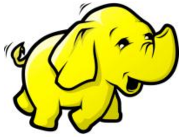
AMPLab
Retreat

Real-Time Analytical Processing (RTAP) Using Spark and Shark

Jason Dai
Engineering Director & Principal Engineer
Intel Software and Services Group

Why Real-Time Analytical Processing (RTAP)?

Big data in large web sites

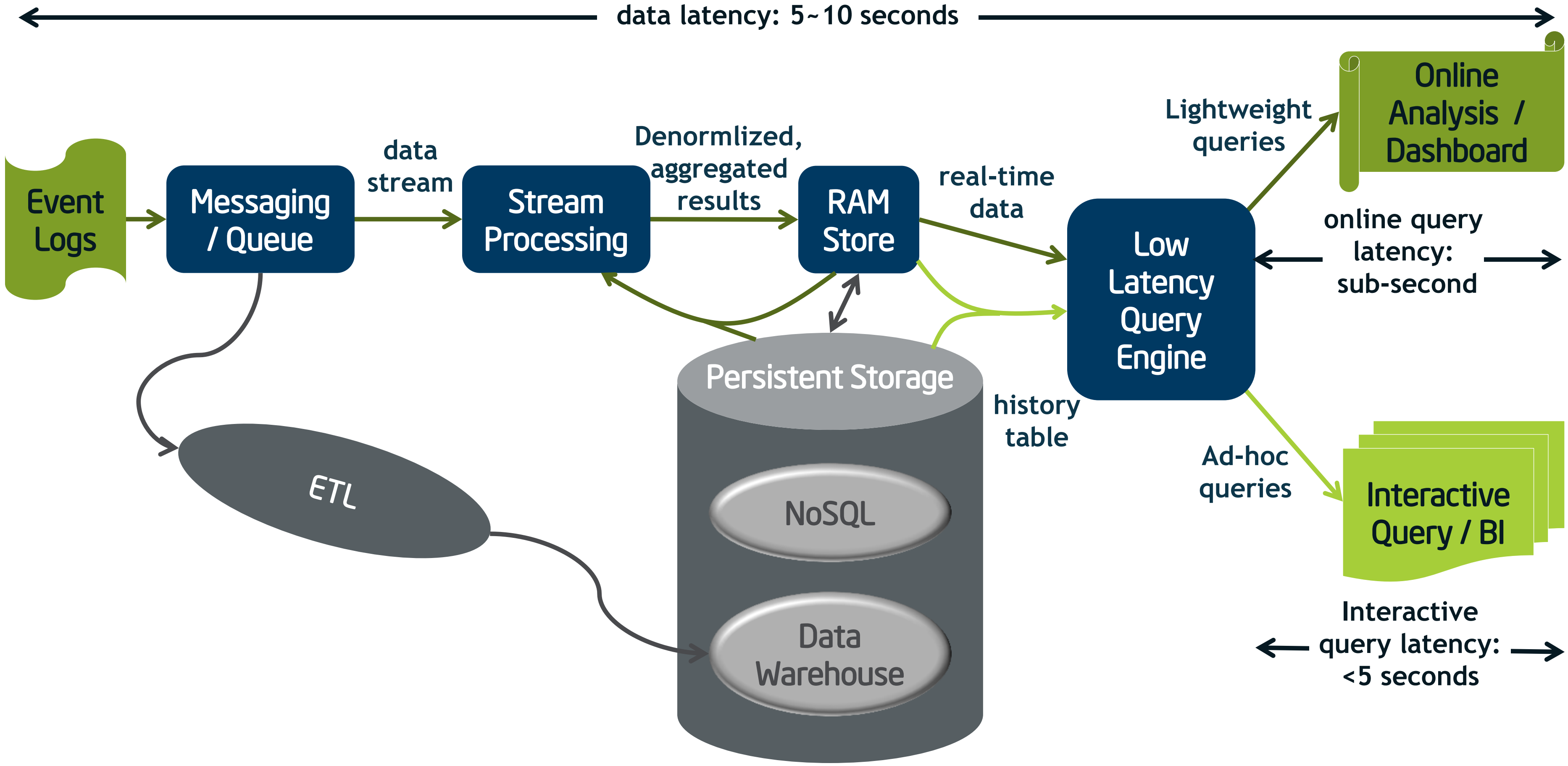
- The  is in the room

Moving beyond the elephant

- Discover and explore data iteratively and interactively for real-time insights
- *RTAP: Real-Time Analytical Processing*
  - Data continuously streamed in & processed in near real-time
  - Real-time data queried and presented in an online fashion
  - Real-time and history data combined and mined interactively
  - Predominantly RAM-based processing

We are partnering with several web sites in China
on building the *RTAP* framework using Spark & Shark

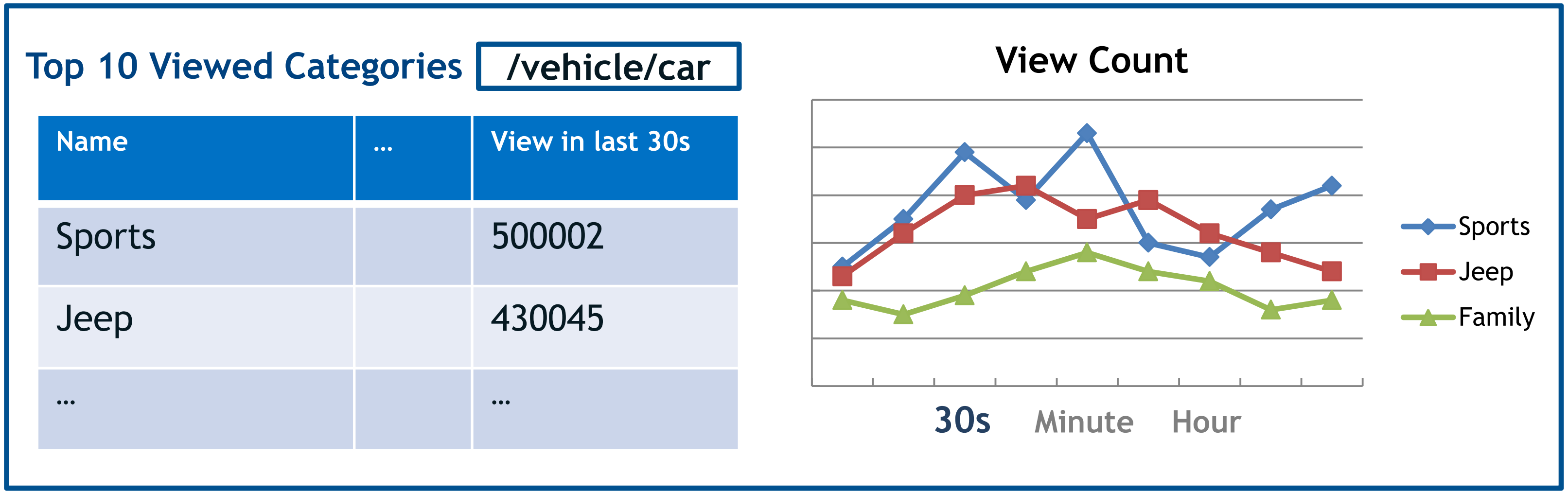
RTAP Architecture



RTAP Use Cases

Online dashboard

- Pages/Ads/Videos/Items — time base aggregations — break-down by categories/demography



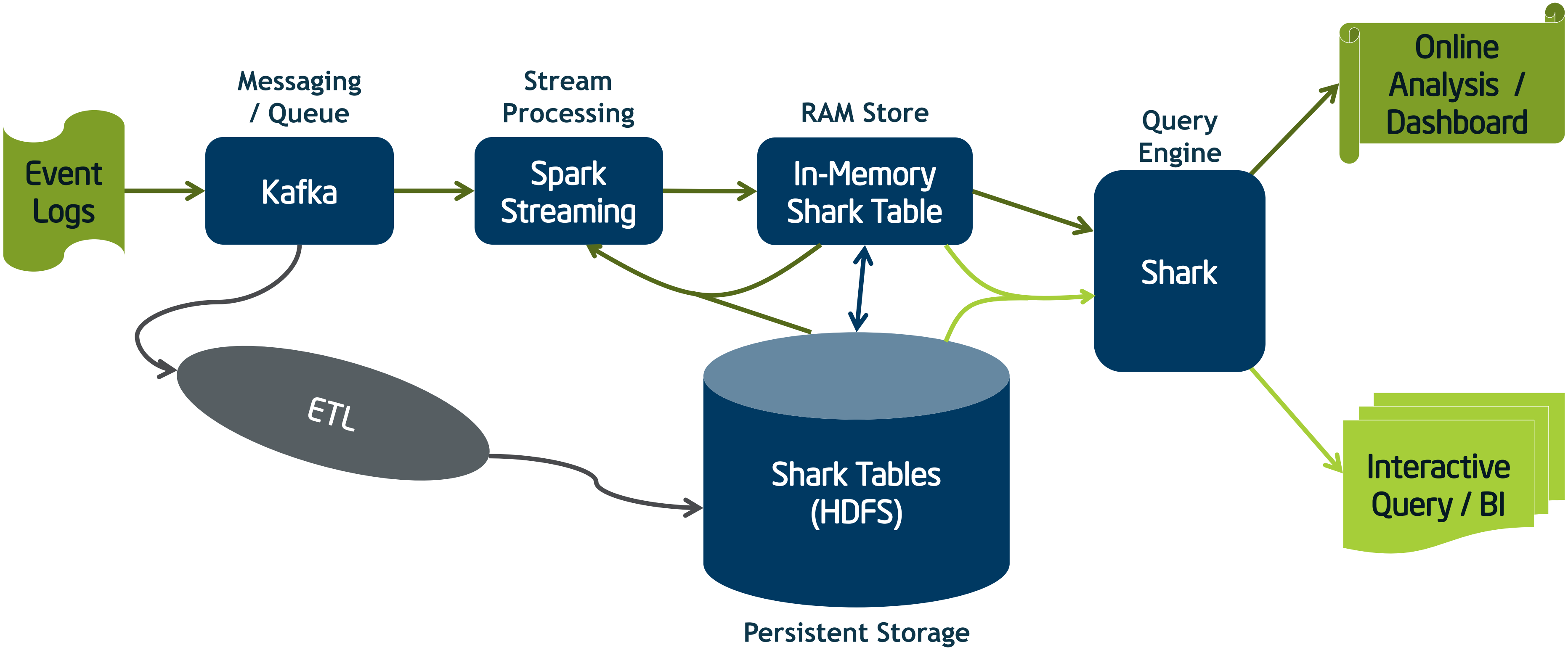
Top 10 Viewed Categories /vehicle/car

Name	…	View in last 30s
Sports		500002
Jeep		430045
…		…

Interactive BI

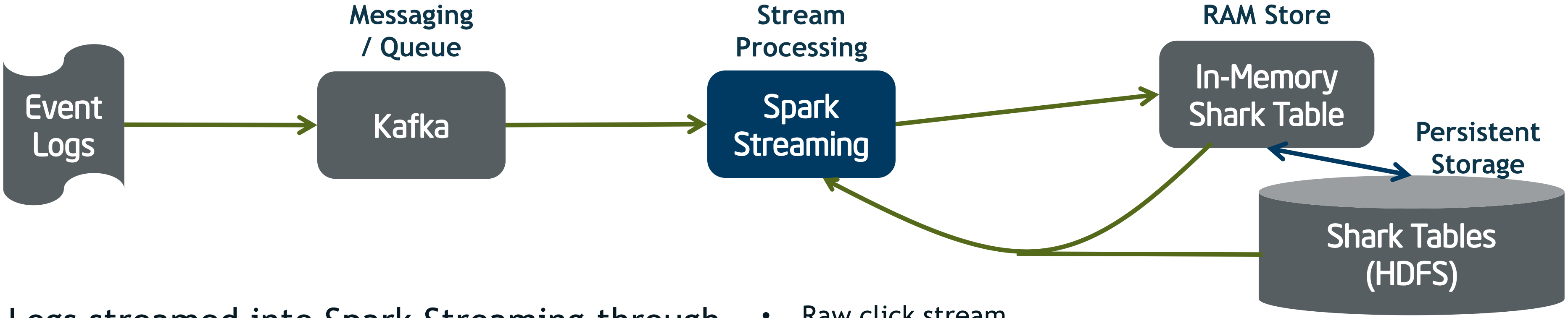
- Combined with history & dimension data when necessary
  - E.g., top 100 viewed videos under each category in the last month

RTAP Framework using Spark & Shark



A work in progress

Real-Time Data Stream Processing



Logs streamed into Spark Streaming through Kafka in real-time

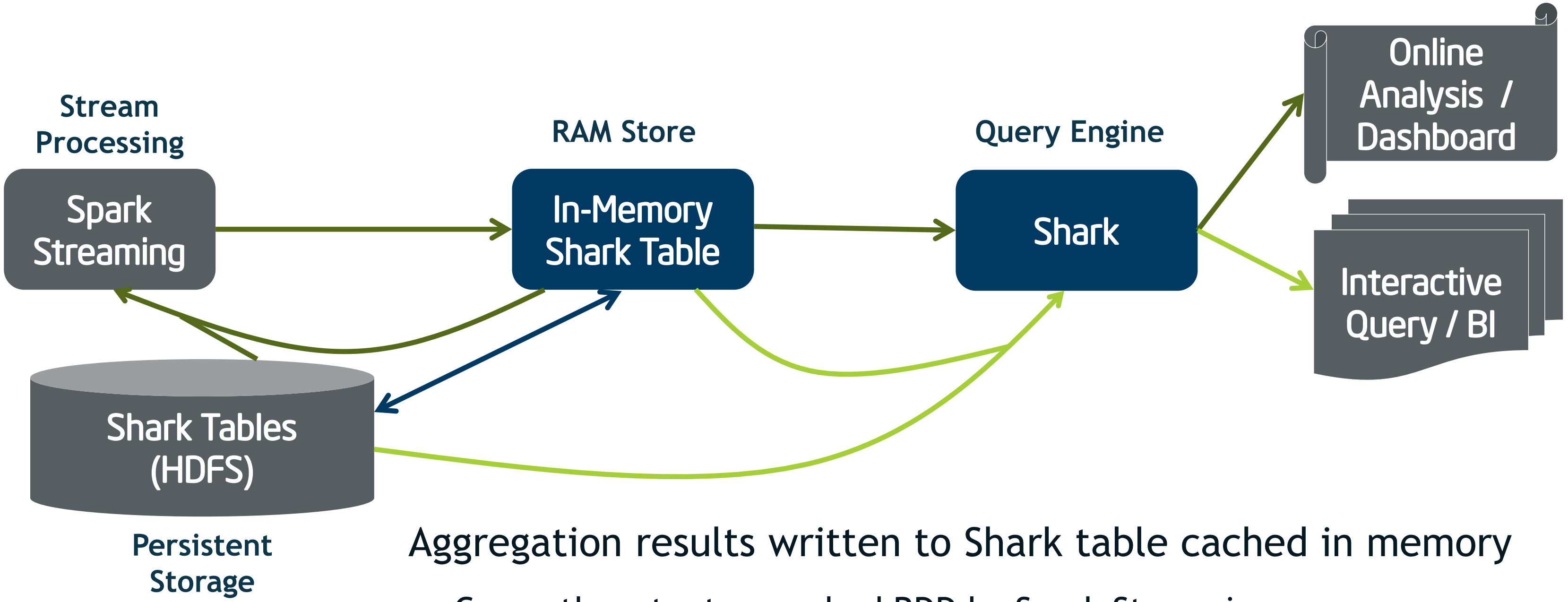
Incoming logs processed by Spark Streaming in small batches (e.g., 10 seconds)

- Compute multiple aggregations over logs received in the last window (e.g., 1 minute)
- Join logs and history tables when necessary

Plan to add the streaming support directly in Shark

- Raw click stream
  - *0.6.38.68 - - BAF42487E0C7076CE576FAAB0E1852EC [14/Dec/2012 8:21:16 -0] "GET ?video=8745 HTTP/1.1" 101 1345 http://www.foo.com/bar/?ivideo=8745 "Mozilla/4.0 (compatible; MSIE 5.5; Windows 98; Win 9x 4.90)“*
- Compute page view in the last minute
  - E.g., *www.foo.com/bar/?video=8745, www.foo.com/bar, www.foo.com,* etc.
- Compute category view count in the last minute
  - E.g., join logs and the video table (assuming *video 8745* belongs to */vehicle/car/sports*) for */vehicle, /vehicle/car, /vehicle/car/sports*, etc.

Real-Time Data Store and Query Engine



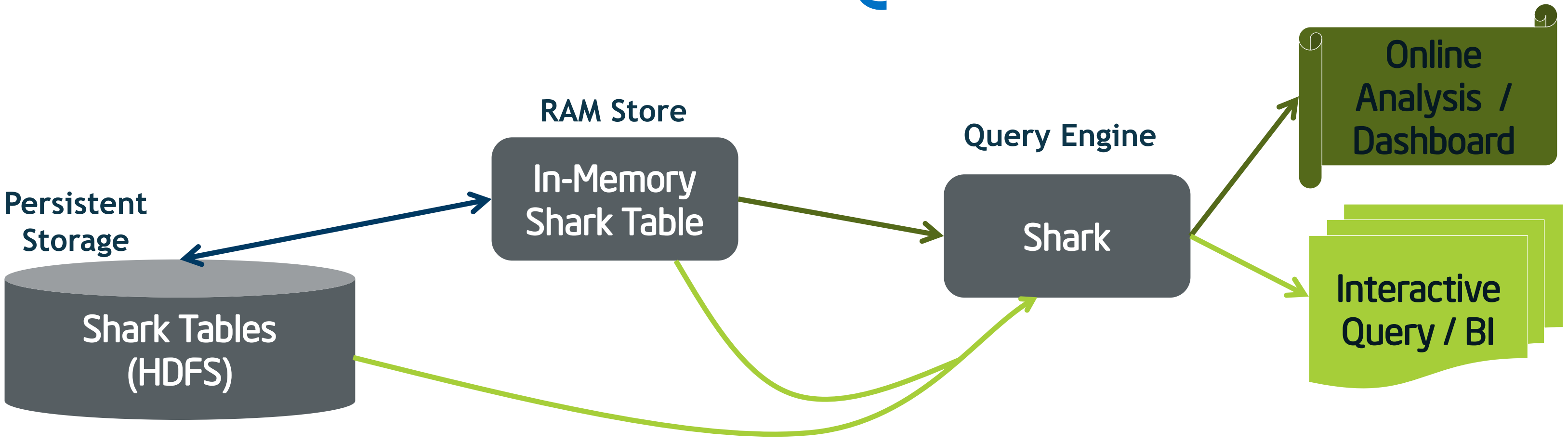
Aggregation results written to Shark table cached in memory

- Currently output as cached RDD by Spark Streaming
  - Require Spark Streaming embedded in the Shark server JVM
- Plan to move to Tachyon for better sharing and fault tolerance

Both real-time aggregations and history data queried through Shark

- History data loaded into memory for iterative mining
- Working on query optimizations & standard SQl-92 support

Online and Interactive Queries



Online analysis

- A lightweight UI frontending Shark for online dashboard
- Mostly time-based lightweight queries (filtering, ordering, TopN, aggregations, etc.) with sub-second latency

Interactive query / BI

- Ad-hoc, (more) complex SQL queries (with <5second latency)
- Heavily denormalized to eliminate join as much as possible

Experience and Current Work

Spark/Shark rocks!

- Lightweight, low latency & RAM-oriented
- Working on reliability & performance improvements
  - HA, isolation, fault tolerance, metadata handling, shuffle, etc.

Current improvements

- Shuffle performance improvements
  - Added a new netty-based shuffle module for Spark
- Shark query optimizations
  - Generated bytecode for expression evaluations
- Better operability
  - Added a new FairScheduler for Spark tasks
  - Added job history logs for Spark jobs

Future Work

- Integrate with Tachyon
- More shuffle improvements
  - Reduce the number of many (small) shuffle files
  - Actively remove shuffle files
  - Pipeline data shuffles and “map” tasks
- Further Shark query optimizations
  - Bytecode generation coverage (esp. aggregation)
  - Co-partitioned table for join
  - TopN pushdown

Future Work

- More Shark features
  - SQL-92 support (integrating “Project Panthera” with Shark)
  - Streaming support in Shark
- Even better operability
  - High availability

intel®