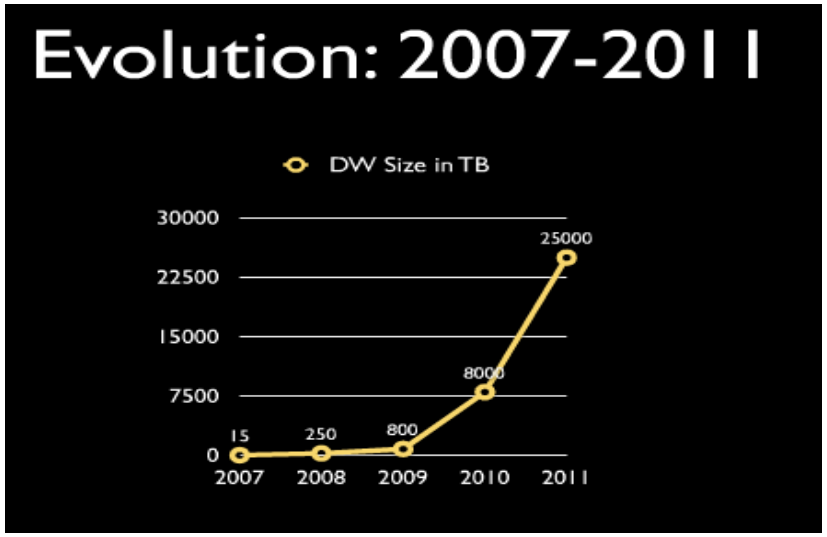
Facebook Architecture Improvements During 2007-2011

Topics:

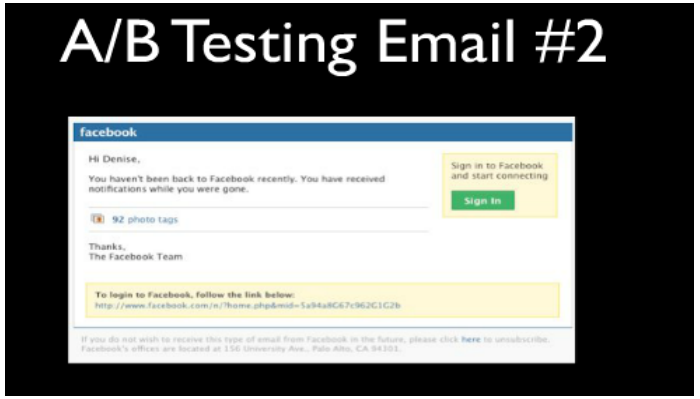
- Big Data at Facebook - Scale & Scope
- Evolution of Big Data Architectures at Facebook during 2007-2011
  - FB Traditional EDW in 2007, Limitations and Pain Points
  - FB Architecture using Hadoop in 2008 and Advantages
  - FB Architecture using Hadoop in 2009 and consequences
  - FB Architecture using Hadoop in 2010
  - FB Architecture using Hadoop in 2011

- As of 2012
  - Clusters at FB is managing about 25PetaBytes (Equal to 300 years of HD-TV program) compressed data
  - This is equal to 150PB uncompressed data
  - Every day 4TB compressed data of received



- Why to collect this and What is the use of this data
  - Various simple to complex applications uses this data to report active user counts, how many ads served etc…
  - Adhoc queries/analysis , Hypothesis testing and data sciences
  - Index generation

- **Facebook Data used by Applications**
  - **A/B Testing Email**
    - **Reaching to group of users with specific content**
    - **Email#2 is 3X times more engaging than Email#1**



An emailing campaign example(A/B Testing):

	Overall	Men	Women
Total sends	2,000	1,000	1,000
Total responses	80	35	45
Code A1	50 / 1,000 (5%)	10 / 500 (2%)	40 / 500 (8%)
Code B1	30 / 1,000 (3%)	25 / 500 (5%)	5 / 500 (1%)

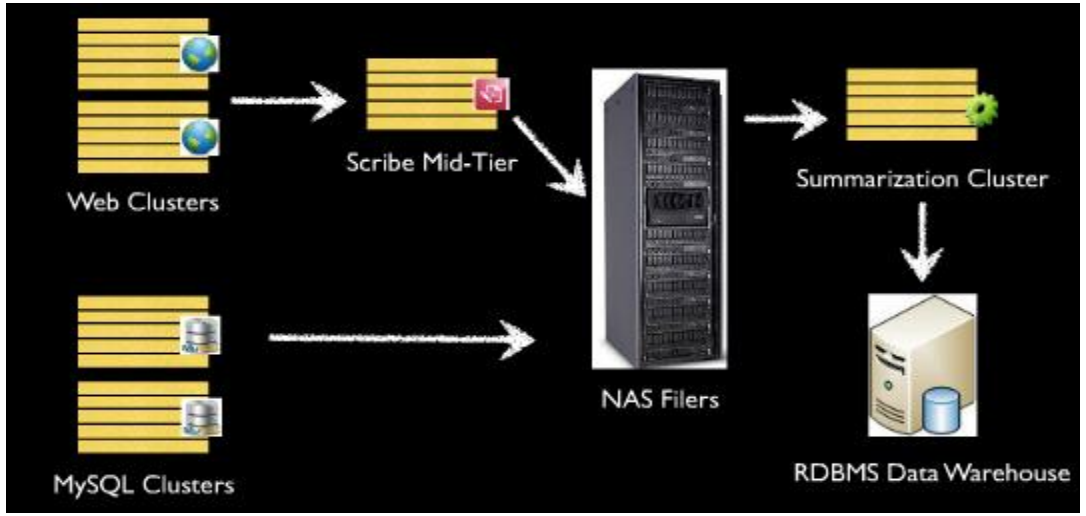
  - **Friend Map**
    - **Visualizing the friendship by plotting between cities where they live**

  - **15% of the companies access the data from FB clusters.**

FB Traditional EDW in 2007, Pain Points and Limitations

Action Data

Log Collection Framework written in FB



DATvsSANvsNAS

Home developed summarized scripts

***Scribe Mid Tier captures the data coming from 10s/1000s of Web clusters and dumps into NAS filers

Dimensional Data

EDW(RDBMS) contains summarized data

Data Dumped into NAS Filers

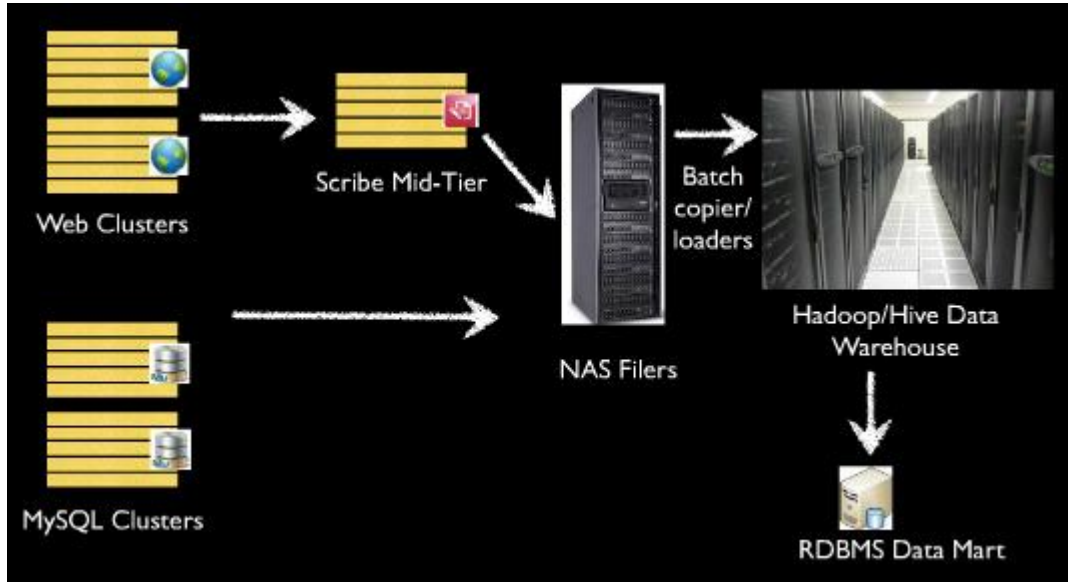
Pain Points:

- ETL Jobs which summarizes the data took >24hrs even after tuning (Even after adding CPUS/Memory/Upgrade Clusters etc..)
- Compute close to the storage using Map-Reduce programming (Beginning of Map-Reduce program)

Limitations:

- Most of the use cases were in business metrics
- Only summarized data was online and detail data was archived offline
- Data Science and Modeling is not possible

FB Architecture using Hadoop in 2008 and Advantages



- Summarized clusters are replaced with Enhanced Hadoop
- As Map-Reduce programs are written in Java, to make the data in all the clusters accessible to all the users, sql kind of interface query tool HIVE brought in

Advantages:

- Data was accessible online
- Data science is possible
- Business metrics can be regenerated with logic change and can be compared with previous metrics
- Infrastructure problem was solved

FB Architecture using Hadoop in 2009 and consequences

- Focused on building the tools in Hadoop/Hive DW to make information available to everyone rather than a select few

Netcar : Create a map and rest this will take care in handling the Map. Evaluating schema built in, New changes will be taken care automatically like column addition, decomposes data into column format by compressing, short term data in the json format with 30min latency

HiPal(GUI): At Facebook querying and analysis of data is done predominantly through Hive. The data sets are published in Hive as tables with daily or hourly partitions. The primary interfaces to interact with Hive for ad hoc query purposes are a web based GUI - HiPal - and the Hive command line interface – Hive CLI. User queries can be tracked and guidance can be made to use right tables with tuning

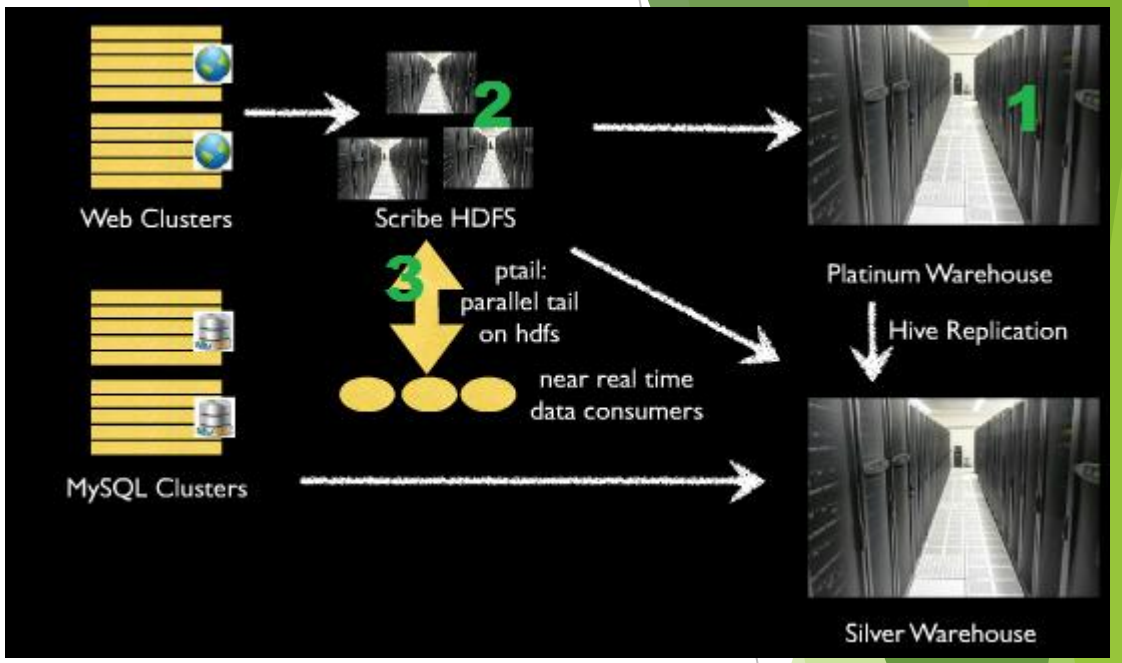
Databee: Is a python framework for specifying the monitoring and alerting in case of failures and good dashboards to check their current status, past runs or time of completion of various periodic batch jobs. It also collects stats on the amount of data handled by different stages in the data pipeline. (ETL/ELT transformations)

Scrapes: The scrape processes dump the desired dimensional data sets from mysql databases, compressing them on the source systems and finally moving them into the Hive-Hadoop cluster.These processes tries to maintain consistency and restores the previous snapshot in case of failures.

Consequences: As more tools available for accessing the data, more and more Map jobs running by various users which Might run for longer time(bad jobs) or blocking the Data in clusters from access, max map slots occupied

FB Architecture using Hadoop in 2010

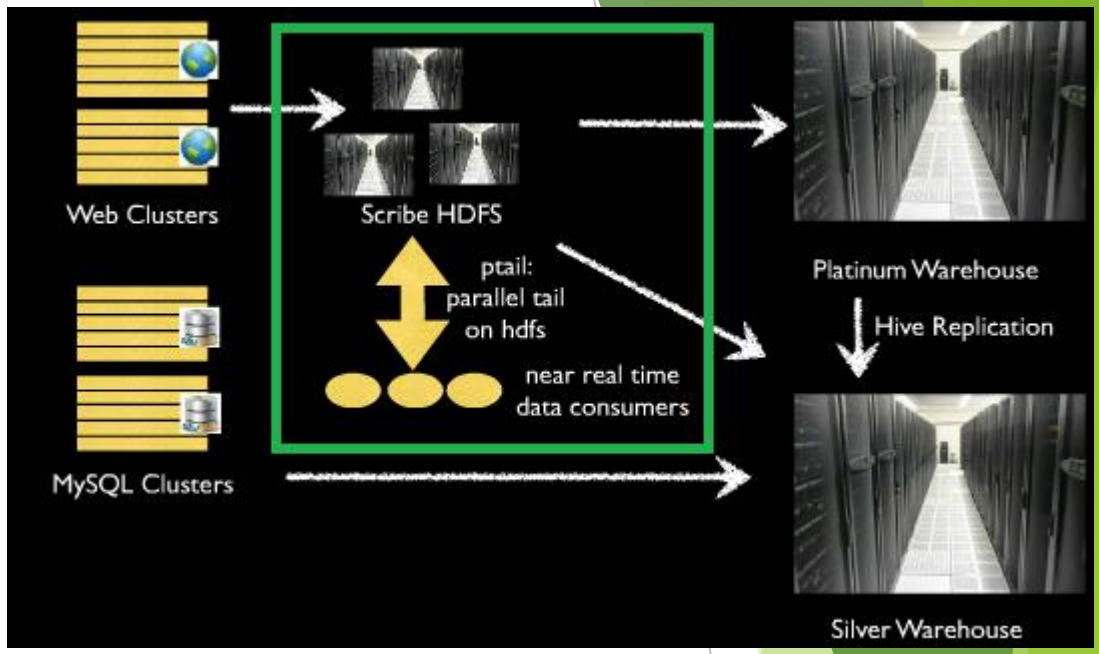
- **Below enhancements are addressed in 2010:**
  - Hadoop Hive DW **Split** into **Platinum DW** (Production data which is most recent and will be used by jobs with SLA) and **Silver DW**(Historical Data)
  - Data replication using Hive from Platinum to Silver and loading non-prod data into Silver DW directly
  - Filers and Scribe-Mid Tire replaced with Hadoop scribe demons running
  - **Ptail** software built to access near real time data by the Infrastructure and security team(Check points maintained for continuous streaming).
  - As Capacity and CPU are still major challenges, **replicas** brought down to 2.2 from 3 which helped in saved in storage
  - Implemented in **RCFile**(Row Columnar at block level
  - Implemented continuous copiers/loaders using the batch jobs by running more **frequently**
  - Added more Hive **optimizations** to save CPU
  - Added more **monitoring** mechanism
    - Job statistics at various levels
    - Expected and actual completion timings
    - Data quality analytics on Job statistics



- **Advantages:**
  - Isolation between the jobs
  - Reduced the operational overhead
  - Better resource utilization
  - Measurement, Ownership and Accountability

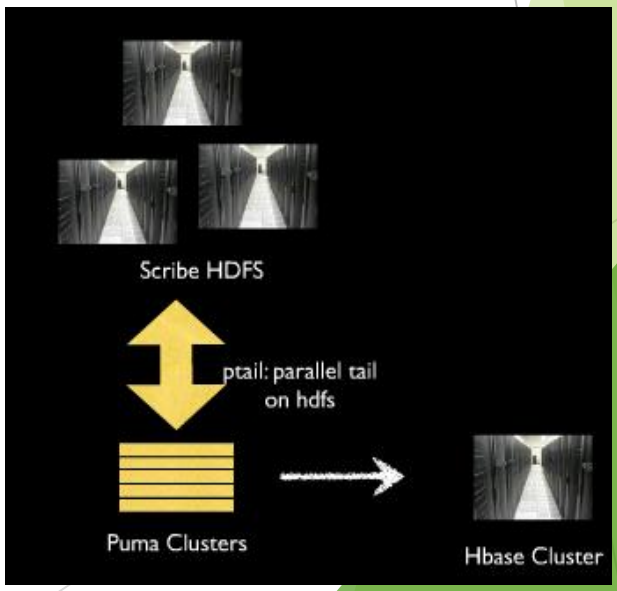
FB Architecture using Hadoop in 2011

- As more and more near time aggregations details demanded by the users to make analysis much faster than Hadoop batch processing
- **Puma brought in to address this**
  - PTail provides parallel data Streams
  - Sharded by Aggregation Key(Shard is hash map in memory)
  - Hash Map is pair of Aggregation Key and User defined aggregation
  - Checkpoint workflow(On startup, brings from Hbase)
  - FB Keeps 80% data in memory
- **Peregrine brought in to address simple and fast queries for data exploration**
  - Peregrine is a map reduce framework designed for running iterative jobs across partitions of data. Peregrine is designed to be FAST for executing map reduce jobs by supporting a number of optimizations and features not present in other map reduce frameworks
  - Peregrine supports a number of optimizations and features not present in other map reduce frameworks including



Challenges due to Hyper Growth in data

- **Data Center movement**
  - Moved 20PB of data(30K disks)
  - Leverage replication with fast switch by enhancing hive replications
- **Moving sustainably fast**



Thank You!!!