The big data system for eBay Paid Social Ads

黄毅铭
yimhuang@ebay.com

Speaker Profile

- Senior engineering manager for eBay paid internet marketing products
- Lead a team of 8 to deliver two key products for marketing (Feeds; Paid Social)
- Also play a role as system architecture
- Before eBay, I'm a developer & performance engineer in IBM focus on Java EE
- Current interests
  - Big data (Hadoop/Spark/NoSQL) eco-system
  - Applied science and machine learning techniques
  - Philosophy and practices on building a high performance team

Agenda

- eBay paid internet marketing introduction
- eBay personalized Ads on Facebook (paid social)
- System architecture
- Business and data volume
- Technologies
- Case study: Feeds lambda architecture
- Case study: Merging two streams
- Case study: Real time model executions

eBay paid internet marketing (channels)

Paid Search	Paid Social	Display	ePN
Google PLA	Facebook Dynamic Ads	Google GDN	hundreds affiliates
Google Text Ads	Incubations (Pinterest, Snapchat)	Others	
Bing PLA			

eBay paid internet marketing (function layers)

Campaign Orchestration

- Channel campaign management
- Cross channel optimizations

Targeting and optimization

- Personalized targeting / recommendation
- Bidding algorithms
- Audience segmentation and ID linking

Contents, Creatives and Rendering

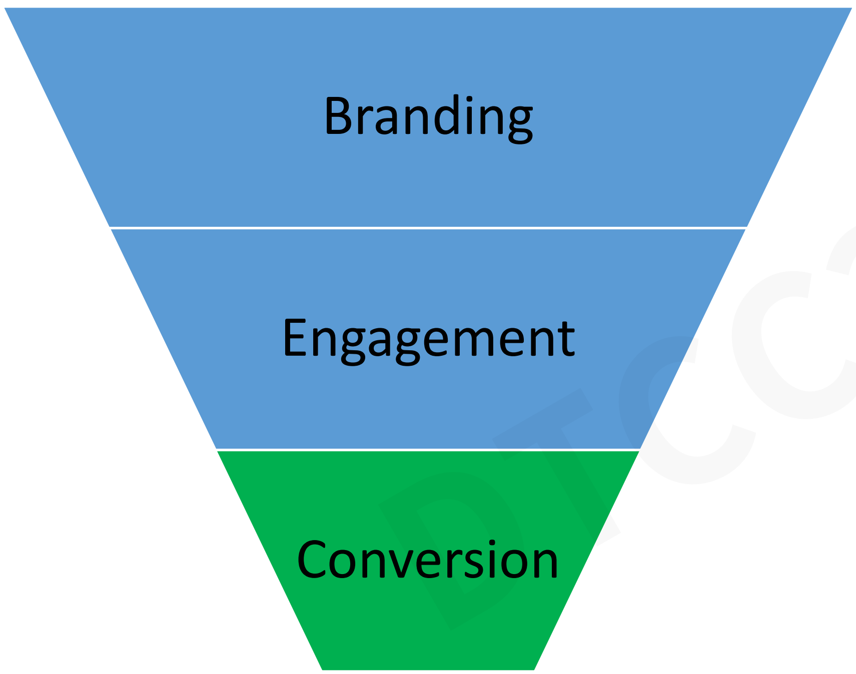
- Feeds listing platform
- Creative media studio
- Ads rendering server

Common data layers and foundations

- Marketing data marts
- Tracking & Attributions
- Experimentation platform

eBay paid social business

Facebook Funnels

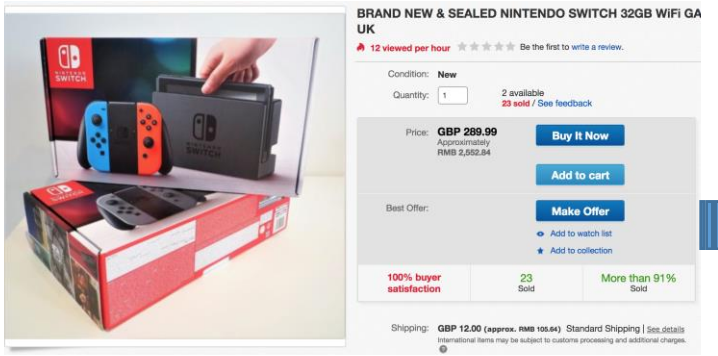


- Reach and convert existing or potential buyers
- Using attractive and personalized contents
- Run the business profitably at scale

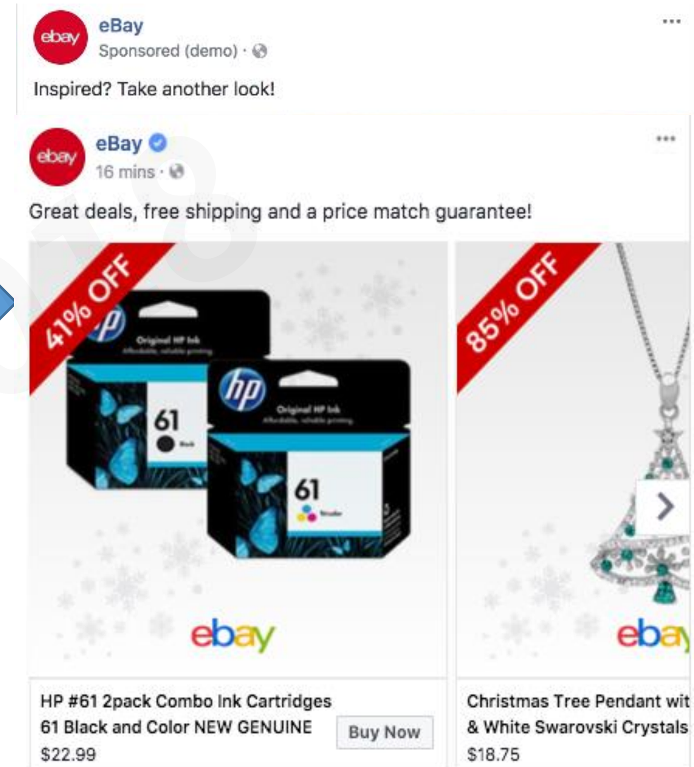
Facebook Dynamic Ads Experiences

See relevant Ads on Facebook

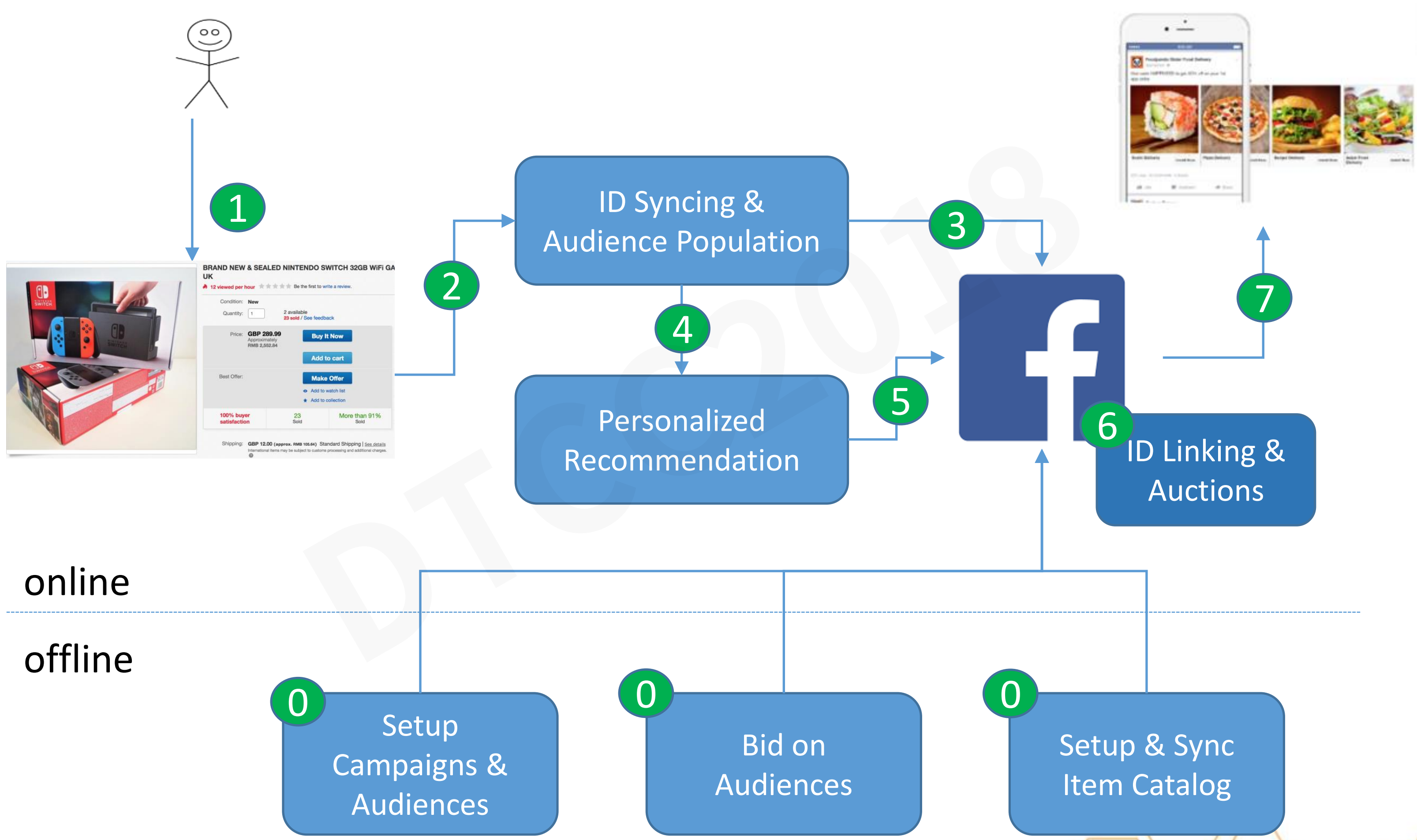
View an item on eBay



3m latency



Facebook Dynamic Ads Architecture



Business and system volume

Profitable Business (2017)

- Hundreds of millions $ business
- Double digit ROI
- Growing (double digit growth)

Scaled Big Data System

- ~40TB daily data processing for 1B Item listings
- ~3 min system latency for retargeting
- ~200M daily recommendations
- ~10K/s QPS for databases

Technologies

Monitor and Alert	• InfluxDB + Grafana • ELK stack
Online Stream Applications	• Kafka + Akka • Kafka + Spark Stream
Databases	• NoSQL: Cassandra + Redis • MySQL
ETL and offline data processing	• Spark RDD + Spark SQL • Teradata

Learnings on Architecture and Choosing proper Tech

No single point of failure

- We favor distributed and highly available components / middle-wares
- We have plans for single node services

Be Resilient

- Tolerate the dependent service failures
- Graceful downgrade for non critical modules
- Retries

Learnings on Architecture and Choosing proper Tech

Ground solid sub systems
In the whole system, we need some components that "never" fail

Monitor and Alerts

- Early detection + facilitate debugging
- Four layers of monitors in the system

Case #1: Learnings on Feeds Lambda architecture

Time	3/1 0	3/1 6	3/1 12	3/1 18	3/2 0	3/2 6	3/2 12	3/2 18	3/3 0
Batch	3/1				3/2				
Speed									
Merge				Query			Query		

Lambda Architecture 101

- Daily batch processing to come out summary view with latency
- Speed (NRT) layer to provide small amount of NRT data w/o latency
- Merge layer to combine the two sources and provide a single, merged view
- Benefits
  - Near real time data accuracy
  - Daily remediation for speed layer data

Case #1: Learnings on Feeds Lambda architecture

Feeds platform

- Create or update item listings in a daily basic
- NRT updates on item prices
- NRT listing for new items
- NRT delete for expired / sold out items
- Hourly updates for item tags (deals, trending, etc…)

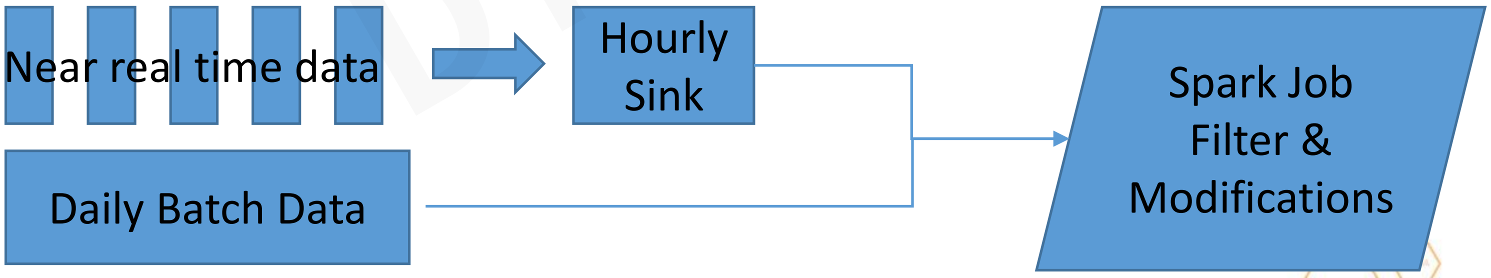
Challenges

- Need to apply item filtering and modification logic in two processes
- Well handling on batch data latency (eg. old price from batch; expired items in batch; etc…)

Case #1: Learnings on Feeds Lambda architecture

Solution to filters/modifications

- Original design
  - Storm for NRT processing
  - Spark for Batch processing
  - Shared filter/modification logic by code pieces (lib)
  - Duplicate efforts on formatting data schema
- New design
  - Retire storm and use hourly Spark job (acceptable latency)
  - From re-using code pieces to re-using components (spark job)



Case #1: Learnings on Feeds Lambda architecture

Solution to “merge” data from different sources

- Merge layer before item synchronization
- Row updates by PK+CK
- Row merge by PK

PK	CK	CF1 (fields)	CF2 (price)	CF3 (del)	Timestamp
123	batch	xxx	3		12:00:00
123	update		4		13:00:00
123	delete			true	13:05:00

Item 123

merge

Item 123 should be removed

Batch Start
@12:30:00

Batch End
@20:30:00

Case #2: A solution to merge two streams

Problem statement

- Two streams of events with the same key, but different value fields
- Need to join/merge the two streams for the events on the same key and fill in all the required fields coming from two streams to create a new stream
- No guarantee of which event will come first, and could have duplicate events

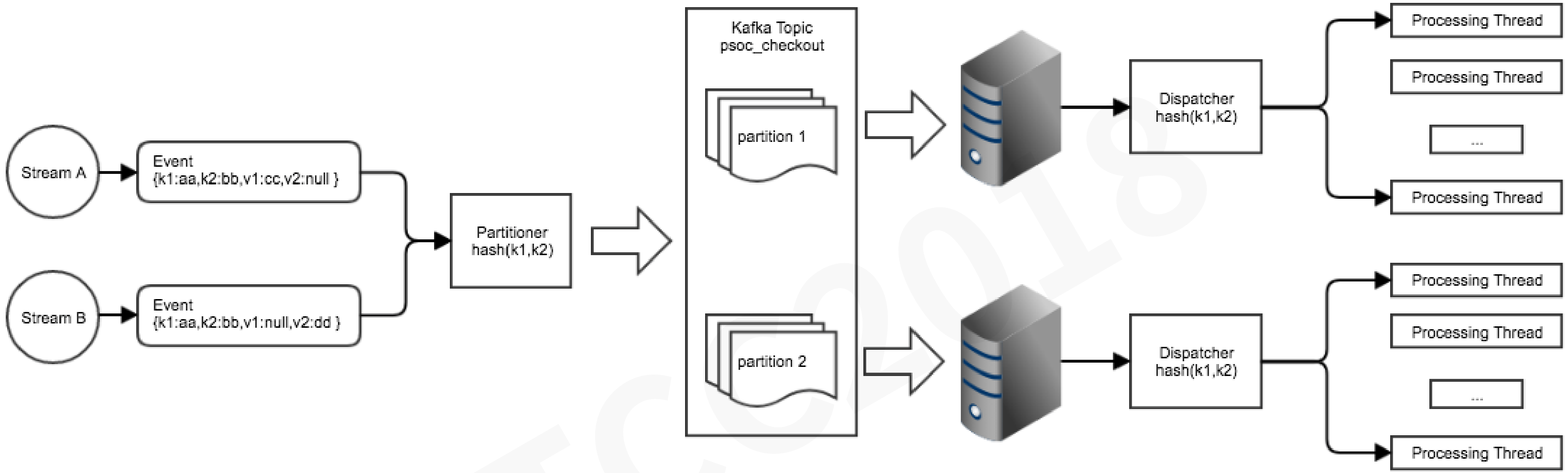
A naïve solution

- Processing the two streams and sink them to a message queue (Kafka)
- Maintain a database (kv store) and upsert the event based on key
- Grab events that have all the required fields and send them out

Key considerations to the solution

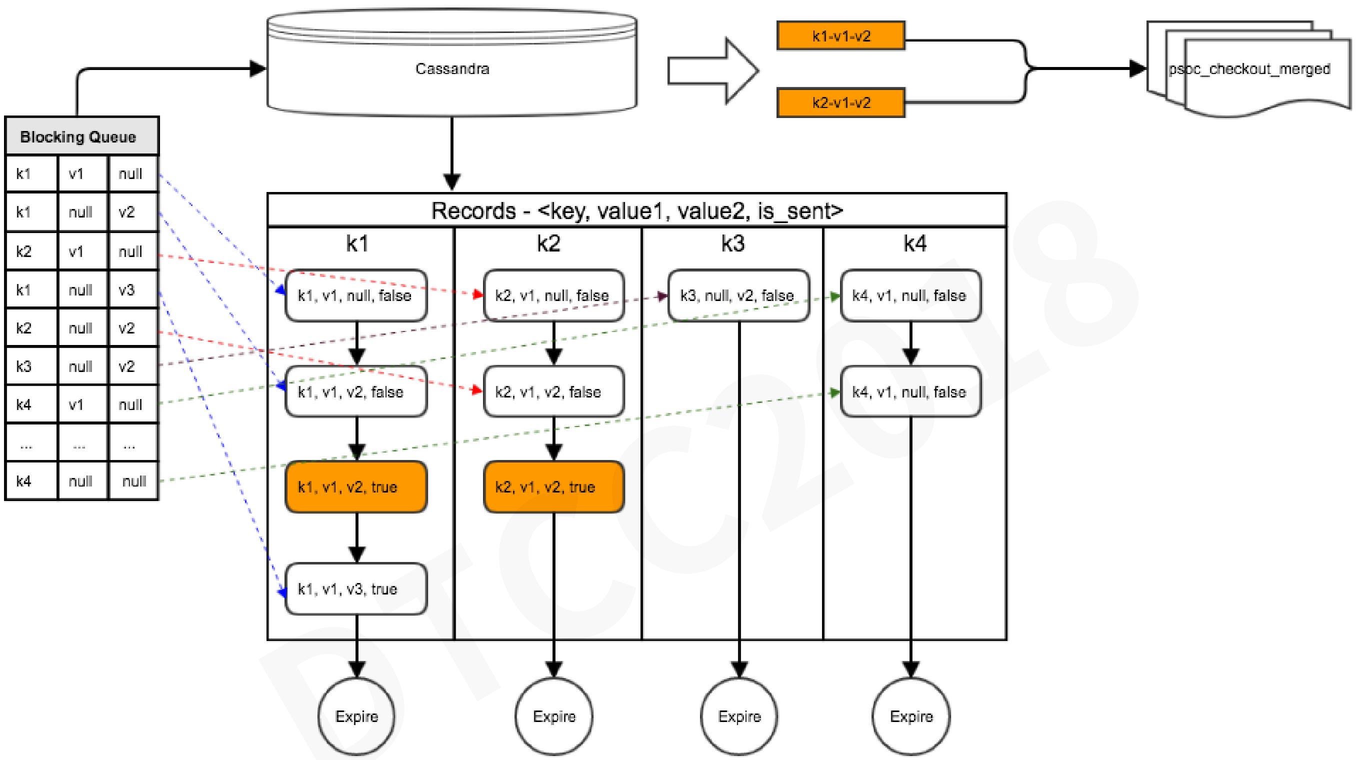
- Need to be a multi-threading App due to the high concurrency requirements
- Update + Query are two actions that could cause “dirty read” in multi-threading environment

Case #2: A solution to merge two streams



- Keep the events that have the same keys to the same partition
- Keep the events that have the same keys to the same thread
- No coupling / restriction for the # of consumers and Kafka partition number

Case #2: A solution to merge two streams



Things happened in a specific thread

- Upsert + Query for each event in a transaction (batch in Cassandra)
- Mark “sent” flag once data fields are complete
- Use Cassandra TTL to implement “caching window”

Case #3: Real time model executions

Recommendation / Personalized Ads backgrounds

- Recommend items that an user is most likely to purchase
- Recommend similar but more appealing items
- Try to capture user recent interests and promoted items (deals)
- Key algorithms
    - Ranked Recently Viewed Items (RRVI)
    - Similar items
    - Rule based massage on Last View Items (LVI), Deals and model results

Problem statement

- The online recommendation process is a NRT pipeline, which needs all communications between components to be service (real time) based
- Need real time resolution for the recommendation list for a given user
- Need real time bid adjustment for a given user

Case #3: Real time model executions

Solution #1 (use case: bid adjustment)

- Daily offline model predictions
- Synchronize & maintain a bidding cache keyed by user
- Real time look up in the App for user level bid adjustment
- Trade-offs: acceptable latency as user monetization powers won’t change frequently

Solution #2 (use case: RRVI model prediction)

- Real time updated feature cache
    - 7 day’s key user behavior events
- Batch updated feature cache
    - Statistic metrics based on items or users
- Real time fetch for features + distributed model predictions
- Limitations
    - Only cache the dominant features
    - Simple models
    - 10K level calls per second with decent amount of machines

Beyond CAP theory, building commercial big data system is all trade-offs between latency, consistency, accuracy, tolerable failures and the resources you have.

THANKS


HANA
Hadoop
SQL
MySQL
DB2
BigData
Spark
SQL Server
Oracle
NewSQL
MPP
NoSQL
OLAP

讲师申请

联系电话（微信号）：18612470168

关注“ITPUB”更多
技术干货等你来拿~

与百度外卖、京东、魅族等先后合作系列分享活动

微学堂

让学习更简单

微学堂是以ChinaUnix、ITPUB所组建的微信群为载体，定期邀请嘉宾对热点话题、技术难题、新产品发布等进行移动端的在线直播活动。

截至目前，累计举办活动期数60+，参与人次40000+。

ITPUB学院

ITPUB学院是盛拓传媒IT168企业事业部（ITPUB）旗下
企业级在线学习咨询平台
历经18年技术社区平台发展
汇聚5000万技术用户
紧随企业一线IT技术需求
打造全方式技术培训与技术咨询服务
提供包括企业应用方案培训咨询（包括企业内训）
个人实战技能培训（包括认证培训）
在内的全方位IT技术培训咨询服务

ITPUB学院讲师均来自于企业
一些工程师、架构师、技术经理和CTO
大会演讲专家1800+
社区版主和博客专家500+

培训特色

无限次免费播放
随时随地在线观看
碎片化时间集中学习
聚焦知识点详细解读
讲师在线答疑
强大的技术人脉圈

八大课程体系

基础架构设计与建设
大数据平台
应用架构设计与开发
系统运维与数据库
传统企业数字化转型
人工智能
区块链
移动开发与SEO

联系我们

联系人：黄老师
电　话：010-59127187
邮　箱：edu@itpub.net
网　址：edu.itpub.net
培训微信号：18500940168