Analytics on Spark & Shark @Yahoo

PRESENTED BY

Tim Tully

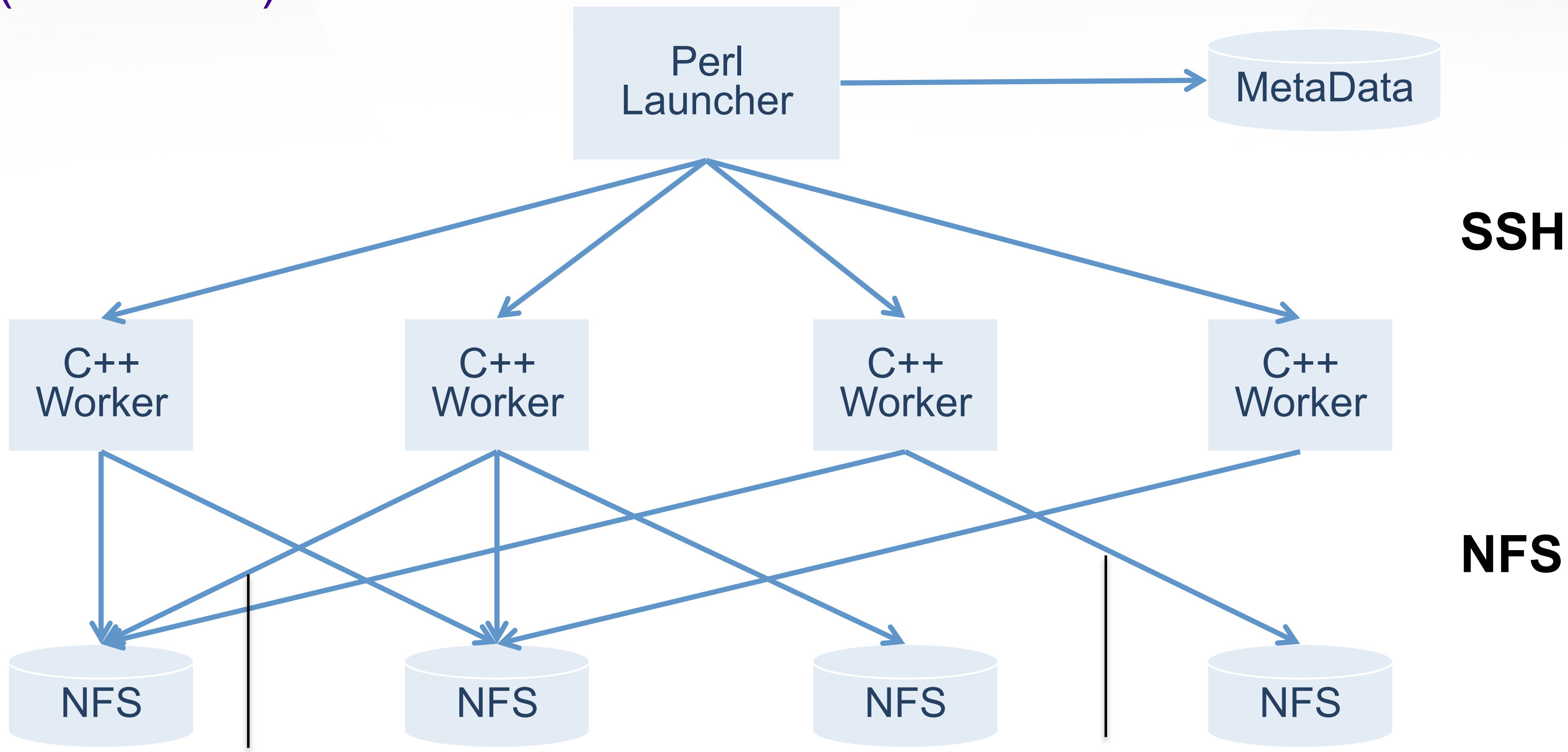
December 3, 2013

Overview

- Legacy / Current Hadoop Architecture
- Reflection / Pain Points
- Why the movement towards Spark / Shark
- New Hybrid Environment
- Future Spark/Shark/Hadoop Stack
- Conclusion

Some Fun: Old-School Data Processing

(1999-2007)



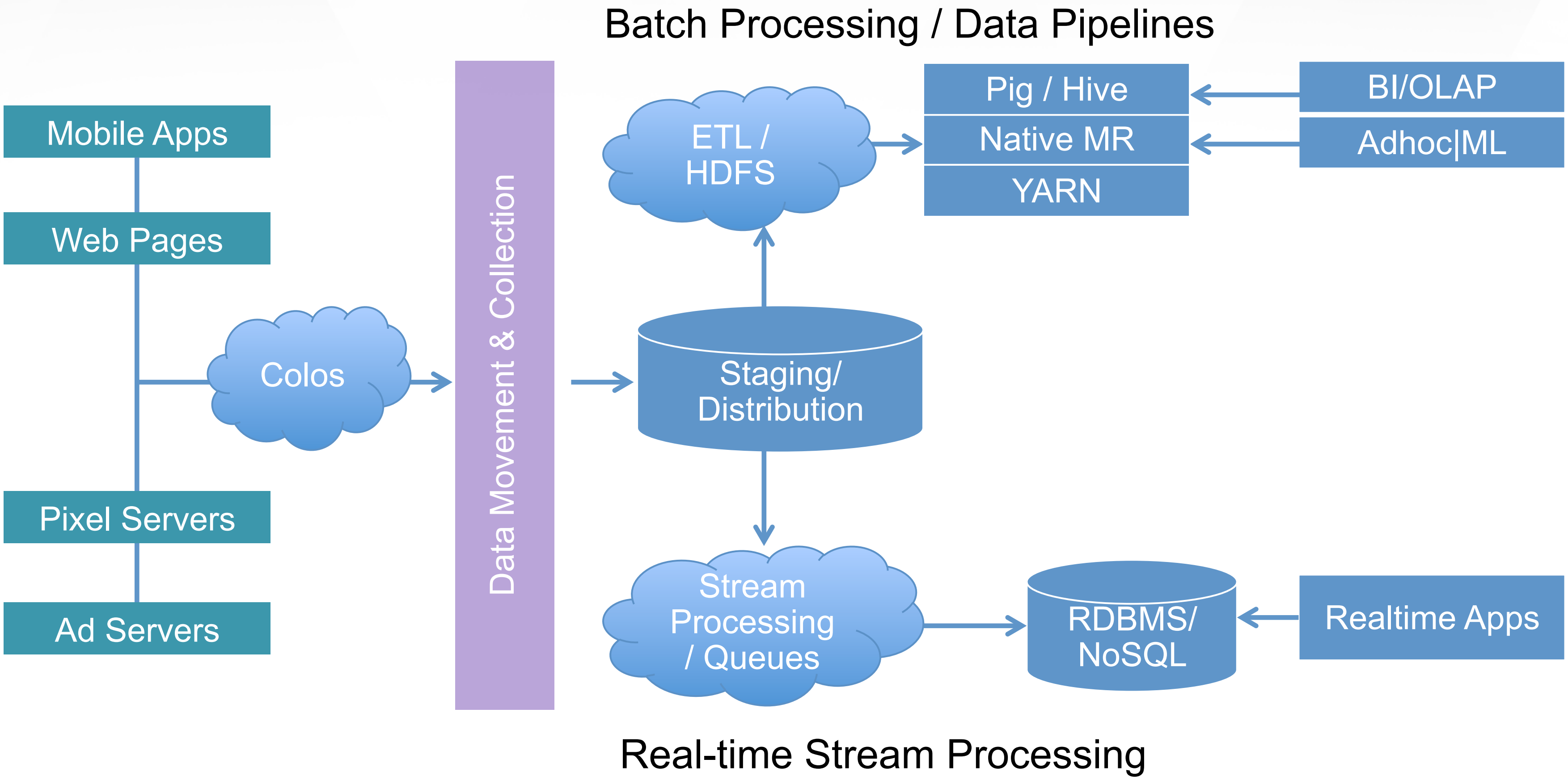
/d4/data/20030901/partition/b16.gz

/d1/data/20030901/partition/b1.gz

Current Analytics Architecture

- Custom log collection infrastructure depositing onto NFS-based storage
- Logs moved onto Hadoop HDFS
  - Multiple Hadoop instances
- Pig/MR ETL processing, massive joins, load into warehouse
- Aggregations / Report Generation in Pig, MapReduce, Hive
- Reports loaded into RDBMS
- UI / web services on top
- Realtime Stream Processing:
  - Storm on YARN
- Persistence:
  - Hbase, HDFS/Hcat, RDBMS's

Current High-Level Analytics Dataflow



Legacy Architecture Pain Points

- Massive data volumes per day (many, many TB)
- Pure Hadoop stack throughout – “Data Wrangling”
- Report arrival latency quite high
  - Hours to perform joins, aggregate data
- Culprit - Raw data processing through MapReduce just too slow
- Many stages in pipeline chained together
- Massive joins throughout ETL layer
- Lack of interactive SQL
- Expressibility of business logic in Hadoop MR is challenging
- New reports and dimensions requires engineering throughout stack

Aggregate Pre-computation Problems

- Problem: Pre-computation of reports
  - “How is timespent per user distributed across desktop and mobile for Y! Mail?”
  - Extremely high cardinality dimensions, ie, search query term
  - Count distincts
- Problem: Sheer number of reports along various dimensions
  - Report changes required in aggregate, persistence and UI layer
  - Potentially takes weeks to months
  - Business cannot wait

Problem Summary

- Overwhelming need to make data more interactive
- Shorten time to data access and report publication
- Ad-hoc queries need to be much faster than Hive or pure Hadoop MR.
  - ➢ Concept of “Data Workbench”: business specific views into data
- Expressibility of complicated business logic in Hadoop becoming a problem
  - ➢ Various “verticals” within Yahoo want to interpret metrics differently
- Need interactive SQL querying
- No way to perform data discovery (adhoc analysis/exploration)
  - ➢ Must always tweak MR Java code or SQL query and rerun big MR job
- Cultural shift to BI tools on desktop with low latency query performance

Where do we go from here?

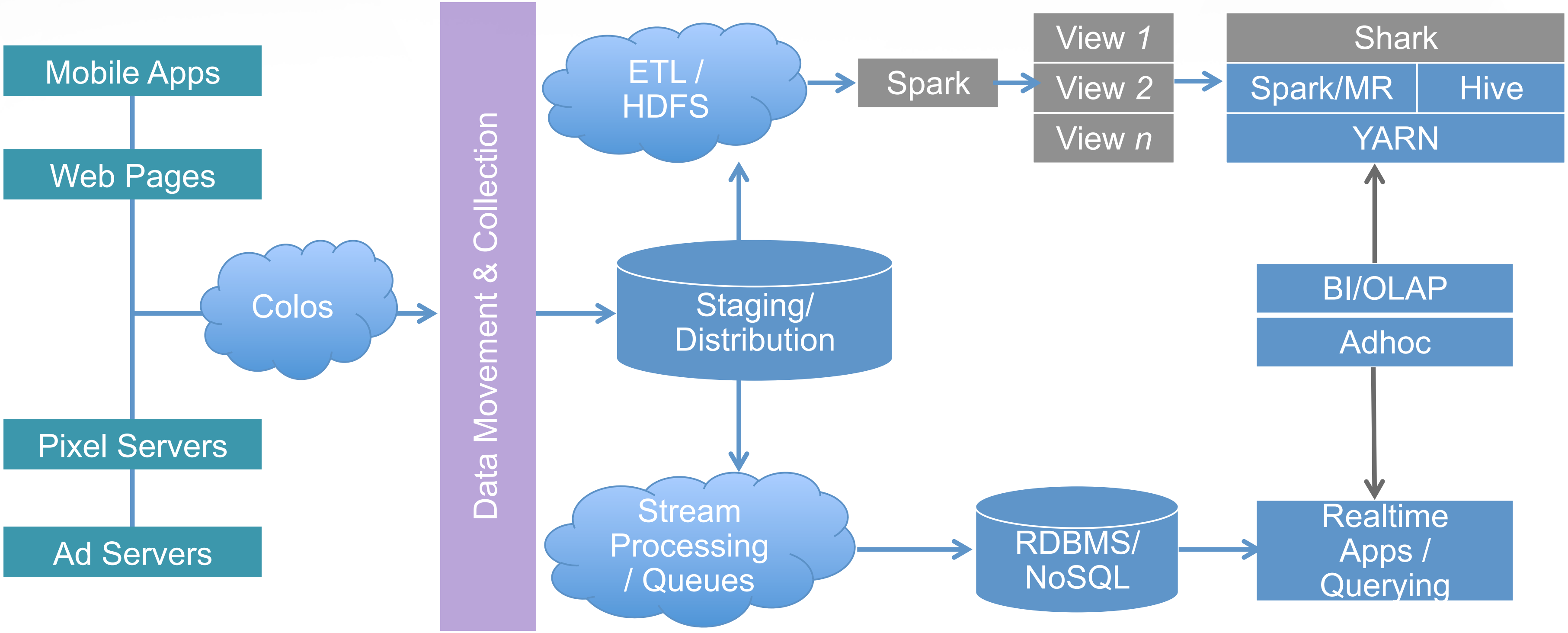
- How do we solve this problem within the Hadoop ecosystem?
    - Pig on Tez?
    - Hive on Tez?
- No clear path yet to making native MR/Pig significantly faster
- Balance pre-aggregated reporting with high demand for interactive SQL access against fact data via desktop BI tools
- How do we provide data-savvy users direct SQL-query access to fact data?

Modern Architecture: Hadoop + Spark

- **Bet on YARN**: Hadoop and Spark can coexist
- Still using Hadoop MapReduce for ETL
- Loading data onto HDFS / HCat / Hive warehouse
- Serving MR queries on large Hadoop cluster
- Spark-on-YARN side-by-side with Hadoop on same HDFS
- Optimization: copy data to remote Shark/Spark clusters for predictable SLAs
  - While waiting for Shark on Spark on YARN (Hopefully early 2014)

Analytics Stack of the Future

Batch Processing / Data Pipelines



Real-time Stream Processing

Why Spark?

- Cultural shift towards data savvy developers in Yahoo
  - Recently, the barrier to entry for big data has been lowered
- Solves the need for interactive data processing at REPL and SQL levels
- In-memory data persistence obvious next step due to continual decreasing cost of RAM and SSD's
- Collections API with high familiarity for Scala devs
- Developers not restricted by rigid Hadoop MapReduce paradigm
- Community support accelerating, reaching steady state
  - More than 90 developers, 25 companies
- Awesome storage solution in HDFS yet processing layer / data manipulation still sub-optimal
  - Hadoop not really built for joins
  - Many problems not Pig / Hive Expressible
  - Slow
- Seemless integration into existing Hadoop architecture

Why Spark? (Continued)

- Up to 100x faster than Hadoop MapReduce
- Typically less code (2-5x)
- Seemless Hadoop/HDFS integration
- RDDs, Iterative processing, REPL, Data Lineage
- Accessible Source in terms of LOC and modularity
- BDAS ecosystem:
  - Spark, Spark Streaming, Shark, BlinkDB, MLlib
- Deep integration into Hadoop ecosystem
  - Read/write Hadoop formats
  - Interop with other ecosystem components
  - Runs on Mesos & YARN
  - EC2, EMR
  - HDFS, S3

Spark BI/Analytics Use Cases

- Obvious and logical next-generation ETL platform
  - Unwind “chained MapReduce” job architecture
    - ETL typically a series of MapReduce jobs with HDFS output between stages
    - Move to more fluid data pipeline
  - Java ecosystem means common ETL libraries between realtime and batch ETL
  - Faster execution
    - Lower data publication latency
    - Faster reprocessing times when anomalies discovered
  - Spark Streaming may be next generation realtime ETL
- Data Discovery / Interactive Analysis

Spark Hardware

- 9.2TB addressable cluster
- 96GB and 192GB RAM machines
- 112 Machines
  - SATA 1x500GB 7.2k
  - Dual hexa core Sandy Bridge
- Looking at SSD exclusive clusters
  - 400GB SSD – 1x400GB SATA 300MB/s

Why Shark?

- First identified Shark at Hadoop Summit 2012
  - After seeing Spark at Hadoop Summit 2011
- Common HiveQL provides seemless federation between Hive and Shark
- Sits on top of existing Hive warehouse data
  - Multiple access vectors pointing at single warehouse
- Direct query access against fact data from UI
- Direct (O/J)DBC from desktop BI tools
- Built on shared common processing platform

Yahoo! Shark Deployments / Use Cases

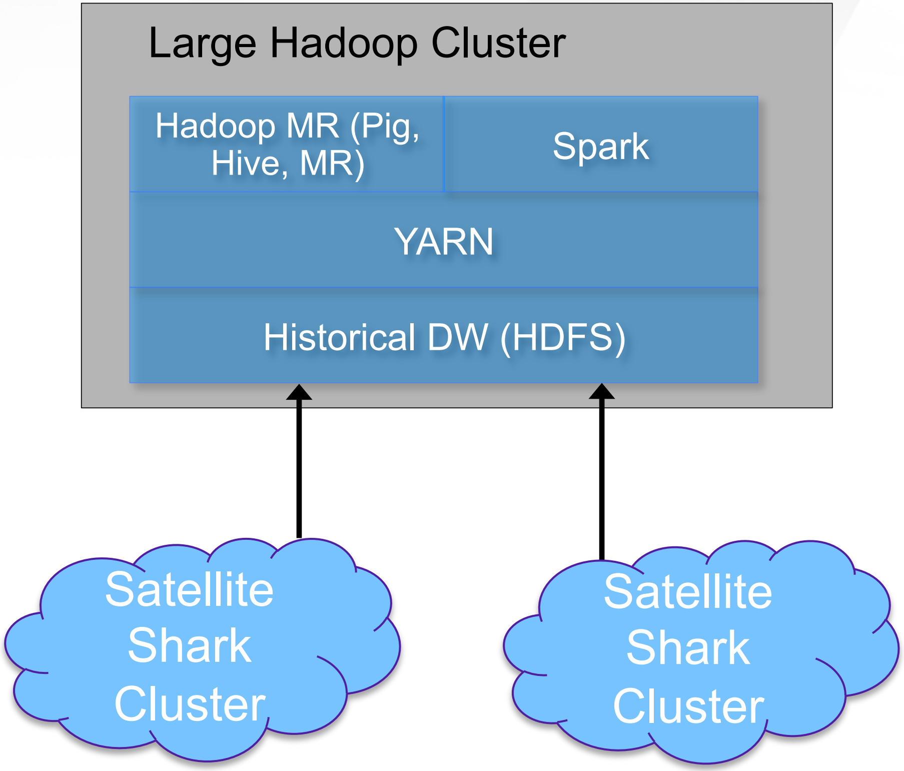
- Advertising / Analytics Data Warehouse
  - Campaign Reporting
    - Pivots, time series, multi-timezone reporting
  - Segment Reporting
    - Unique users across targeted segments
    - Ad impression availability for given segment
  - Overlap analysis – fact to fact overlap
  - Other Time Series Analysis
- OLAP
  - Tableau on top of Shark
  - Custom in-house cubing and reporting systems
- Dashboards
- Adhoc analysis and data discovery

Yahoo! Contributions

- Began work in 2012 on making Shark more usable for interactive analytics/ warehouse scenarios
  - Shark Server for JDBC/ODBC access against Tableau
    - Multi-tenant connectivity
    - Threadsafe access
  - Map Split Pruning
    - Use statistics to prune partitions so jobs don't launch for splits w/o data
    - Bloom filter-based pruner for high cardinality columns
  - Column pruning – faster OLAP query performance
  - Map-side joins
  - Cached-table Columnar Compression (3-20x)
  - Query cancellation

Physical Architecture

- Spark / Hadoop MR side-by-side on YARN
- Satellite Clusters running Shark
  - Predictable SLAs
  - Greedy pinning of RDDs to RAM
  - Addresses scheduling challenges
- Long-term
  - Shark on Spark-on-YARN
  - Goal: early 2014



Future Architecture

- Prototype migration of ETL infrastructure to pure Spark jobs
  - Breakup chained MapReduce pattern into single discrete Spark job
  - Port legacy Pig/MR ETL jobs to Spark (TB's / day)
  - Faster processing times (goal of 10x)
  - Less code, better maintainability, all in Scala/Spark
  - Leverage RDDs for more efficient joins
- Prototype Shark on Spark on YARN on Hadoop cluster
  - Direct data access over JDBC/ODBC via desktop
  - Execute both Shark and Spark queries on YARN
- Still employ "satellite" cluster model for predictable SLAs in low-latency situations
- Use YARN as the foundation for cluster resource management

Conclusions

- Barrier to entry for big data analytics reduced, Spark at the forefront
- Yahoo! now using Spark/Shark for analytics on top of Hadoop ecosystem
- Looking to move ETL jobs to Spark
- Satellite cluster pattern quite beneficial for large datasets in RAM and predictable SLAs
- Clear and obvious speedup compared to Hadoop
- More flexible processing platform provides powerful base for analytics for the future