

大数据项目实战

基于Spark交互式用户行为分析系统

主讲人：Gerry

上海育创网络科技有限公司



如何学习大数据项目

■ 项目功能

- ◆ 模块/业务

■ 核心模块功能开发

- ◆ 依据业务场景/分析数据
- ◆ 技术选型
- ◆ 功能开发
- ◆ 测试
- ◆ 性能、调优（没有环境，集群，数据）

项目课程安排

■ Day01

- ◆ 了解项目做什么，涉及到哪些功能
- ◆ 前端如何与我们后台数据处理相结合
- ◆ 熟悉里面所有的JAVA相关的东西（代码，思考）
- ◆ 模块一：会话模块分析讲解(SparkCore)

■ Day02

- ◆ 模块三：各区域热门商品
Top10统计讲解（SparkSQL）
- ◆ 模块四：广告点击流量实时统计讲解(SparkStreaming)

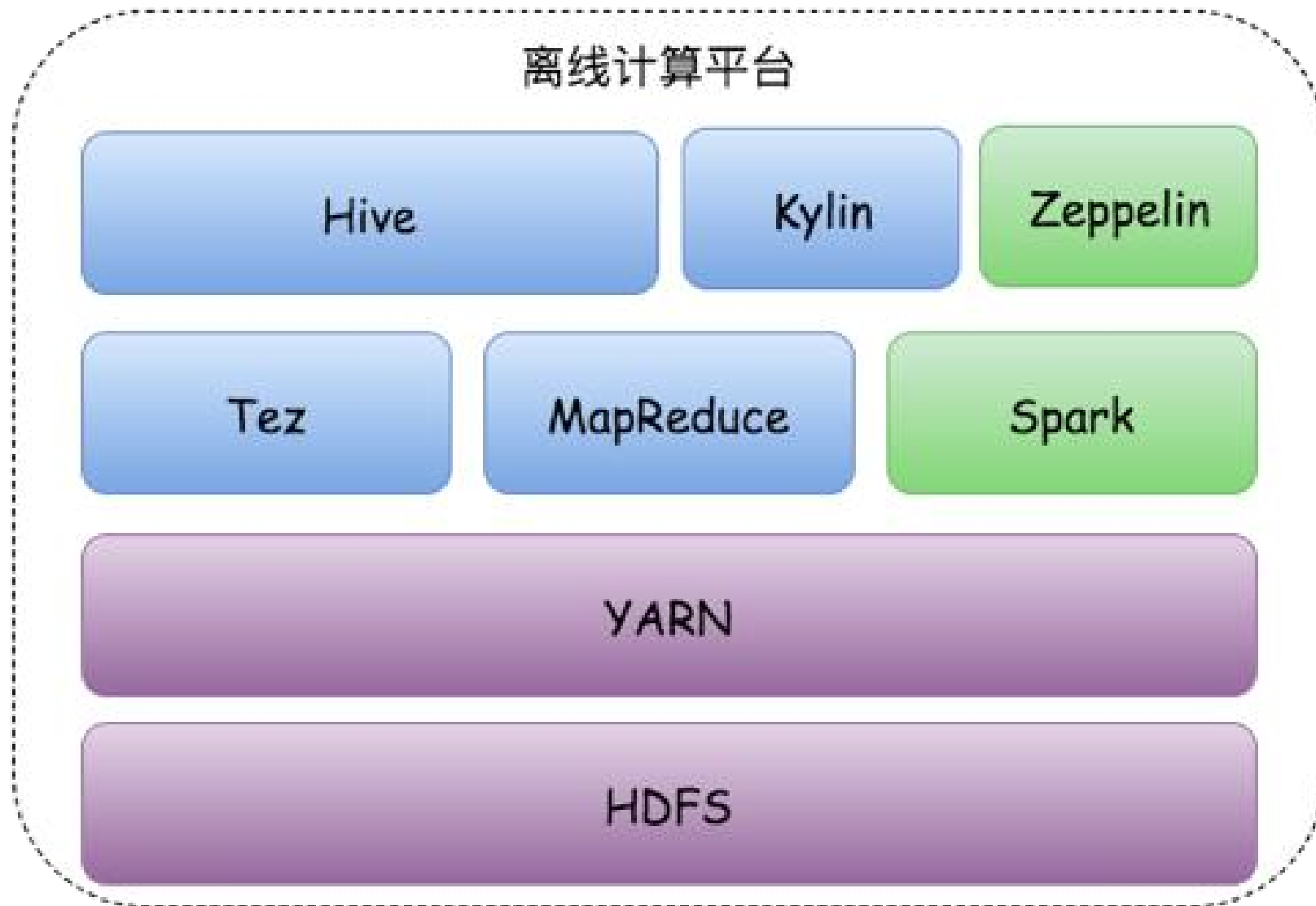
Spark在美团的做法

- 美团是数据驱动的互联网服务，用户每天在美团上的**点击、浏览、下单支付行为**都会产生海量的日志，这些**日志数据**将被**汇总处理、分析、挖掘与学习**，为美团的**各种推荐、搜索系统甚至公司战略目标制定提供数据支持**。大数据处理渗透到了美团各业务线的各种应用场景，选择合适、高效的数据处理引擎能够大大提高数据生产的效率，进而间接或直接提升相关团队的工作效率。
- 美团**最初的数据处理以Hive SQL为主**，底层**计算引擎为MapReduce**，**部分相对复杂的业务会由工程师编写MapReduce程序**实现。随着业务的发展，单纯的Hive SQL查询或者MapReduce程序已经**越来越难以满足数据处理和分析的需求**。

Spark在美团的实践

- 一方面，MapReduce计算模型对**多轮迭代的DAG作业**支持不给力，每轮迭代都需要**将数据落盘**，极大地影响了作业执行效率，另外只提供Map和Reduce这两种计算因子，使得用户在实现迭代式计算（比如：机器学习算法）时**成本高且效率低**。
- 另一方面，在**数据仓库的按天生产中**，由于**某些原始日志是半结构化或者非结构化数据**，因此，对其进行**清洗和转换操作时**，需要结合SQL查询以及复杂的过程式逻辑处理，这部分工作之前是由**Hive SQL结合Python脚本来完成**。这种方式存在效率问题，当数据量比较大的时候，流程的**运行时间较长**，这些ETL流程通常**处于比较上游的位置**，会**直接影响到一系列下游的完成时间以及各种重要数据报表的生成**。
- 基于以上原因，美团在**2014年的时候**引入了Spark。为了充分利用现有Hadoop集群的资源，我们采用了Spark on Yarn模式，所有的Spark app以及MapReduce作业会通过Yarn统一调度执行。

Spark在美团数据平台架构中的位置



Spark在美团的做法

- 经过近两年的推广和发展，从最开始只有少数团队尝试用Spark解决数据处理、机器学习等问题，到现在已经覆盖了美团各大业务线的各种应用场景。从上游的ETL生产，到下游的SQL查询分析以及机器学习等，Spark正在逐步替代MapReduce作业，成为美团大数据处理的主流计算引擎。目前美团Hadoop集群用户每天提交的Spark作业数和MapReduce作业数比例为4：1，对于一些上游的Hive ETL流程，迁移到Spark之后，在相同的资源使用情况下，作业执行速度提升了十倍，极大地提升了业务方的生产效率。
- Spark在美团的实践，包括我们基于Spark所做的平台化工作以及Spark在生产环境下的应用案例。其中包含Zeppelin结合的交互式开发平台，也有使用Spark任务完成的ETL数据转换工具，数据挖掘组基于Spark开发了特征平台和数据挖掘平台，另外还有基于Spark的交互式用户行为分析系统以及在SEM投放服务中的应用。

Spark在交互式用户行为分析系统中的实践

- 美团的交互式用户行为分析系统，用于**提供对海量的流量数据进行交互式分析的功能**，系统的**主要用户为公司内部的PM和运营人员**。普通的BI类报表系统，只能够提供对聚合后的指标进行查询，比如PV、UV等相关指标。
- 但是PM以及运营人员除了**查看一些聚合指标**以外，还需要**根据自己的需求去分析某一类用户的流量数据**，进而**了解各种用户群体在App上的行为轨迹**。根据这些数据，**PM可以优化产品设计，运营人员可以为自己的运营工作提供数据支持**，用户核心的几个诉求包括：
 1. 自助查询，不同的PM或运营人员可能随时需要执行各种各样的分析功能，因此系统需要支持用户自助使用。
 2. 响应速度，大部分分析功能都必须在几分钟内完成。
 3. 可视化，可以通过可视化的方式查看分析结果。

Spark在交互式用户行为分析系统中的实践

■ 要解决上面的几个问题，技术人员需要解决以下**两个核心问题**：

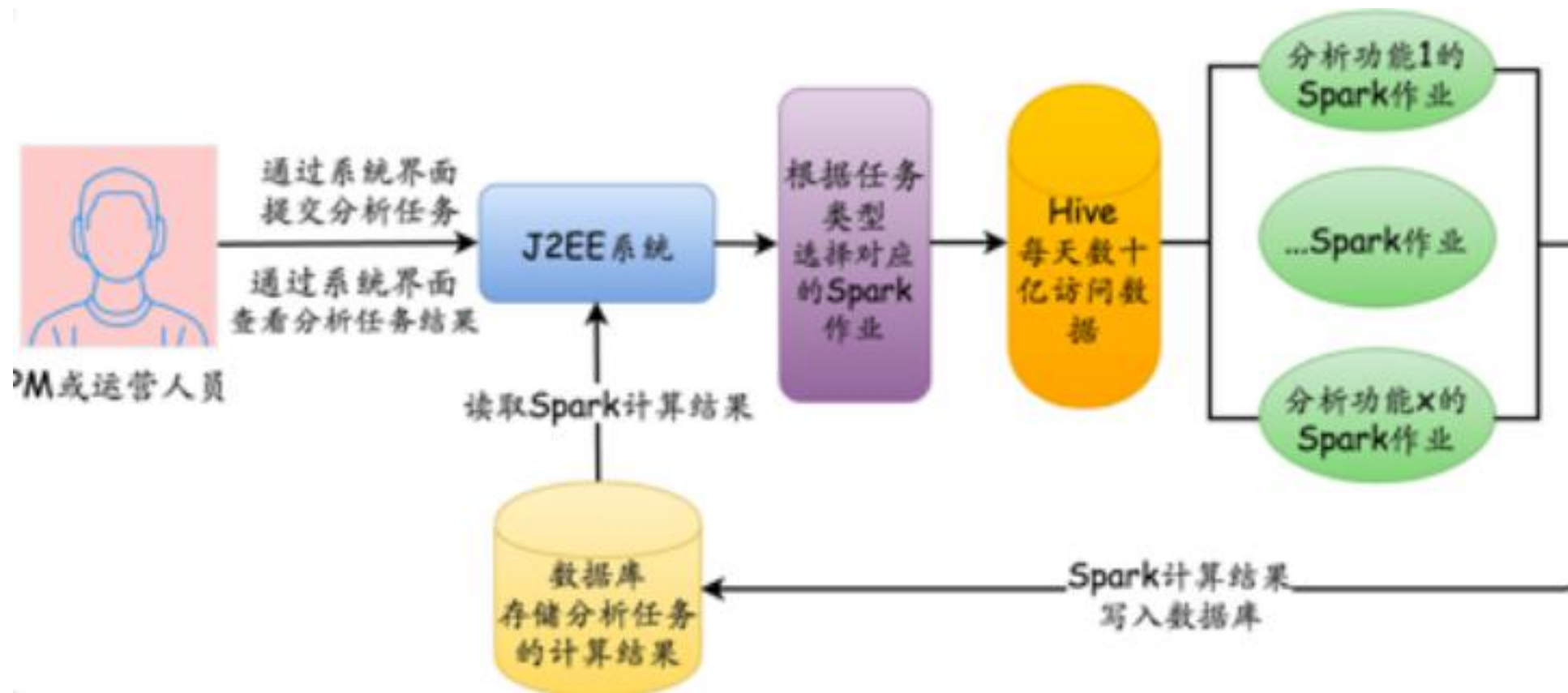
1. 海量数据的处理，用户的流量数据全部存储在Hive中，数据量非常庞大，每天的数据量都在数十亿的规模。
2. 快速计算结果，系统需要能够随时接收用户提交的分析任务，并在几分钟之内计算出他们想要的结果。

■ 要解决上面两个问题，目前**可供选择的技术主要有两种：MapReduce和Spark**。在初期架构中选择了使用MapReduce这种较为成熟的技术，但是通过测试发现，**基于MapReduce开发的复杂分析任务需要数小时才能完成，这会造成极差的用户体验，用户无法接受。**

Spark在交互式用户行为分析系统中的实践

- 尝试使用Spark这种**内存式的快速大数据计算引擎**作为系统架构中的核心部分，主要使用了**Spark Core**以及**Spark SQL**两个组件，来实现各种复杂的业务逻辑。实践中发现，虽然Spark的性能非常优秀，但是在目前的发展阶段中，还是**或多或少会有一些性能以及OOM方面的问题**。
- 因此在项目的开发过程中，对大量Spark作业进行了各种各样的性能调优，包括**算子调优、参数调优、shuffle调优以及数据倾斜调优**等，最终实现了所有Spark作业的**执行时间都在数分钟左右**。并且在实践中解决了一些shuffle以及数据倾斜导致的OOM问题，保证了系统的稳定性。

系统架构与工作流程



- 该系统上线后效果良好：90%的Spark作业运行时间都在5分钟以内，剩下10%的Spark作业运行时间在30分钟左右，该速度足以快速响应用户的分析需求。通过反馈来看，用户体验非常良好。目前每个月该系统都要执行数百个用户行为分析任务，有效并且快速地支持了PM和运营人员的各种分析需求。

项目工程 (Maven Project、JAVA+SCALA)

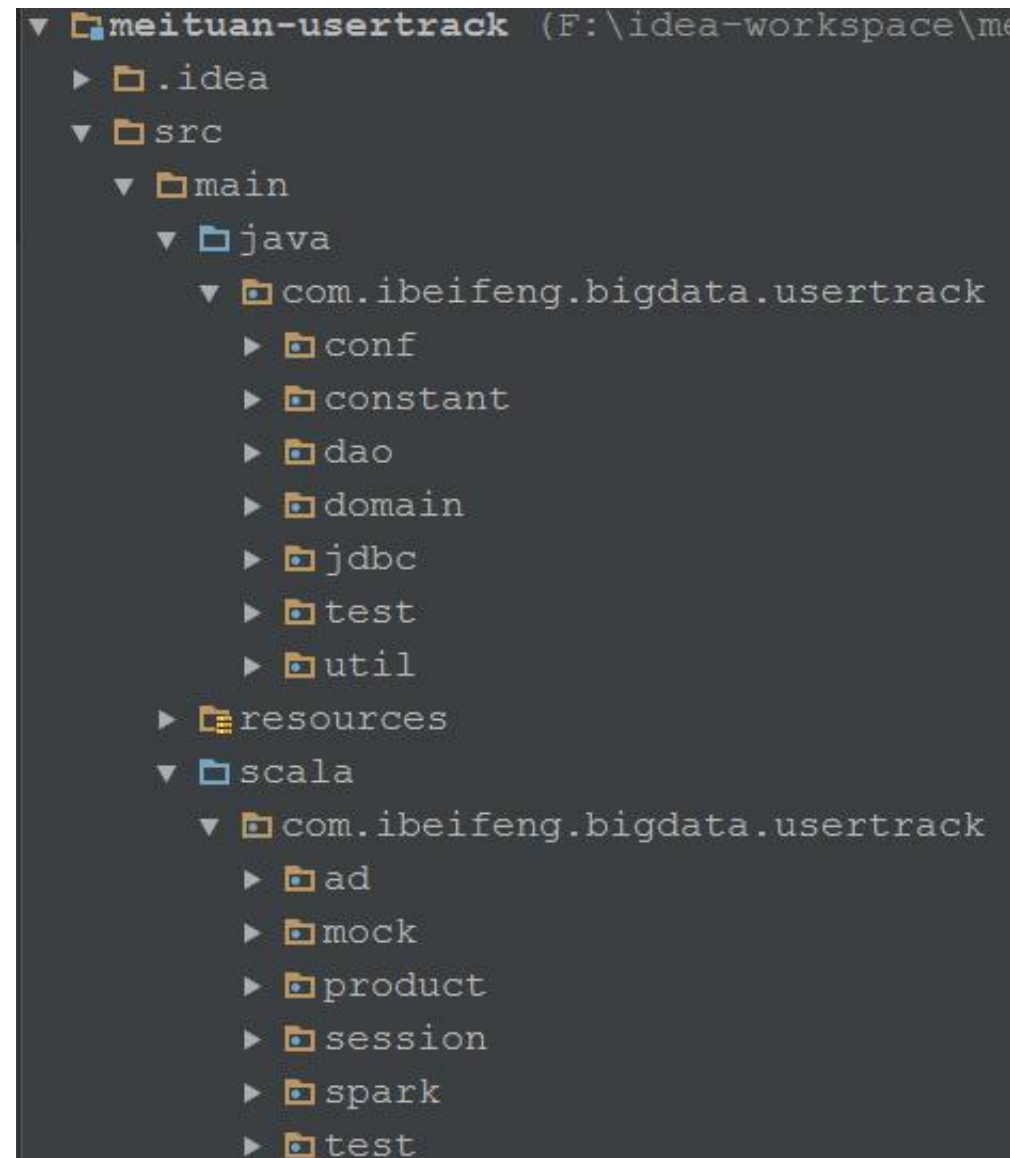
■ Maven Project

- ◆ CMD命令行创建
- ◆ 导入IDEA工具
- ◆ 配置POM.xml文件

■ 相关说明

- ◆ 前端、工具类基于JAVA编程
- ◆ 数据分析，基于SCALA编程
- ◆ 涉及到RDBMS数据交互基于JDBC

■ JSON是fastjson解析



如何实现“交互式分析”与“自助查询”

■ 项目的核心抽象

◆ Task (tb_task)

每个用户的一个需求分析功能就是一个Task/任务

◆ Task对应于RDBMS中的一张表

◆ task_param , 格式数据 , 封装了用户所有的对改任务的参数

```
{
  "startDate": "20160910",
  "endDate": "20160911"
}
{
  "startDate": "20160901",
  "endDate": "20160911",
  "sex": "female"
}
```



基于SPARK交互式行为分析系统

基于**SPARK**交互式 行为分析系统

模块一：用户访问SESSION分析

模块二：页面单跳转化率统计

模块三：各区域热门商品统计

模块四：广告点击流量实时统计

模块一：用户访问Session分析



模块三：各区域热门商品统计Top10

■ 需求分析

- ◆ 每个区域点击最多的 TOP10 商品

■ 数据调研

- ◆ 分析的数据：用户行为分析数据（user_visit_action），存储Hive表中

- ◆ 分析的字段：click_product_id（点击商品ID）、city_id（城市ID）

- ◆ 纬度信息表

- ▶ 城市信息表（city_info），存储RDBMS表中

city_id（城市ID）、city_name（城市名称）、area（城市所属区域）

- ▶ 商品信息表（product_info），存储Hive表中

product_id（商品ID）、product_name（商品名称）、extend_info（扩展信息，JSON格式数据）、product_status

模块四：广告流量实时统计分析

■ 思路分析（关键词）

◆ 1，实时统计

数据源源不断地实时地产生，就需要立即进行计算分析，快速将分析统计结果进行返回，以便前端页面进行展示。

◆ 2，广告点击流量

要分析的数据，当用户点击广告时，就会产生日志记录，表示点击。

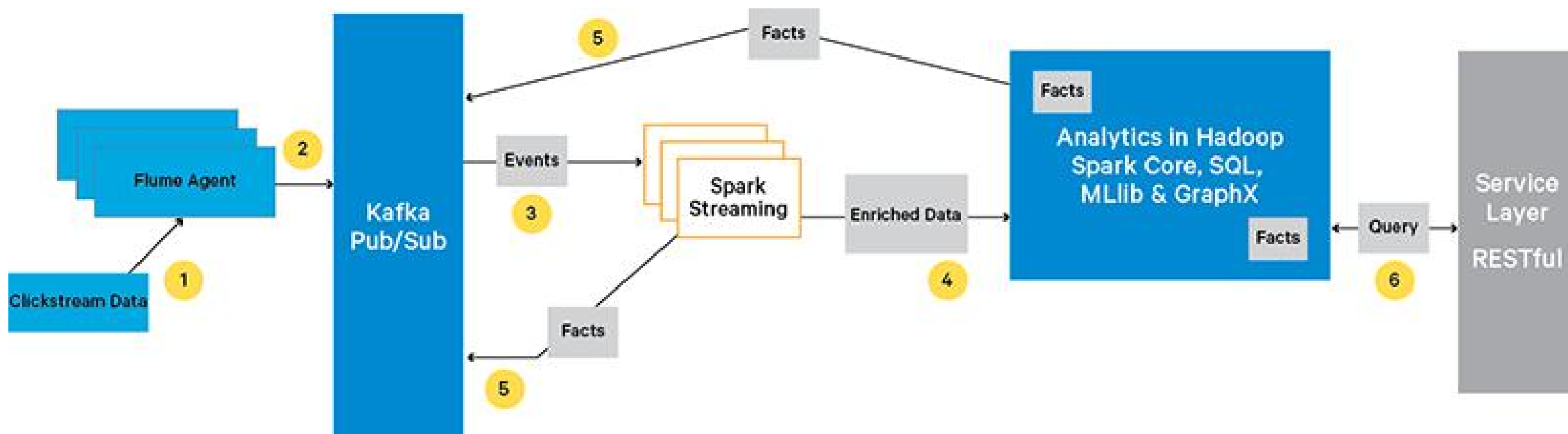
◆ 3，对于一个电商网站、CMS网站或者门户网站来说，收入的一个重要来源就是网站广告。

■ 美团日志收集系统：Flume-NG

<http://tech.meituan.com/mt-log-system-arch.html>

<http://tech.meituan.com/mt-log-system-optimization.html>

SparkStreaming实时流式计算



SparkStreaming实时流式计算

The `spark-submit` command we use to run the Spark app is shown below. It reflects all the options that, together with coding improvements, resulted in significantly less processing time: from 4-8 minutes to under 25 seconds.

```

/opt/app/dev/spark-1.5.2/bin/spark-submit \
--jars \
/opt/cloudera/parcels/CDH/jars/zkclient-0.3.jar,/opt/cloudera/parcels/CDH/jars/kafka_2.10-0.8.1.1.jar,/opt/app,
--files /opt/app/dev/spark-1.5.2/conf/hive-site.xml,/opt/app/dev/jars/log4j-eir.properties \
--queue spark_service_pool \
--master yarn \
--deploy-mode cluster \
--conf "spark.ui.showConsoleProgress=false" \
--conf "spark.driver.extraJavaOptions=-XX:MaxPermSize=6G -XX:+UseConcMarkSweepGC -Dlog4j.configuration=log4j-e"
--conf "spark.sql.tungsten.enabled=false" \
--conf "spark.eventLog.dir=hdfs://nameservice1/user/spark/applicationHistory" \
--conf "spark.eventLog.enabled=true" \
--conf "spark.sql.codegen=false" \
--conf "spark.sql.unsafe.enabled=false" \
--conf "spark.executor.extraJavaOptions=-XX:+UseConcMarkSweepGC -Dlog4j.configuration=log4j-eir.properties" \
--conf "spark.streaming.backpressure.enabled=true" \
--conf "spark.locality.wait=1s" \
--conf "spark.streaming.blockInterval=1500ms" \
--conf "spark.shuffle consolidateFiles=true" \
--driver-memory 10G \
--executor-memory 8G \
--executor-cores 20 \
--num-executors 20 \
--class com.bigdata.streaming.OurApp \ /opt/app/dev/jars/OurStreamingApplication.jar external_props.conf

```

模块四：广告流量实时统计分析

■ 1、过滤数据

◆ 过滤黑名单用户点击广告流量数据

分析：

黑名单：存储刷单用户

思考：

黑名单用户从哪里来？？？？

当一个用户某一天/一段时间，点击某个广告的次数超过100次

实现动态黑名单用户的实现

■ 2、实时统计每天各省TOP5热门广告

updateStateByKey

■ 3、实时统计最近某个时段广告点击趋势

window

■ 4、数据格式

timestamp

时间戳，用户何时点击广告

province

省份，用户在哪个省份点击广告

city

城市

userid

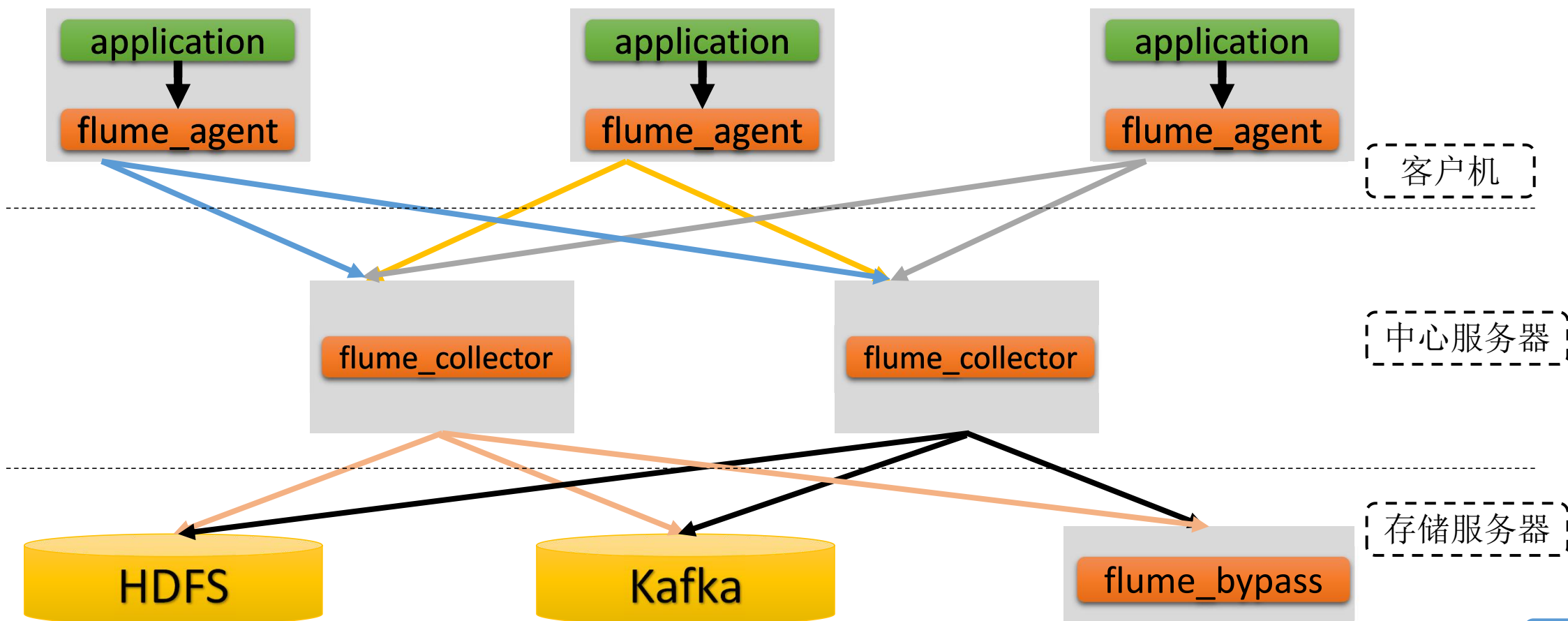
点击广告的用户

adid

点击的广告

基于Flume的美团日志收集系统

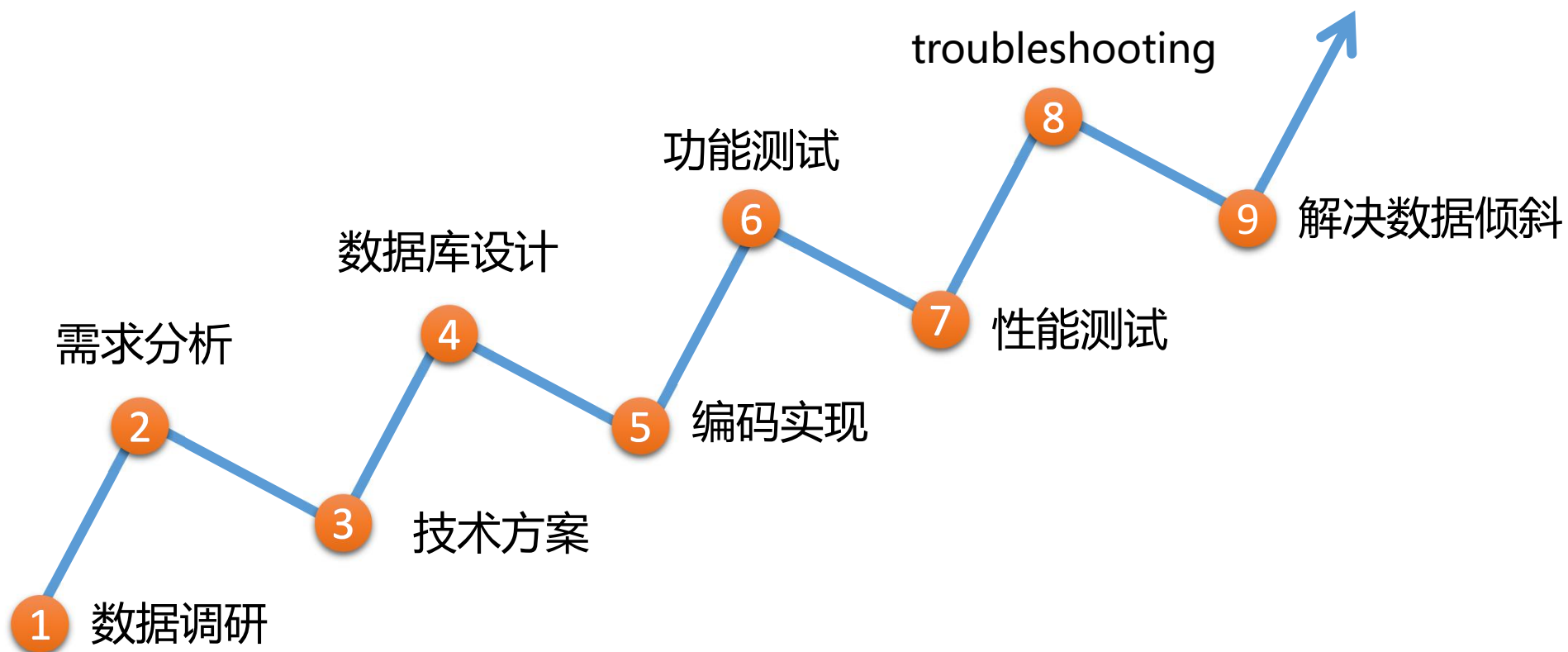
- 美团的日志收集系统负责美团的所有业务日志的收集，并分别给Hadoop/Spark平台提供离线数据和Storm平台提供实时数据流，**基于Flume设计搭建而成**，目前**每天收集和处理的日志数据约T级别**。



基于Flume的美团日志收集系统

- 1、整个系统**分为三层：Agent层，Collector层和Store层**。
 - ◆ Agent层**每个机器部署一个进程**，负责对单机的日志收集工作；
 - ◆ Collector层部署在中心服务器上，**负责接收Agent层发送的日志，并且将日志根据路由规则写到相应的Store层中**；
 - ◆ Store层负责提供永久或者临时的日志存储服务，或者将日志流导向其它服务器。
- 2、Agent到Collector**使用LoadBalance策略**，将**所有的日志均衡地发到所有的Collector上**，达到负载均衡的目标，同时并处理单个Collector失效的问题。
- 3、Collector层的目标主要有三个：**SinkHdfs, SinkKafka和SinkBypass**。分别提供离线的数据到Hdfs，和提供实时的日志流到Kafka和Bypass。其中SinkHdfs又根据日志量的大小分为SinkHdfs_b，SinkHdfs_m和SinkHdfs_s三个Sink，以提高写入到Hdfs的性能。
- 4、对于Store来说，**Hdfs负责永久地存储所有日志；Kafka存储最新的7天日志**，并给实时系统提供实时日志流；Bypass负责给其它服务器和应用提供实时日志流。

大数据项目开发流程



我们只做唯一

我们只做唯一





THANK YOU

上海育创网络科技有限公司