Hitchhiker's Guide to Building A Data Science Platform

In-Memory Computing Summit
June 2015, San Francisco CA

David Abercrombie, Tapjoy

Tapjoy® | REWARDING MOBILE

You

- Decision Makers
  - Purchase
  - Design
  - POC

- Want to get a head start
  - Some familiarity
  - Curious about current practice

Me

- Database person
  - Oracle since 1996
  - HP Vertica since 2010

- Data structure design & ERD
- SQL tuning
- Execution plans
- Application interactions
- Scalability and Capacity
- Diagnostics

- Most recently: data warehouse & BI

Presentation Approach

- Tie data science to business needs
- Describe data pipeline and storage
  - Not the modeling pipelines
  - Evolution and growth
  - New technologies
- Relate data science needs to tool selection
- Warn about unexpected problems

Tapjoy

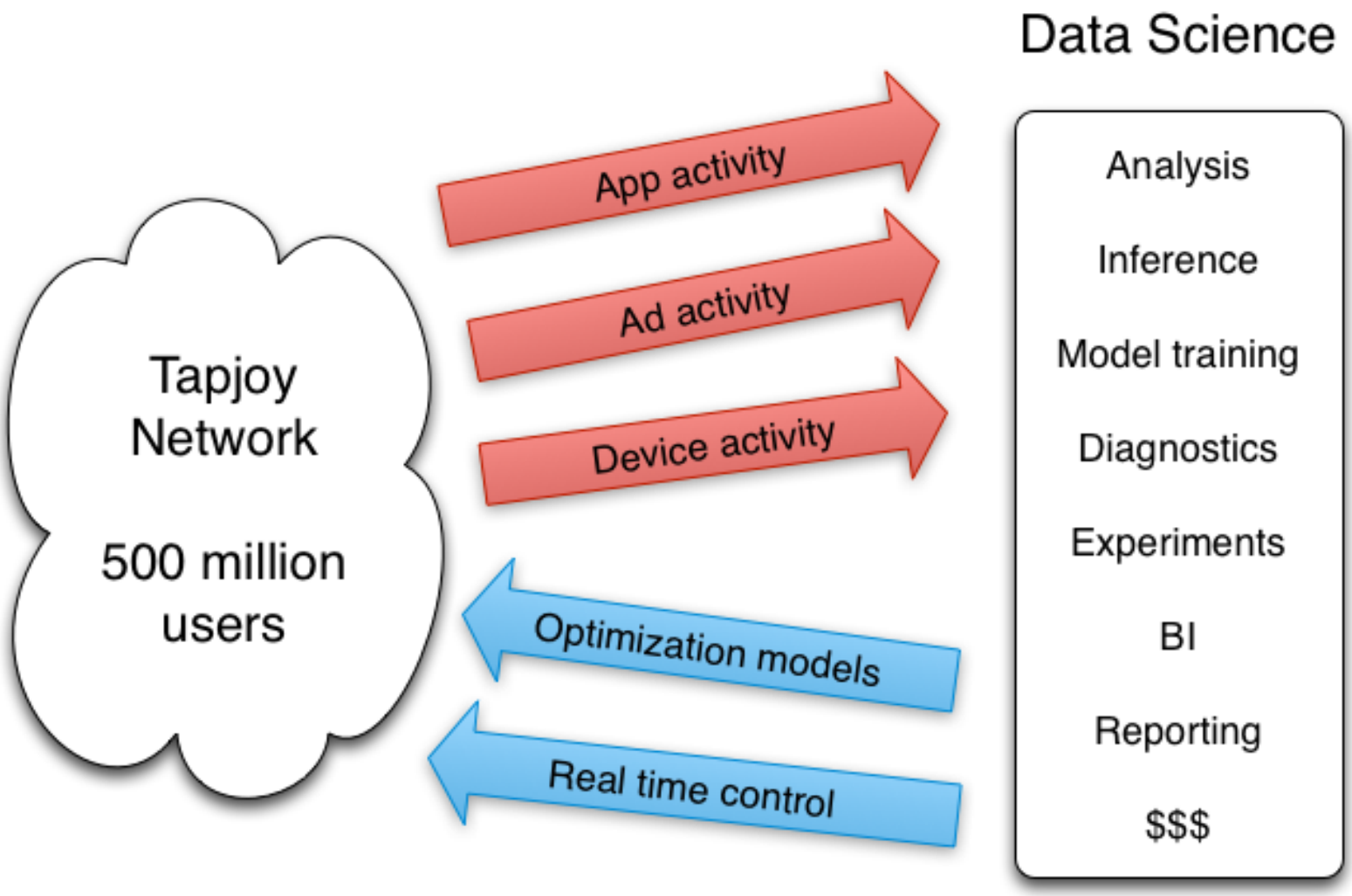
- **Rewarded mobile advertising**
  - Publishers monetize
  - Advertisers get message out, new users
  - Mobile users get rewards
  - Similar to web analytics (funnel, etc.)

- Over 500 million global users
- Thousands of active apps and ads
- Millions transactions/min, billions/day

Data Science Team

- Machine learning & predictive models
- ETL pipeline
- Business Intelligence
- Analysts

Data Driven Decisions



Many data systems

- RDBMS: PostgreSQL, MySQL, RDS
- HDFS: Cloudera, Hive, Pig, Hbase
- Key-value: SimpleDB and Riak
- BI: HP Vertica and MicroStrategy Cloud
- Exploratory: Tableau, dashboards, ad hoc
- Real-time: Spark streaming and **MemSQL**
- Archive: S3, Glacier
- OLTP metadata: MySQL RDS, Memcache
- Message queuing: Riak, Rabbit MQ, SQS, Kafka
- ETL: simple custom framework
- Algorithms: too many to mention

Lambda Architecture

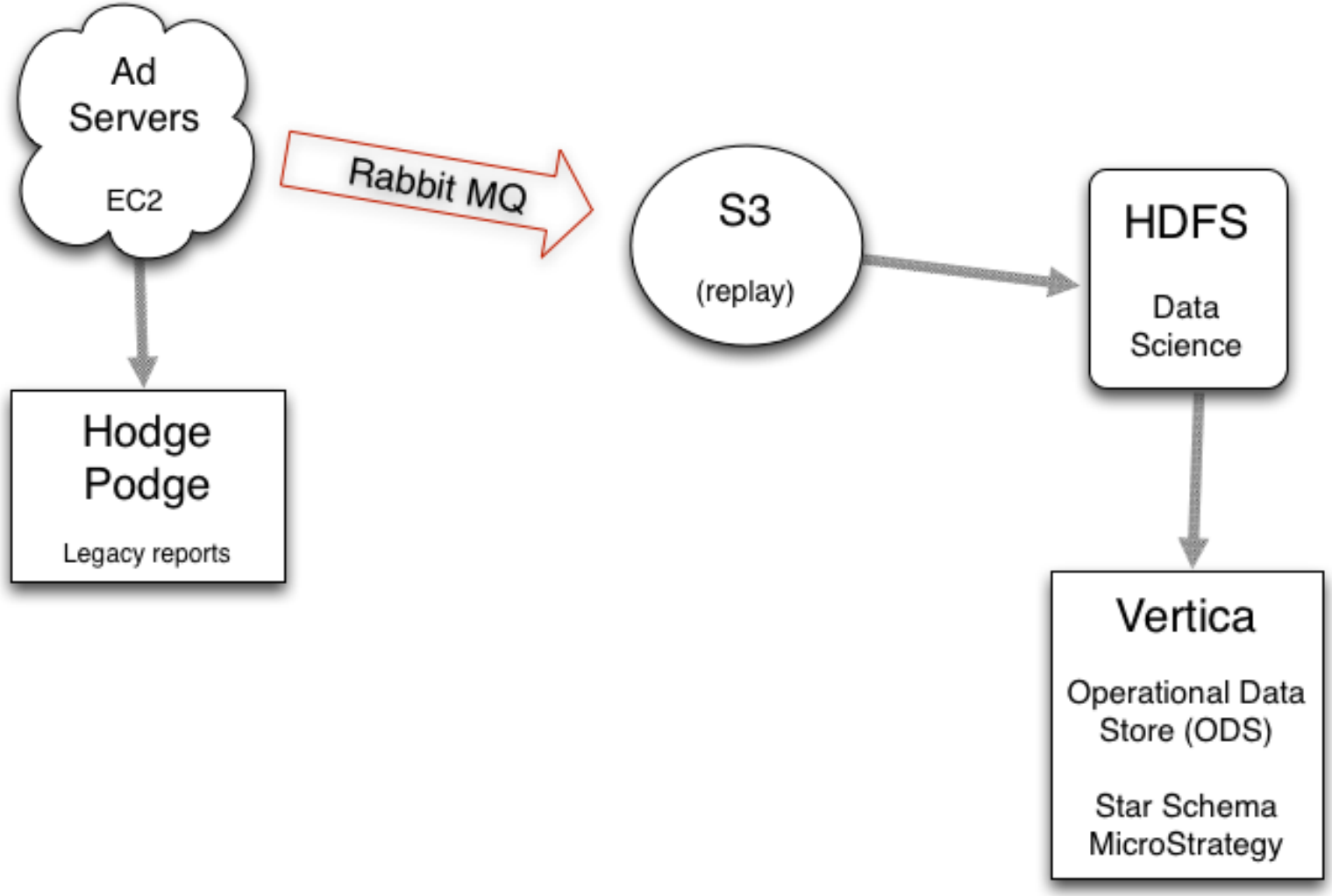
Append-only	Streaming
Long term	Short term
Timestamps	Timestamps
Batch processing	Minimal processing
System of record	
HDFS, HP Vertica	*Spark, MemSQL*

Contrast with legacy twinkling OLTP in well normalized relational data model.

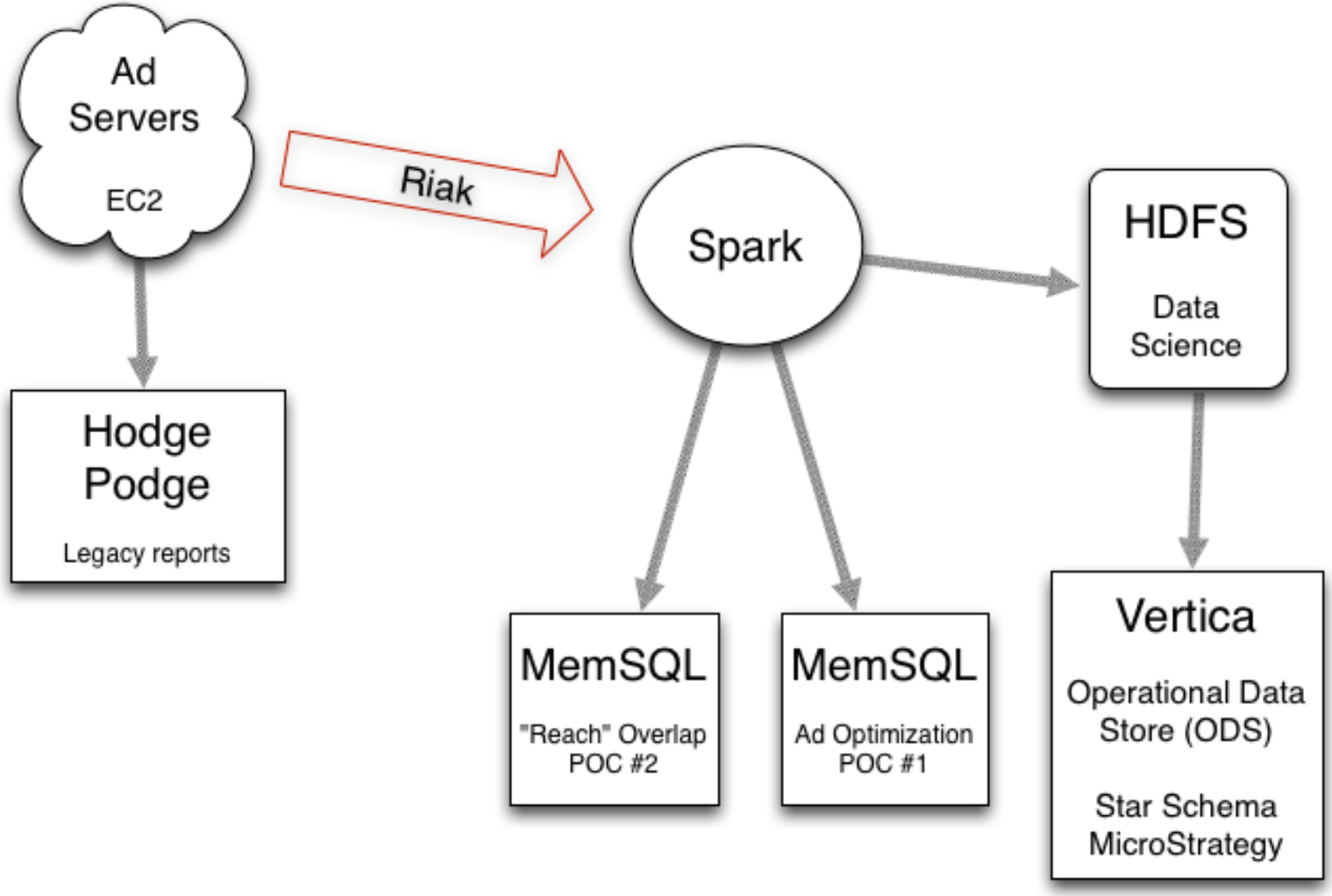
Well suited for clickstream analysis.

Data pipelines

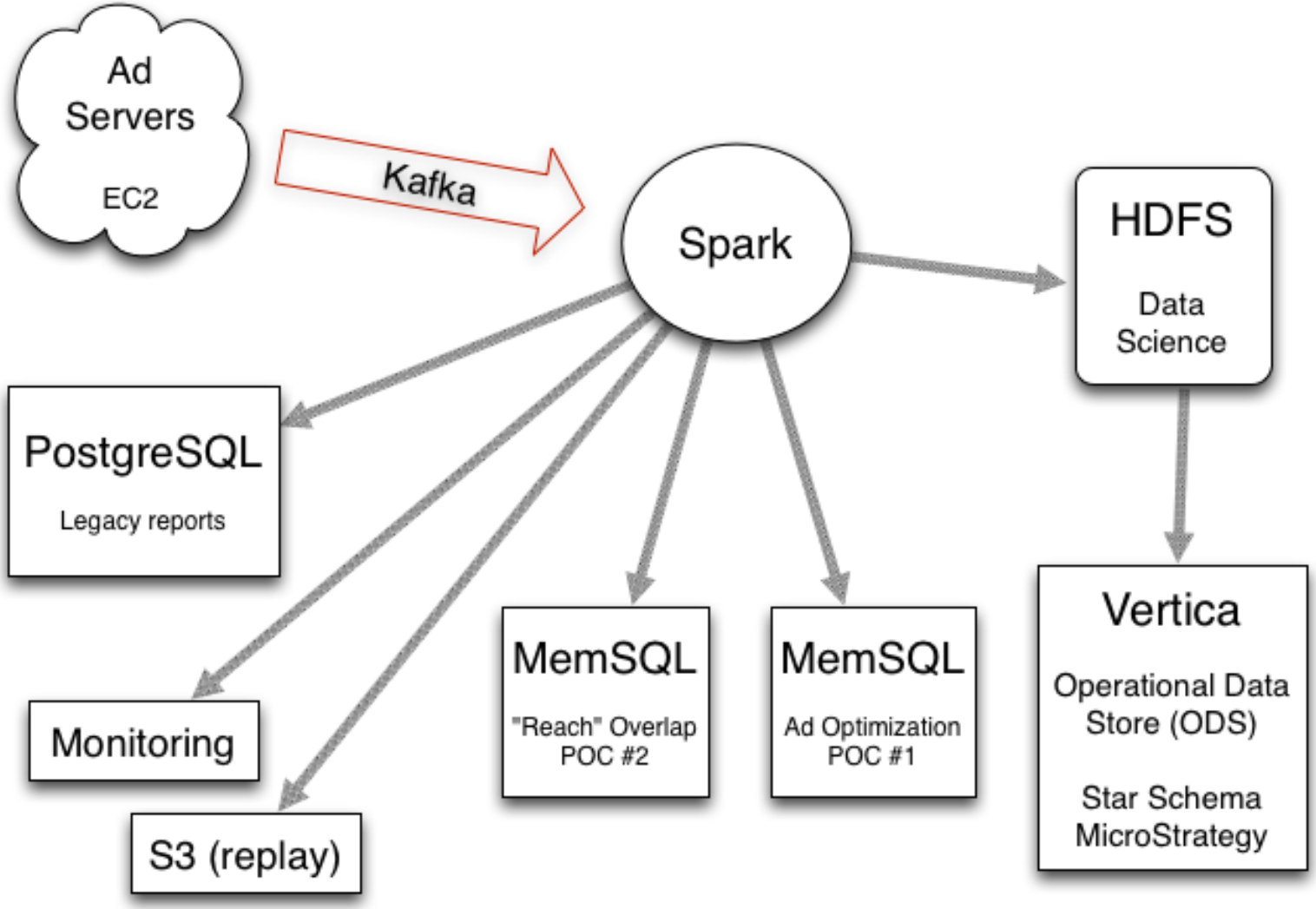
Version 1 – Four years ago



Version 2 – Spark and MemSQL today



Version 3 – All roads will lead from Spark



Tapjoy & MemSQL Use Cases

Tapjoy and MemSQL Use Cases

- Real time add optimization
  - Millisecond decision making
  - Low latency data – ten seconds
  - Aggregation and primary key lookup
- Overlap analysis
  - Ad targeting
  - Estimate size of audience
  - High dimensionality – personas and demography
  - Exploratory

Use Case 1 - Real-time ad optimization

- API returns an ordered list of ads to show a user

- Rules
  - Ads that are performing well right now, etc.
  - Except ads that user has already seen recently

- Two inline views, one for each rule
- Left join, looking for nulls on right side (anti-join)

- Replaced HBase
  - High performance
  - Easy SQL
  - No ETL, no latency

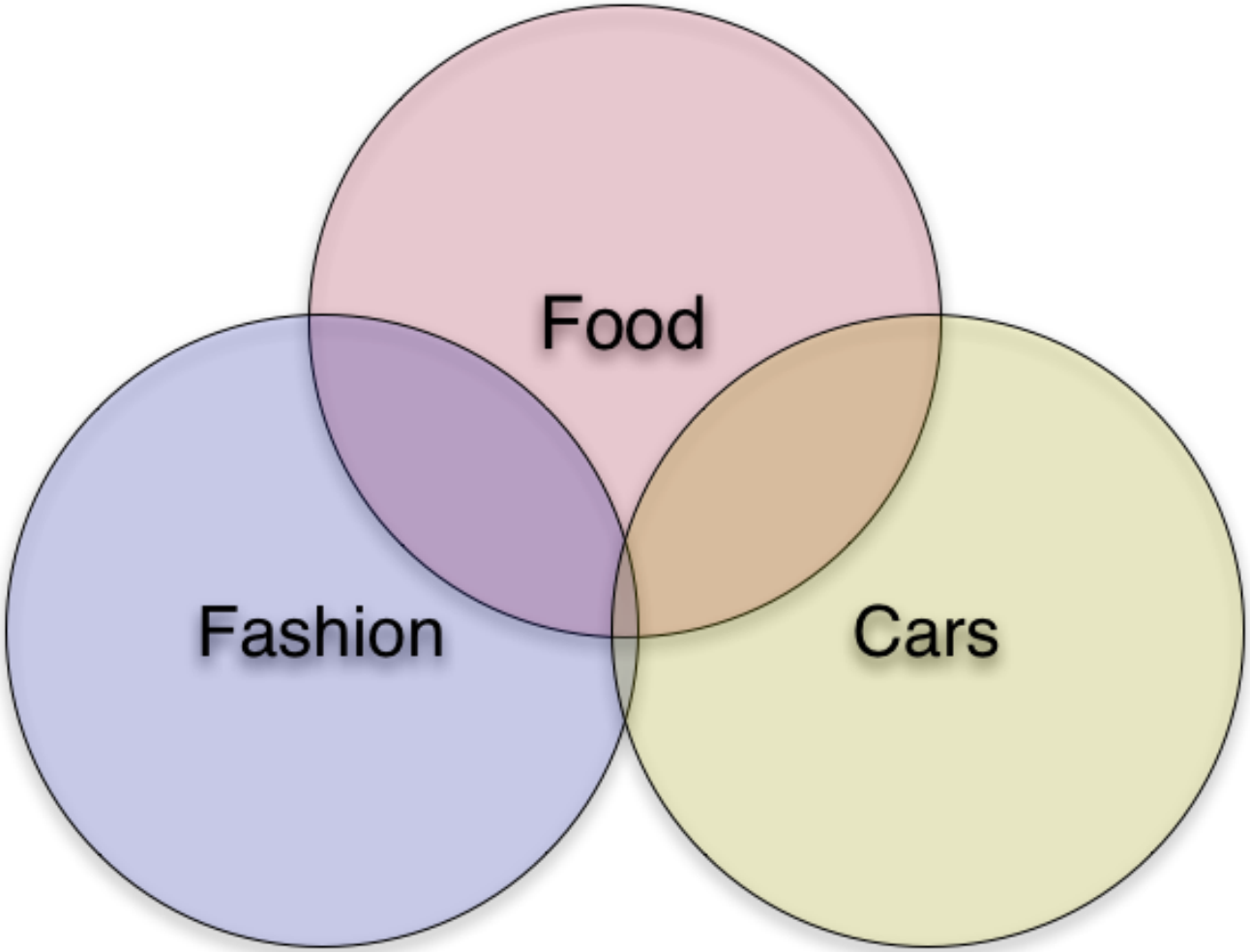
Use Case 1 - Real-time ad optimization results

- Eight nodes
- Very stable

- Throughput: 60,000 queries/second
- Response time: <10 milliseconds
- Includes aggregation over 1 day

- Building new cluster for 30-day aggregations

Use Case 2 – Overlap Analysis



Use Case 2 – Overlap Analysis

- Exploratory data analysis via dashboard
- Calculate size of targeted audience (ad ops)
- Fine tune targeting, predict activity

- Find overlap between
  - User "personas" (e.g. "Gamer", "Sports Fan," "Mom")
  - User demography (age, gender, income)
  - Geography
  - Recent user activity (real-time data stream)

- Dimensionality too high to pre-compute
- Boost Conversions with Overlap Ad Targeting

Business Impacts of MemSQL Use Cases

- Real-time ad optimization
  - **Easier to fine tune rules (SQL)**
  - Simpler ETL
  - Extraordinary performance

- Overlap targeting analysis
  - Handles complexity that is infeasible to pre-compute
  - Eliminates pre-computing
  - **Easy to use (SQL)**
  - Extraordinary performance

Coming to terms with HTAP

What is HTAP?

- Hybrid Transaction / Analytical Processing
  - Online transaction processing (OLTP) and
  - Online analytical processing (OLAP)

- Simplifies data transfer
- Analytics can rely upon freshest data

- Paradigm shift

Ingestion not enough

- Integration
- Usability

- Hybrid Transactional and Analytic Processing (HTAP)
  - High volume, high velocity, low latency ingestion
  - SQL interface (expressive, simple, universal)
  - Reduces ETL and pipeline complexity
    - No need to pre-aggregate, fewer systems
  - High request rate, fast response time

Ugly keys and precomputed data

- Hbase Key-value format
  - Pre-aggregate all possible key combinations
  - Construct a key to express query

- Example 1
  - How many users in California?
  - Key: **US-CA-$-$-$-$-$-$-$-$-$**
- Example 2
  - How many "offerwall" users in California?
  - Key: **US-CA-$-$-$-$-OFFERWALL-$-$-$-$**

E.F. Codd 1970

Future users of large data banks must be protected from having to know how the data is stored in the machine.

The relational model of data appears to be superior in several respects ...

... maximal independence between programs and machine representations ...

E.F. Codd. 1970. A Relational Model of Data for Large Shared Data Banks *Commun.* ACM 13, 6 (June 1970), 377-387

MemSQL HTAP benefits

- **SQL!**
- Updates and deletes!
- Use standard tools and APIs

- Rethink use cases
- Combine transactions and analysis in one system
- Very high throughput and low latency
- Simplify ETL

Rethink data structure design

- Data structure design is key for success at scale
  - True of all data systems, by the way!
- Minimizing disk IO is no longer the main goal
  - Design for selectivity, rather than disk compression
  - A hard habit to break!
- Skills are transferable from other databases
  - An Oracle expert can quickly master MemSQL

Data accuracy

- Algorithmic work more tolerant of data errors,
- BI needs high accuracy

- Doing both in HTAP requires tricky balancing acts

- Needs vary among data science team members

Metadata

- Traditional BI analysis requires metadata (**lookup**)
  - **Dimensions** in a star schema

- API-based real-time applications can use IDs only
  - Metadata ETL is a hassle
  - Metadata data integrity must be pristine
  - Metadata not in clickstream fire hose
  - Data engineers rarely appreciate metadata

- Do not forget metadata if you want to analyze!

Semantics and Instrumentation

- What do those beacons mean?

- Often overlooked
- Tricky, subtle, complicated
- Cannot be left to developers alone
- Gap between engineering and business
- Needs ownership

- Bug Amplifier

Lessons

Lessons

- **Don't Panic!**
- SQL is expressive!
- Data structure engineering required
  - Legacy database skills are transferable
- Do not neglect semantics, metadata, and accuracy
- HTAP works

David.Abercrombie@tapjoy.com