

# Apache Kylin在京东云的实践

王晓雨

Apache Kylin PMC

京东数据云

2016年4月

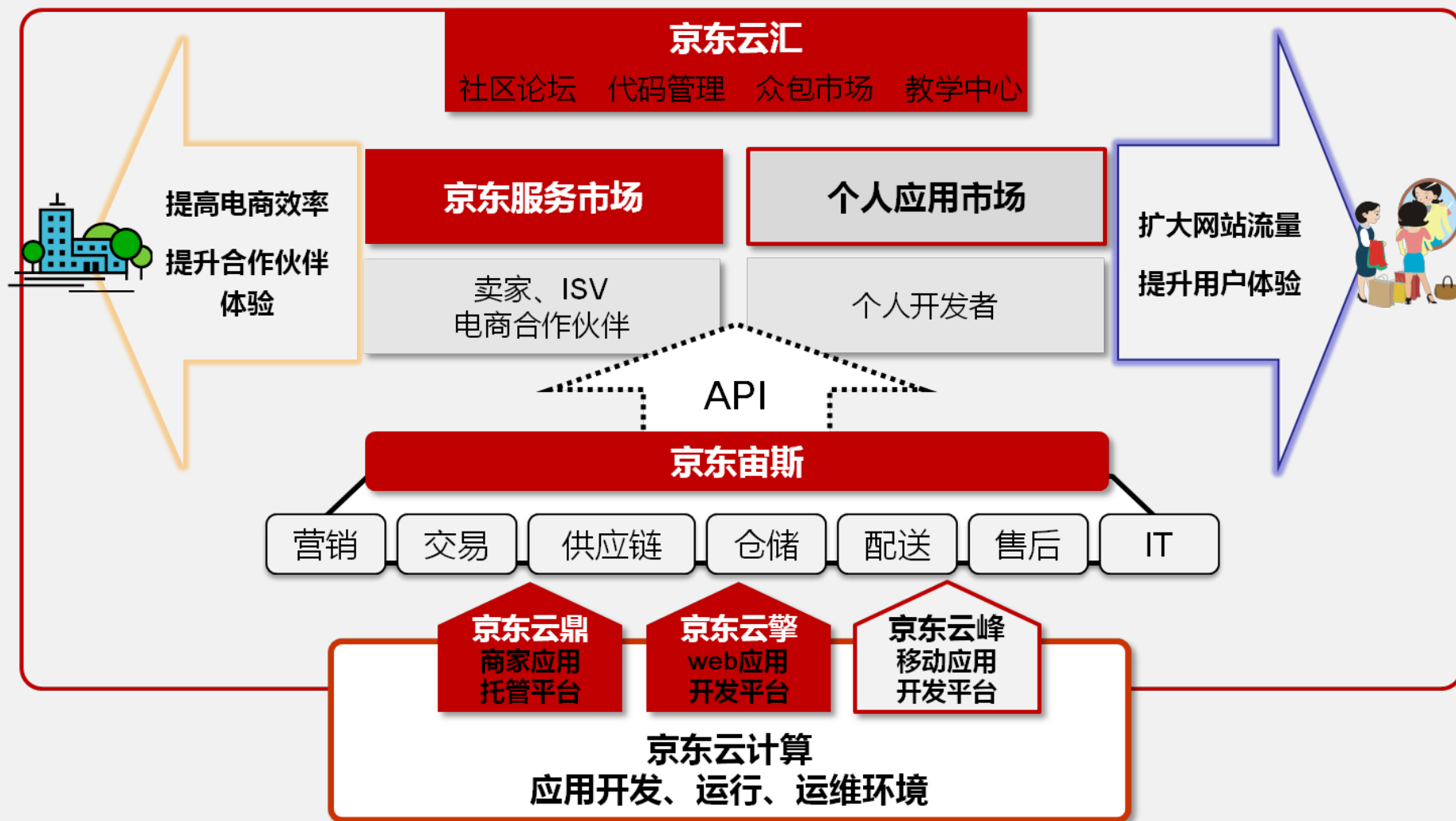
# 目录

- 1** Apache Kylin@JCloud JOS Api调用分析
- 2** Apache Kylin@JCloud 云海
- 3** Apache Kylin@JCloud 数据云
- 4** Apache Kylin@JCloud 二次开发
- 5** Apache Kylin@JCloud 实践经验

# Apache Kylin

## @JCloud JOS Api调用分析

# Apache Kylin@JCloud JOS Api调用分析



# Apache Kylin@JCloud JOS Api调用分析

Apache Kylin作为运营人员对JOS API调用情况分析的OLAP多维分析查询引擎

JOS开放接近500个API，每天调用量在7亿次左右。日志数据量为100G/天。针对API的调用情况进行多维分析，分析查询延迟要求达到秒级，并使用BI工具进行分析展现。

分析包括对API调用成功率，调用延时等情况的分析，帮助各个应用进行产品改进和故障定位等。

JOS的API访问日志数据通过近实时抓取存储在Hive数据仓库中。需要一种能够在大数据量情况下进行交互式多维分析的OLAP引擎。并且要支持和BI工具的集成，提供标准的JDBC、ODBC接口。

# Apache Kylin@JCloud JOS Api调用分析

使用情况：

数据量：单个Cube最大维度16个，最大数据条数100+亿，Cube占用最大存储空间400G。

查询场景：跨天，周，月方式多维分析查询

优化：

基于原始数据Cube的Row Key做分组，减少Cube存储空间。

通过视图做轻度汇总，基于此结果Build Cube，减少基础数据量大小。

性能：

数据分析人员采用BI工具进行查询，95%的查询响应时间在15秒以内。

2015年6月上线使用

# Apache Kylin@JCloud JOS Api调用分析

遇到的问题：

1、复杂函数表达式

```
max(case when field_name = 'spName' then uid_value else " end) as spName,  
max(case when field_name = 'name' then uid_value else " end) as name
```

2、维表重复数据

3、16个维度，Cube构建时间长，数据膨胀严重

解决方式：

1、通过Hive视图方式将复杂表达式在视图中进行处理，Kylin直接用结果字段

2、在Kylin Build之前进行数据清洗

3、将维度进行分组，16个维度按照查询场景分为三组每个组不超过6个维度

**Apache Kylin  
@JCloud 云海**



# Apache Kylin@JCloud 云海

第三方合作商ISV

开发者

商家

## 京东云海集成开发环境IDE

数据质量管理

离线计算

实时计算

JOS API

元数据管理

批处理引擎

Spark Streaming

MySQL

任务管理

数据仓库

Kylin

任务监控

Spark SQL

Hive

HBase

实时Kafka源

Hadoop

离线数据推送

# Apache Kylin@JCloud 云海

Apache Kylin作为云海数据开放后台原始数据高频率低延时查询引擎

云海：

云海将京东POP商家数据进行ETL处理，分主题进行数据表开放。

ISV通过商家授权后，通过京东云海数据开放API获取授权后的数据进行应用开发。

ISV开发的应用在京东服务市场进行销售。

商家购买应用对自己店铺各项情况进行分析。

使用场景：

数据开放API中允许ISV定义标准SQL模板，在接口调用时传递不同参数来查询数据，SQL模板相对固定；

接口单次调用返回数据量限制为5000条；

接口查询延迟要保证在毫秒级别，并支持高并发调用。

# Apache Kylin@JCloud 云海

使用情况：

数据量：单个Cube最大维度8个，最大数据条数4亿，最大存储空间800G。30个Cube占用总空间4T左右。

查询场景：SELECT X FROM T WHERE X ORDER BY X OFFSET N LIMIT N

性能：单次查询返回数据条数限制在5000条以内，查询QPS在50左右，所有查询平均响应时间200ms，查询QPS在200左右平均响应时间可以保持在1s以内。

查询的并发能力和响应时间和HBase集群规模有关，这两个场景的数据只使用了一个小集群（30台和其他业务共用），可以对Apache Kylin Query Server和HBase集群水平扩容来提高并发查询能力和减小响应时间。

2015年5月上线使用

# Apache Kylin@JCloud 云海

遇到的问题：

- 1、不能查询事实表内所有字段明细数据
- 2、Job Server多任务并发构建
- 3、Query Server多查询并发
- 4、Merge跨越较大分区段时，JobServer内存不足
- 5、Build Cube Step 2 “Extract Fact Table Distinct Columns” OOM

解决方式：

- 1、二次开发
- 2、按照实际情况配置JobServer内存及Job并发数
- 3、多QueryServer水平扩展，优化HBase参数
- 4、减小Merge间隔
- 5、`mapreduce.reduce.java.opts`，`mapreduce.reduce.memory.mb`

**Apache Kylin**  
**@JCloud 数据云**

# Apache Kylin@JCloud 数据云



# Apache Kylin@JCloud 数据云

Apache Kylin作为京东数据云在线数据分析产品数千工坊底层支撑平台

数千工坊（DF）：  
通过拖拽，所见即所得的数据操作方式，对海量数据进行可视化分析，使用户轻松上手大数据处理及分析。

2016年4月上线对外开放

## 产品特性

操作简单方便



拖拽方式进行所见即所得的数据处理，低门槛成就数据分析，轻松掌握数据分析利器。

多维数据模型



多维度数据模型，轻松自定义数据处理逻辑，为业务方向提供依据。

极速海量数据



支持百亿级数据超低延迟处理，适应高并发的交互式分析场景。

丰富可视化效果



支持丰富的数据可视化图表效果，智能推荐个性化报表展示方案。

## 产品功能



数图联动，操作简单

通过拖拽方式，自定义数据处理逻辑，进行大数据快速处理，获得可视化图表。



数据可视化功能丰富

支持线图、柱图、散点图、饼图、雷达图、热力图、仪表盘、漏斗、韦恩图等多种可视化图表形式，满足用户不同使用场景。



表格下钻，数据多样化导出

支持单元格下钻功能，使数据分析更灵活，满足多种数据导出格式。



ANSI SQL接口

底层引擎支持ANSI SQL，并提供ODBC/JDBC驱动无缝集成BI工具。

# Apache Kylin@JCloud 数据云





# Apache Kylin@JCloud 数据云



# Apache Kylin @JCloud 二次开发

# Apache Kylin@JCloud 二次开发

对Apache Kylin的主要二次开发

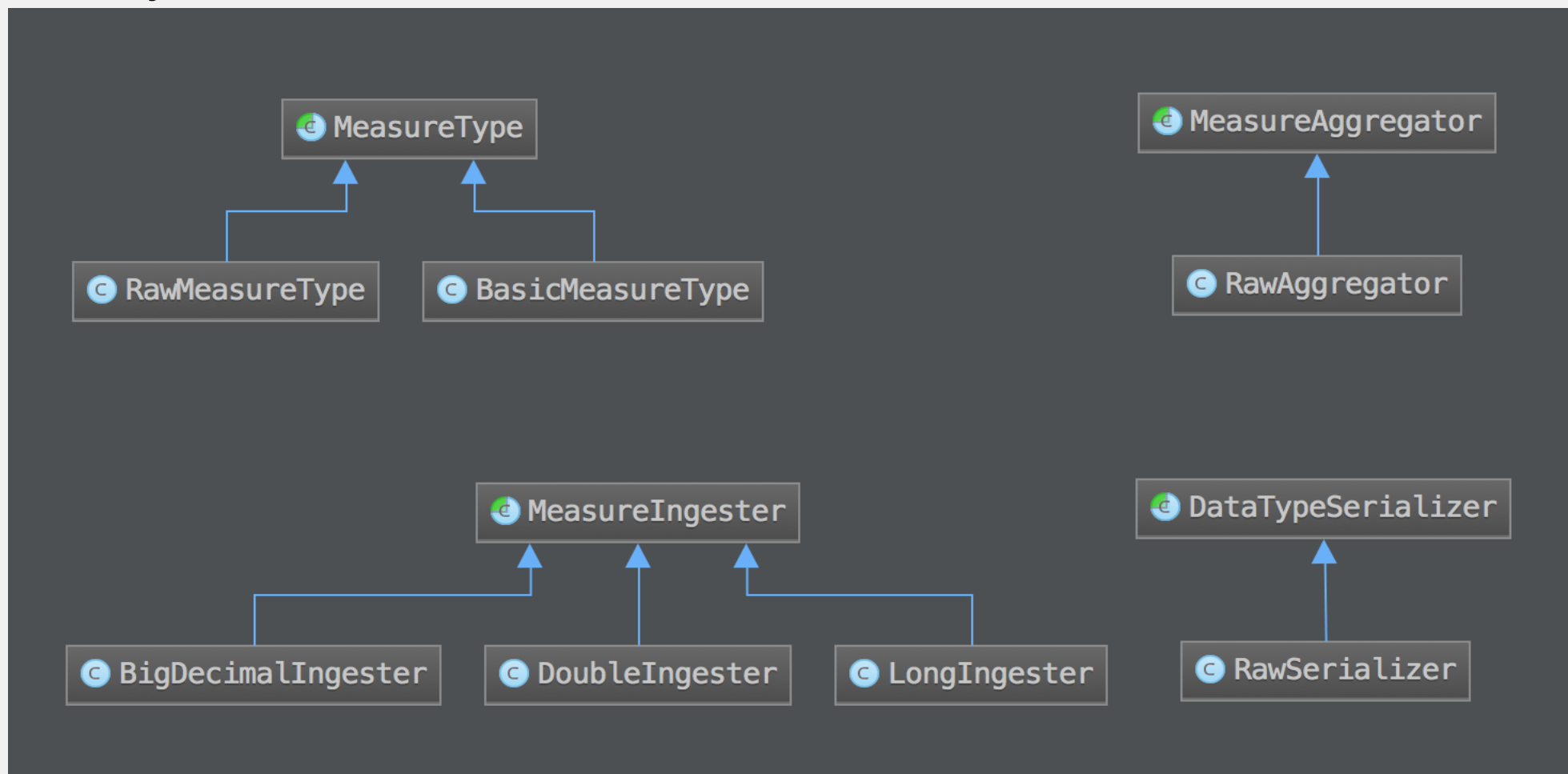
原始明细数据查询改造

Apache Kylin默认只支持聚合查询，即查询中至少包含一个Group by子句。  
通过Apache Kylin获取RAW数据可以有几种实现方案：

- 1、所有需要查询明细的列都在Cube中都创建为维度
- 2、增加唯一列，并作为Cube的一个维度，增加自定义聚合函数，函数返回原值
- 3、Inverted Index
- 4、增加自定义聚合函数，聚合函数能够返回当前key值的所有明细数据

# Apache Kylin@JCloud 二次开发

在Apache Kylin1.5中实现自定义函数



# Apache Kylin@JCloud 二次开发

## 在Apache Kylin1.5中实现自定义函数

### MeasureType

validate(FunctionDesc)	void
isMemoryHungry()	boolean
onlyAggrInBaseCuboid()	boolean
newIngester()	MeasureIngester<T>
newAggregator()	MeasureAggregator<T>
getColumnsNeedDictionary(FunctionDesc)	List<TblColRef>
influenceCapabilityCheck(Collection<TblColRef>, Collection<F	
needRewrite()	boolean
getRewriteCalciteAggrFunctionClass()	Class<?>
adjustSqlDigest(MeasureDesc, SQLDigest)	void
needAdvancedTupleFilling()	boolean
fillTupleSimply(Tuple, int, Object)	void
getAdvancedTupleFiller(FunctionDesc, TupleInfo, Map<TblColRe	

### DataTypeSerializer

peekLength(ByteBuffer)	int
maxLength()	int
getStorageBytesEstimate()	int
valueOf(String)	T

### MeasureAggregator

reset()	void
aggregate(V)	void
getState()	V
getMemBytesEstimate():	

### MeasureIngester

valueOf(String[], MeasureDesc, Map<TblColRef, Dictionary<String>>)	V
reEncodeDictionary(V, MeasureDesc, Map<TblColRef, Dictionary<String>>, Ma	

# Apache Kylin @JCloud 实践经验

# Apache Kylin@JCloud 实践经验

使用Kylin前根据实际环境考虑以下因素：

- 1.模型结构：表关系为星形结构，如果非星形关系需要转换
- 2.环境要求：目前依赖Hadoop、HBase、Hive
- 3.数据源：用Hive命令生成平表，新版本（1.3+）支持beeline
- 4.集群：HBase和Hadoop集群可以是两个独立集群，Kylin可以单独部署集群
- 5.周期调度：需要自行集成周期调度

Hive提前处理：

- 1.维表不能有重复数据，提前做清洗处理
- 2.Cube定义中关联字段类型必须匹配，提前处理Hive模型
- 3.查询中存在复杂的表达式计算，建议提前创建Hive view再基于view建Cube

# Apache Kylin@JCloud 实践经验

安装过程中的典型问题：

1、和社区版Hadoop集成，hadoop-env.sh中读取scheduler jar的问题

```
java.io.FileNotFoundException: $HADOOP_HOME/contrib/capacity-scheduler/*.jar
```

2、和HBase兼容问题

```
java.lang.NoSuchMethodError:
```

```
...hbase.client.Put.getFamilyMap()Ljava/util/NavigableMap;
```

```
...rest.security.MockHTable.put(MockHTable.java:494)
```

3、HCatalog的jar包必须在集群中所有节点存在，否则第二步MR报错

```
java.lang.ClassNotFoundException:
```

```
Class org.apache.hive.hcatalog.mapreduce.HCatInputFormat
```

4、和Yarn兼容问题，需要集成Hadoop2.4+的HBase版本或者调整classpath

```
java.lang.NoSuchMethodError:
```

```
...yarn.conf.YarnConfiguration.getServiceAddressConfKeys(Lorg/apache/hadoop/conf/Configuration;)Ljava/util/List;
```



# Apache Kylin@JCloud 实践经验

## 存储更少：

开启压缩 Hive , HBase

Cube维度优化 Mandatory & Derived & Hierarchy Column

Cube维度优化 Aggregation Group & New Aggregation Group(1.5+)

## Build更快：

维度都使用字典，高基数维度可以选择不使用字典

事实表分区 Increment Build and Merge

并发Job数量--Job Server内存，主要和Build字典相关

MR优化--Hive Flat Table , Build N-Dimension , Convert HFile

## 查询更快：

RowKey顺序--Mandatory维度，Where条件出现频率较多的维度，高基数维度，低基数维度

提高并发速度--Query Server内存，Query Server个数，HBase参数优化

# Apache Kylin@JCloud 实践经验

HBase只做Kylin scan用途参数调整 ( 0.98 )

**hbase.client.scanner.caching** 默认 100 适当调大 可设置为5000

**hfile.block.cache.size** 默认值0.4 适当调大 可设置为0.5

**hbase.regionserver.global.memstore.upperLimit/lowerLimit** 默认值 0.4/0.38 适当调小 可设置为0.3/0.28

**hbase.regionserver.handler.count** 默认值 30 适当调大 可设置为 100

# Apache Kylin@JCloud 实践经验

---

Apache Kylin相关资源：

官方网站：<http://kylin.io>

Mail List：dev@kylin.apache.org , user@kylin.apache.org

Apache Kylin与Pentaho Mondrian集成：

<https://github.com/mustangore/kylin-mondrian-interaction>

京东到家Apache Kylin集成Mondrian测试

<http://gree2.github.io/bi/2016/01/20/kylin-mondrian-saiku-interaction>

# Q&A

---

# 谢谢



关注京东云  
官方微信获  
取更多资讯

