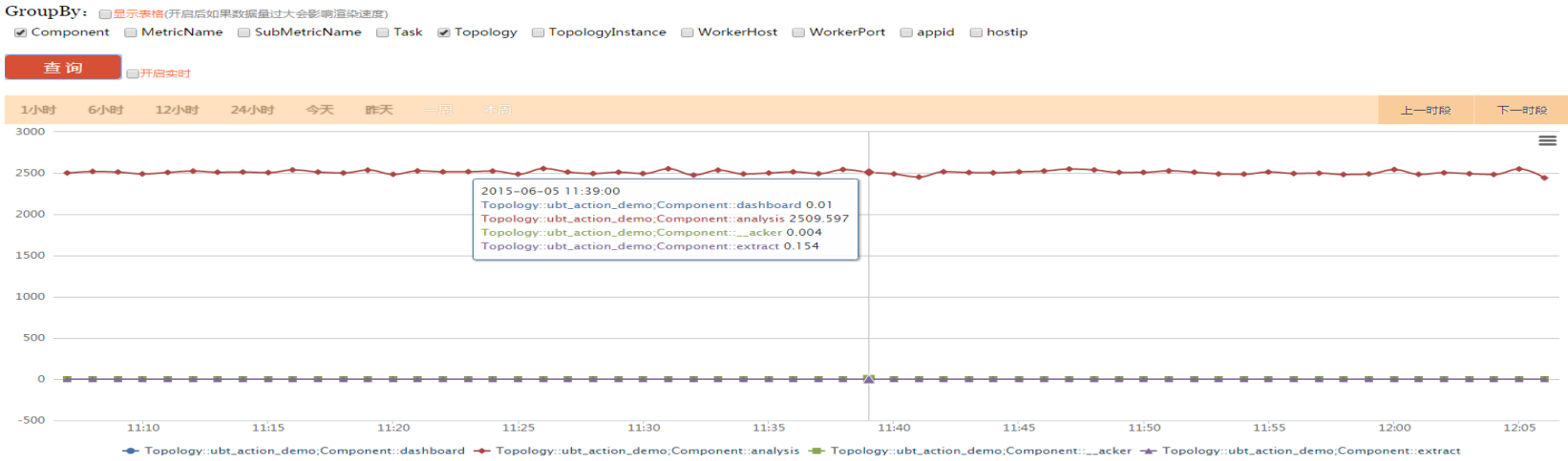


Metrics

基于Storm封装的API中增加通用的埋点：

- 消息从到达Kafka到开始被消费所花费的时间
- Topic / Task Level的一些统计信息

实现自定义的Metrics Consumer把信息输出到携程的Dashboard和Graphite（告警）



告警系统

任何Storm内置的或是用户自定义的Metrics都能够配置
默认配置Topology的Fails数的告警

ubt_pagevisit_statistics错误数	性能监控	2016-02-03 16:43:47
ops_hangout_dimg错误数	性能监控	2016-01-14 16:35:28
flight_ubt_custom_analyser错误数	性能监控	2016-01-20 18:22:51

Storm

快速检索

ID	监控项	状态
9	ubt_elasticsearch错误数	
19	user_behavior_cmatrix_pageview错误数	
23	hotel_coupon错误数	
27	user_behavior_cmatrix_abnorm错误数	
28	user_behavior_cmatrix_custom错误数	
29	user_behavior_cmatrix_malfunction错误数	
37	ttd_owl_statistic错误数	
41	frt_pageview错误数	
42	frt_userprefer错误数	
43	ubt_entity_broker错误数	
44	user_behavior_cmatrix_mobilemonitor错误数	
45	infosec_nile_detect_attack错误数	
62	ubt_useraction_stat_newcluster错误数	
64	basebiz_hermes_abtesting_newhbase错误数	
65	infosec_pprobe_filter_heraloginscan错误数	
68	ubt_direct_conversion错误数	
70	ubt_pagevisit_statistics错误数	
72	user_behavior_cmatrix_usermetric错误数	

通用Spout和Bolt的开发

开发了适配携程通用MQ的Spout，使接入的数据源得到了进一步的扩展，更多的业务数据能够被Storm使用

通用的Bolt，开发了3种针对于不同数据源的Bolt，方便用户把数据输出到外部存储：

- Redis Bolt：仿照原生的实现，集成携程封装的Redis的客户端
- HBASE Bolt：支持Kerberos的认证
- DB Bolt：集成携程的DAL框架

封装API的版本迭代

我们自己在Storm-core和Storm-kafka的基础上封装了自己的API：muise-core

muise-core在不断地迭代和升级，添加各种各样的小功能，并且修复各种各样的问题，随着接入作业的变多，要推动业务进行升级变成一个很沉重的负担

在muise-core 2.0版本我们把API相关的接口都整理了一下，之后的版本最大程度地不修改，然后推动业务全线升级了一遍（当时接入的业务不多）

然后我们把muise-core作为标准的Jar放到每台Supervisor Storm安装目录的lib文件夹下，每次有API升级的时候可以直接替换，然后重启supervisor进程

- 非强制升级 - 等到用户重启topology生效
- 强制升级 - 在和用户确定影响后，重启每个topology

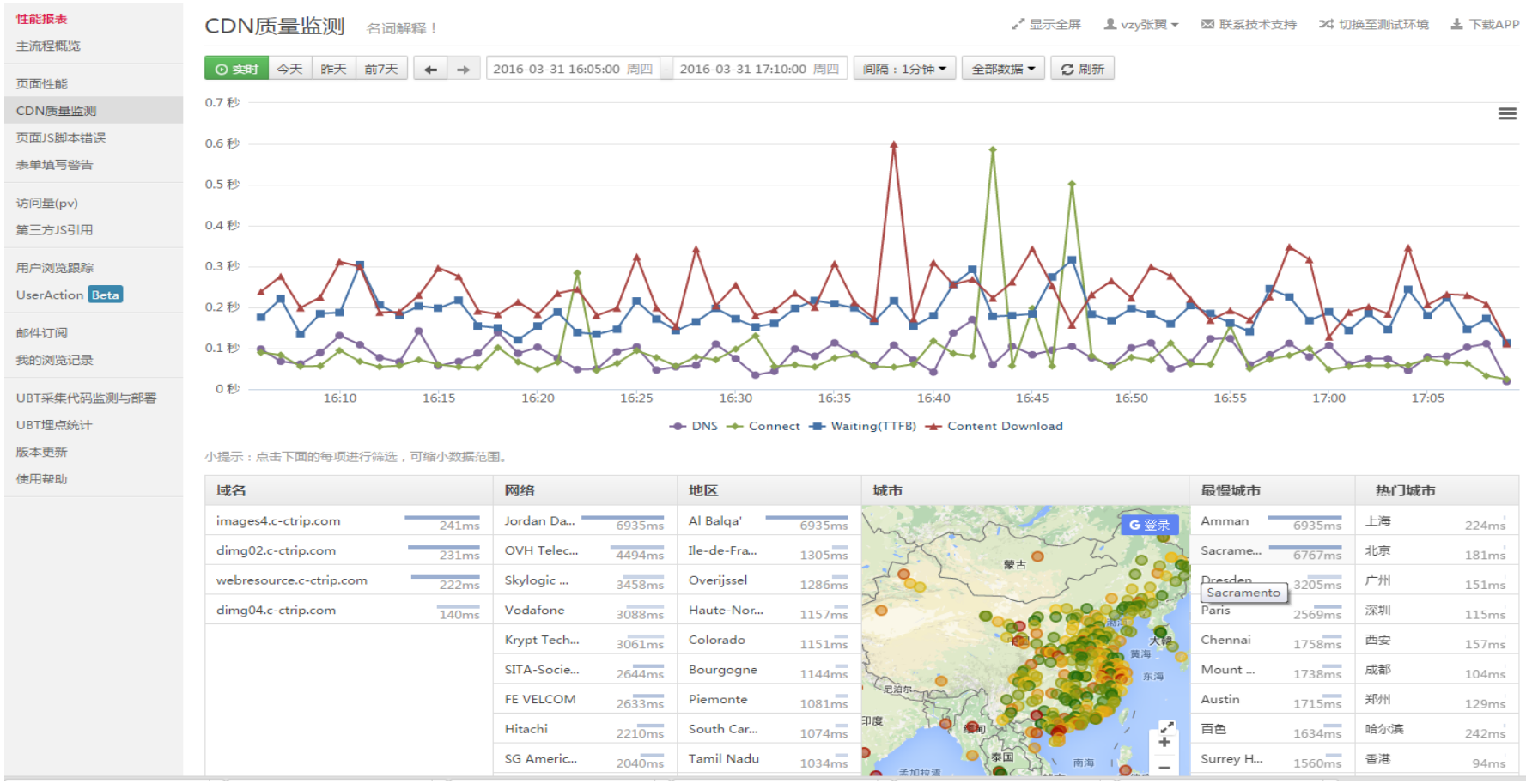
大规模的业务接入

业务方从原来的1个部门（框架）增加到酒店，机票，度假，团队游，攻略等BU以及搜索，风控，信息安全等技术部门，基本上覆盖了携程所有的大部门

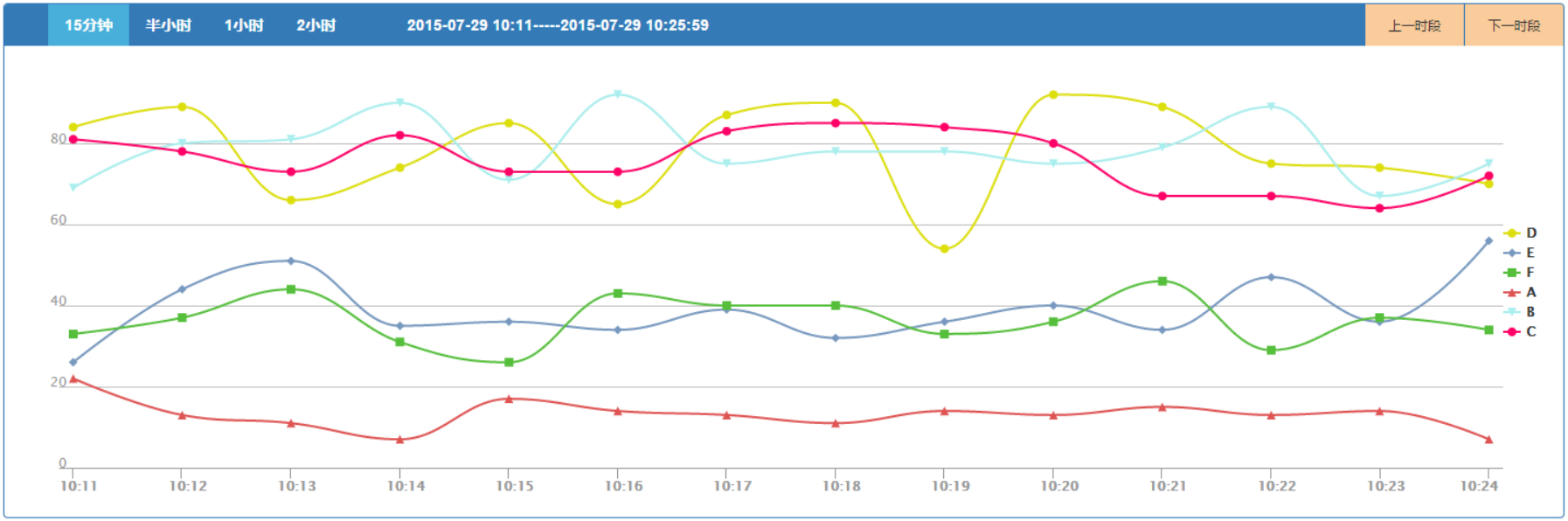
应用类型也比初期要丰富地多，主要应用的类型和领域包括：

- 实时数据报表
- 业务数据的监控
- 基于用户实时行为的营销
- 风控和信息安全的应用

应用实例01


性能报表
主流程概览
页面性能
CDN质量监测
页面JS脚本错误
表单填写警告
访问量(pv)
第三方JS引用
用户浏览跟踪
UserAction Beta
邮件订阅
我的浏览记录
UBT采集代码监测与部署
UBT埋点统计
版本更新
使用帮助
CDN质量监测
名词解释！
显示全屏
vzy张翼
联系技术支持
切换至测试环境
下载APP
实时
今天
昨天
前7天
2016-03-31 16:05:00 周四
2016-03-31 17:10:00 周四
间隔：1分钟
全部数据
刷新
0.7 秒
0.6 秒
0.5 秒
0.4 秒
0.3 秒
0.2 秒
0.1 秒
0 秒
16:10
16:15
16:20
16:25
16:30
16:35
16:40
16:45
16:50
16:55
17:00
17:05
DNS
Connect
Waiting(TTFB)
Content Download
小提示：点击下面的每项进行筛选，可缩小数据范围。
域名
网络
地区
城市
最慢城市
热门城市
images4.c-ctrip.com 241ms
dimg02.c-ctrip.com 231ms
webresource.c-ctrip.com 222ms
dimg04.c-ctrip.com 140ms
Jordan Da... 6935ms
OVH Telec... 4494ms
Skylogic ... 3458ms
Vodafone 3088ms
Krypt Tech... 3061ms
SITA-Socie... 2644ms
FE VELCOM 2633ms
Hitachi 2210ms
SG Americ... 2040ms
Al Balqa' 6935ms
Ile-de-Fra... 1305ms
Overijssel 1286ms
Haute-Nor... 1157ms
Colorado 1151ms
Bourgogne 1144ms
Piemonte 1081ms
South Car... 1074ms
Tamil Nadu 1034ms
登录
蒙古
Sacramento
印度
泰国
南海
Amman 6935ms
Sacrame... 6767ms
Dresden 3205ms
Paris 2569ms
Chennai 1758ms
Mount ... 1738ms
Austin 1715ms
百色 1634ms
Surrey H... 1560ms
上海 224ms
北京 181ms
广州 151ms
深圳 115ms
西安 157ms
成都 104ms
郑州 129ms
哈尔滨 242ms
香港 94ms

应用实例02



功能：

- 实时查看AB Testing的分流效果，有配置问题能够及时发现
- 每个分组的订单数据的监控，如果订单出现下降可以及时停止AB Testing

应用实例03

历史偏好 + **实时偏好** -> 推荐产品

相似应用：

- 攻略根据用户实时的行为推送用户感兴趣的攻略
- 团队游根据用户实时的访问推送限时的优惠券
- 酒店根据用户实时的行为和订单的情况给用户推送营销类的Push消息

我们遇到的坑

我们使用的版本是0.9.4，在这个版本上，我们遇到过两个偶发的问题：

- STORM-763：Nimbus已经将worker分配到其他的节点，但是其他worker的netty客户端不连接新的worker
  应急处理：Kill掉这个worker的进程或是重启相关的作业

- STORM-643：当failed list不为空时，并且一些offset已经超出了Range范围，KafkaUtils会不断重复地去取相关的message

另外我们的用户在使用Storm的过程中也遇到过一些问题，这边简单和大家分享下：

- localOrShuffleGrouping的使用：大多数情况下推荐用户使用；前提上下游的Bolt数要批配；否则会出现下游的大多数Bolt没有收到数据的情况

- Bolt中的成员变量都要是可以序列化的

经验总结

- 大量接入前，监控和告警的相关设施需要完善
- 清晰的说明文档 / Q & A能够节约很多支持的时间
- 把握接入的节奏
  - 全员客服
  - 控制同时接入的项目数
  - 授人以“渔”

目录

Stream CQL

Stream CQL (Stream Continuous Query Language)是华为开源的实时流处理的SQL引擎，它的做法是把StreamCQL -> Storm Topology

Stream CQL的语法和标准的SQL或是HQL很类似，它支持实时处理的窗口函数

下面我们通过一个简单的例子来“感受”下Stream CQL:

- 从kafka中读取数据，类型为ubt_action
- 取出其中的page，type，action，category等字段然后每五秒钟按照page，type字段做一次聚合
- 最后把结果写到console中

Stream CQL例子

Storm:

```
public class CtripKafkaSpout implements IRichSpout {
   ……
}

class ExtractBolt extends
AbstractMuiseHermesBoltAutoAcked<UserAction> {
   ……
}

class ConsoleBolt extends CtripBaseBoltAutoAcked {
   ……
}

class AnalyseBolt extends CtripBaseBoltWithoutAutoAcked {
   ……
}

public static void main(String[] args) {
   ……
   CtripStormSubmitter.submitToCluster(conf, builder);
   ……
}
```

SteamCQL:

```
create input stream kafka_avro (context__page
String, context__type String, action__category
String , action__type String)
serde "HermesSerDe"
source "HermesSourceOp"
properties(avroclass="hermes.ubt.action.UserAction
",topic="ubt.action",groupid="json_hermes");

create output stream console_field(page String, type
String, target String , actionType String, count Int)
sink consoleoutput;

insert into stream console_field select *,count(1)
from kafka_avro [range 5 seconds batch] group by
context__page, context__type;

submit application ubt_cql_demo;
```

Stream CQL的尝试和工作

- 增加Redis，HBASE，HIVE,DB（小表，加载内存）作为Data Source
- 增加HBASE，MySQL / SQL Server，Redis作为数据输出的Sink
- 修正MultiInsert语句解析错误，并反馈到社区
- 为where语句增加了In的功能
- 支持从携程的消息队列Hermes中读取数据

StreamingCQL的应用

Streaming CQL作为Storm的补充
目前的使用场景：能让BI的同学自主地开发逻辑相对简单的**实时数据报表**和**数据分析**的应用

实例：
度假BU需要实时地统计每个用户访问“自由行”，“跟团游”，“半自助游”产品的占比，进一步丰富用户画像的数据

- 数据流：UBT的数据
- Data Source：使用Hive中的product的维度表
- 输出：Hbase

70左右的代码就能完成整个功能，缩短了开发时间

JStorm

JStorm是阿里开源的实时计算引擎，API上兼容Storm，内核使用Java编写

去年它被Storm项目正式接纳，之后会逐步融合到Storm之中去

目前与Storm比较，JStorm在计算性能上，资源的隔离上有一定优势；他也支持与Twitter Heron类似的Back pressure的机制，能更好地应对消息拥塞的情况

阿里的JStorm的团队非常Open，也非常Professional，帮我们解决了不少问题，互相之间的合作也非常愉快！

Jstorm的尝试和经验

我们的目标：把携程现有的实时应用从Storm上迁到JStorm上去
目前使用的版本：2.1.1

经验分享：

1. 与Kafka集成：

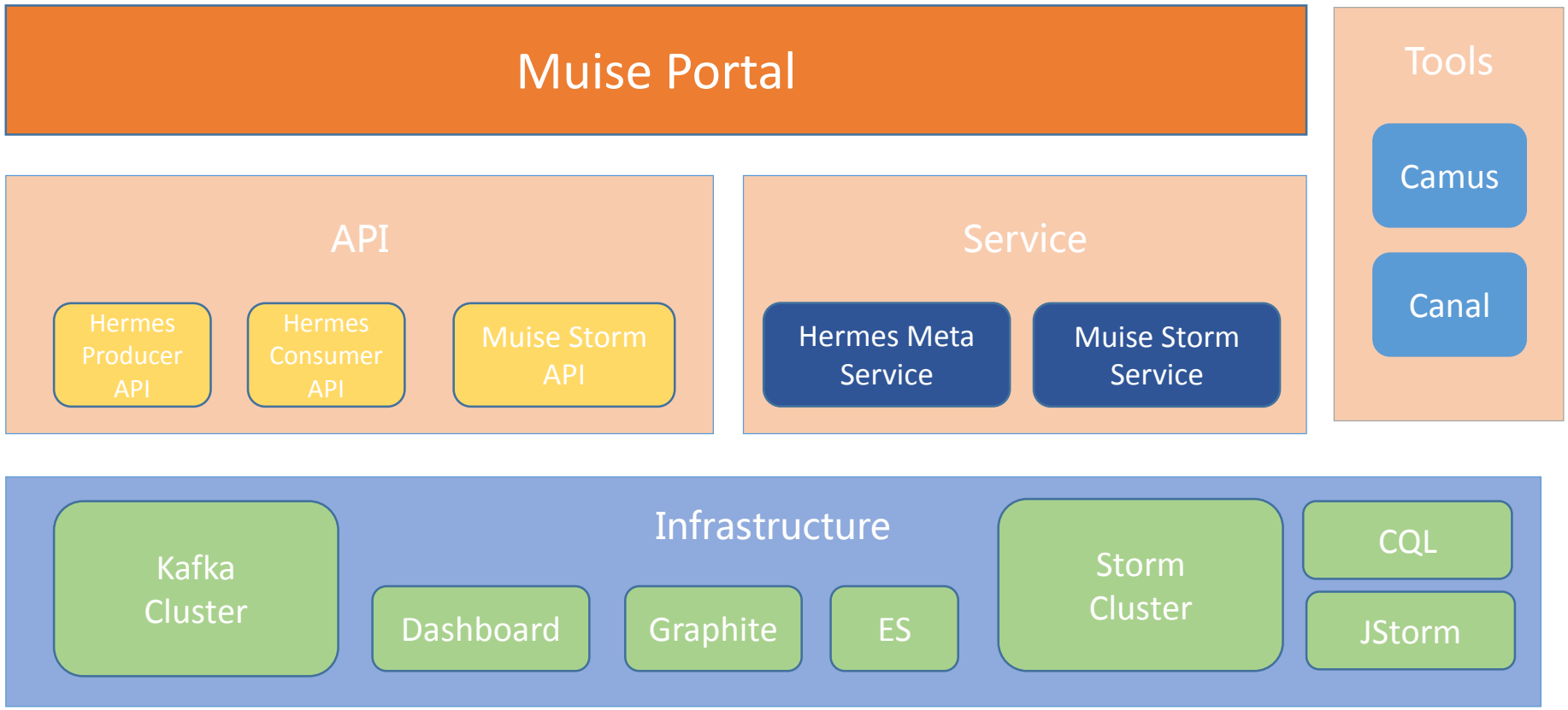
- 在Jstorm中，Spout的实现有两种不同的方式：Multi Thread（nextTuple，ack & fail方法在不同的进程中调用）和Single Thread，原生的Storm的Kafka Spout需要使用Single Thread的方式运行
- 修复了Single Thread模式的1个问题（新版本已经修复）

2. Metrics：

- Jstorm不支持Storm的Metrics Consumer的机制，Jstorm有一套新的Metrics的API，感兴趣的同学可以参看AsmMetrics<T>类，以及子类
- 适配了Kafka Spout和我们Storm的API中的Metrics
- 使用MetricsUploader的功能实现了数据写入Dashboard和Graphite的功能

目录

技术架构的总结


Muise Portal
Tools
Camus
Canal
API
Hermes Producer API
Hermes Consumer API
Muise Storm API
Service
Hermes Meta Service
Muise Storm Service
Infrastructure
Kafka Cluster
Dashboard
Graphite
ES
Storm Cluster
CQL
JStorm

未来的方向

平台整体向Jstorm迁移，贡献社区

- 作业按照优先级逐步向Jstorm迁移
- 熟悉Jstorm的架构和实现
- 参与Jstorm的开发，贡献Jstorm的社区

关注Spark 2.0在实时处理上的进展

- Structured Streaming，支持在流上的SQL的查询，查询可以在运行时改变
- 寻找能够落地的业务场景

调研Flink

- 进行基础的调研，对比Spark Streaming（& 2.0中的Structured Streaming）
- 寻找能够落地的业务场景

Q & A

THANKS

SequeMedia 盛拓传媒 IT168.com ChinaUnix ITPUB