Inspire…Educate…Transform.

Big Data

Session 5: Data Ingestion (Flume, Sqoop, Kafka), BigData Review

Suryaprakash Kompalli
Senior Mentor, INSOFE

Review of Last Lecture

- Class 1:
  - Different architectures
  - Transition from Databases to data warehouses and data lakes
  - Thinking of Large Jobs as Task Decompositions
  - How BigData is changing IT and business operations
- Class 2:
  - Hadoop: Storage

Review of Last Lecture

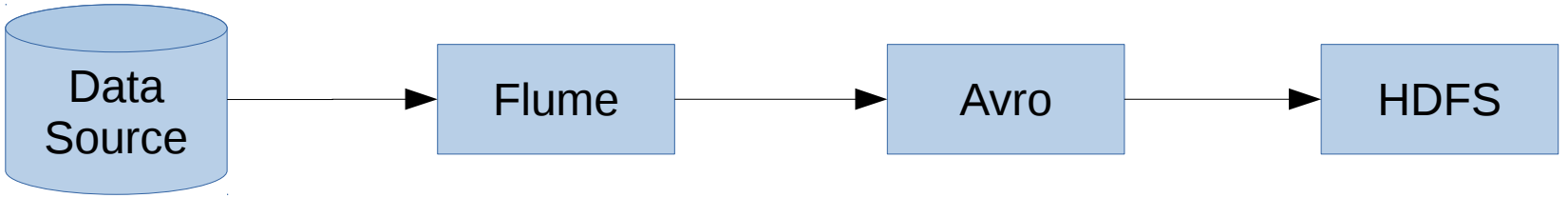
- Class 3:
  - Map-Reduce, YARN
  - Spark
- Class 4:
  - NoSQL:
    - Hbase
  - Scripting in Big Data:
    - Hive / Pig
    - Other tools
      - Impala
      - Spark SQL
      - Apache Drill

Agenda

- Data Ingestion
  - Flume
  - Sqoop
  - Kafka
- Stream Processing
  - Storm
  - Spark Stream
- Big Data Review
- Lambda Architecture

Data Ingestion Systems: Why use them?

- As structured or unstructured data comes in, it is a good idea to process it. E.x.:
  - Route incoming data into different HDFS stores based on “types” of processing to be performed on them: store data of a geography in a specific cluster, use HA cluster for some data and regular cluster for other
  - Split the input data into parts and add meta data: time stamp, user id, source machine etc. Queries can be run on this meta data. [Avro does some of this]
  - A possible flow could be like this:



Data Ingestion Systems

	Structured	Unstructured
Data@Rest (Documents, static files, directories etc)	SQOOP (We will see this with SQL)	scp (File move into hadoop), Flume, Chukwa
Data in motion (Transactional data, streaming data like twitter, video)	Kafka, Storm, Spark Streaming [Note: Most of these also process data as it is being injested]	Flume, Chukwa (Designed mostly for logs, but will work for others), Avro

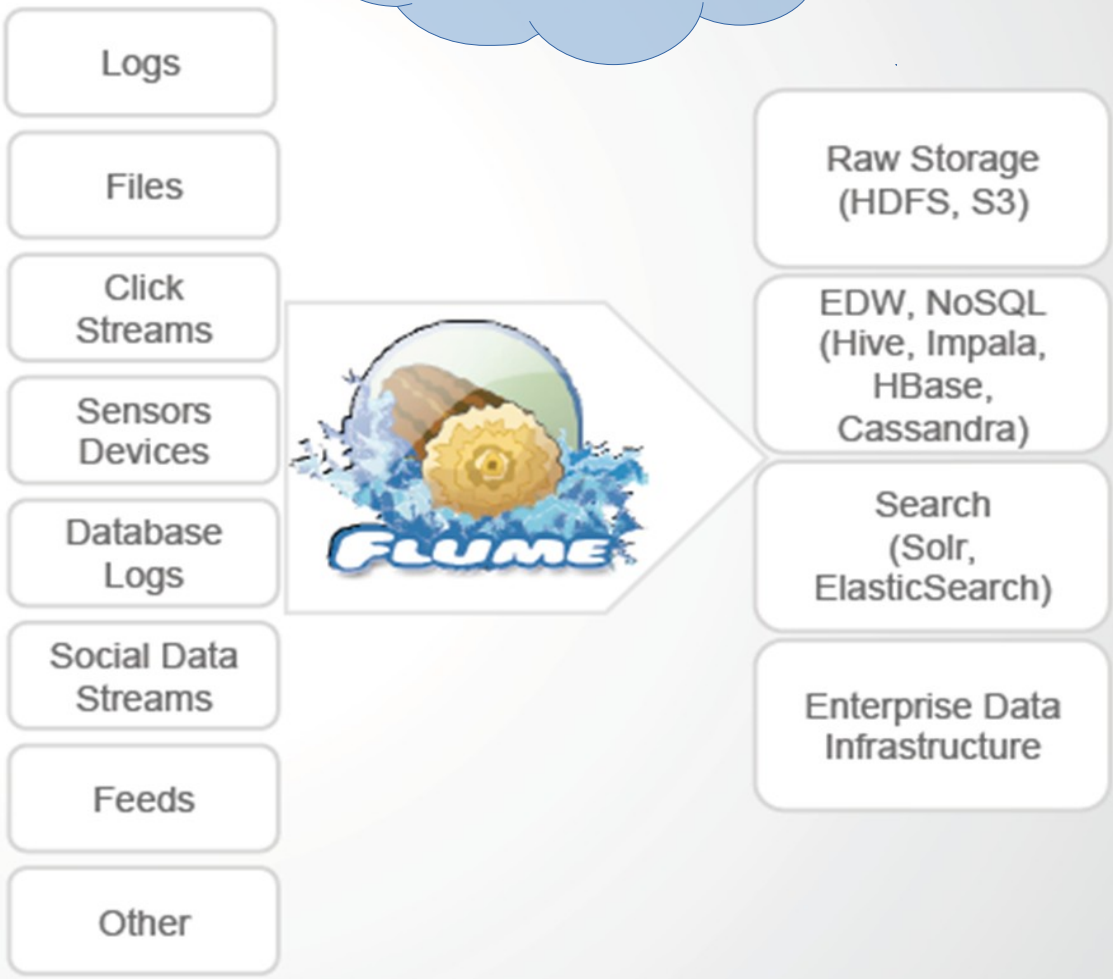
Flume

Even audio and TV channes can be an input into flume

Apache Flume is a **continuous data ingestion system** that is...

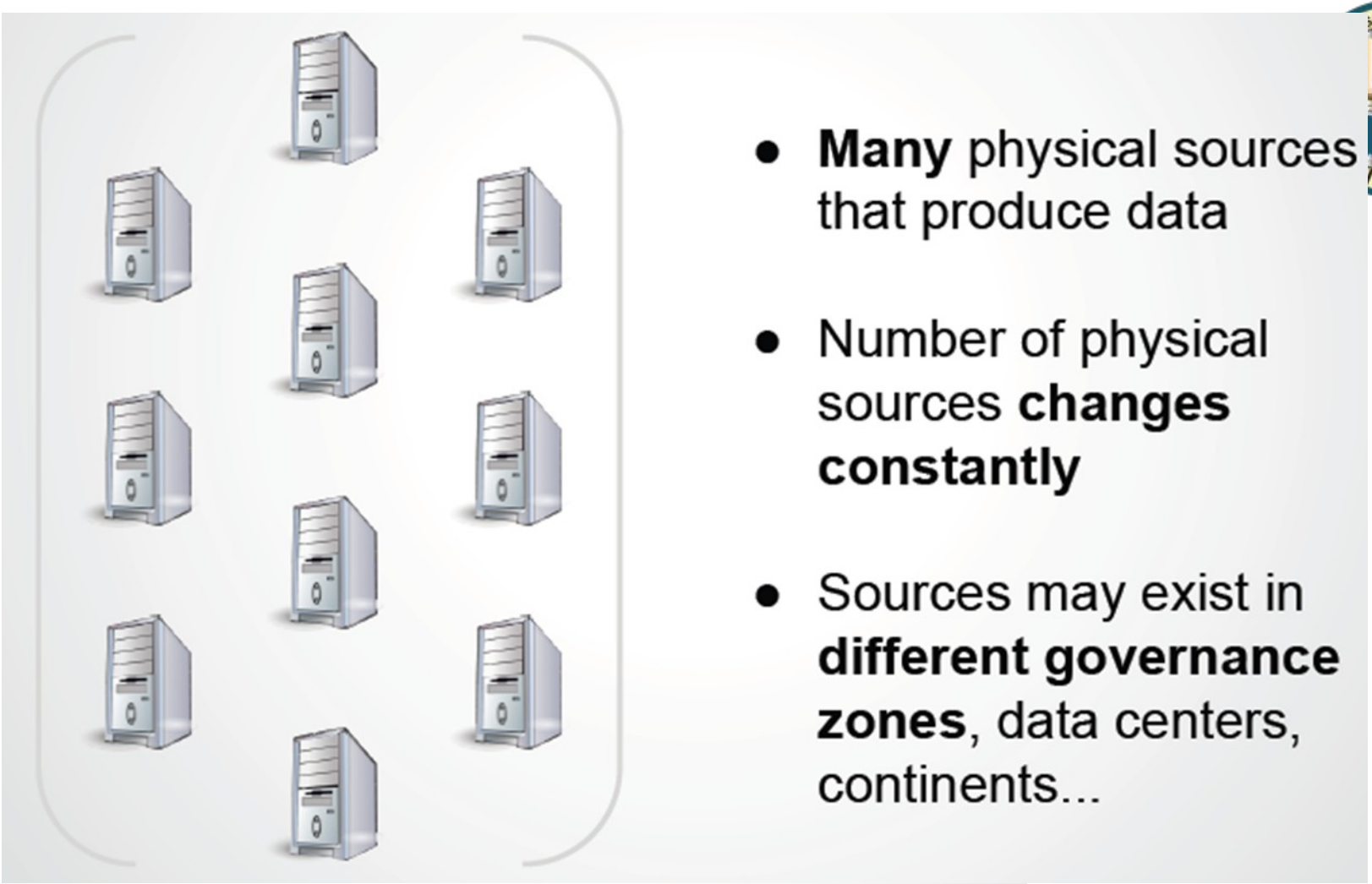
- *open-source,*
- *reliable,*
- *scalable,*
- *manageable,*
- *customizable,*

...and designed for ***Big Data*** **ecosystem**.



Some slides from Apache Flume – Data aggregation at scale. Arvind Prabhakar, Stream Set's 2015 talk. http://events.linuxfoundation.org/sites/events/files/slides/RealTimeDataIngestUsingFlume.pdf

Flume



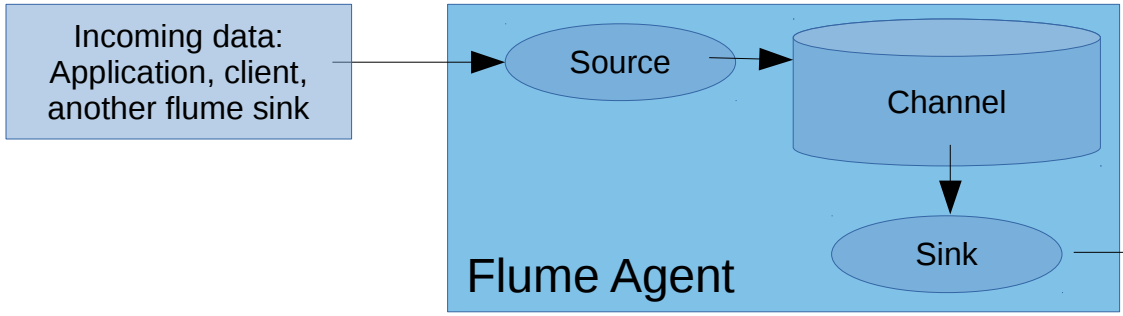
- **Many** physical sources that produce data
- Number of physical sources **changes constantly**
- Sources may exist in **different governance zones**, data centers, continents...

- Weather
- Traffic
- Automobiles
- Trains
- Airplanes
- Geological/Seismic
- Oceanographic

- Smart Phones
- Health Accessories
- Medical Devices
- Home Automation
- Digital Cameras
- Social Media
- Geolocation

- Shop Floor Sensors
- Network Activity
- Industry Appliances
- Security/Surveillance
- Server Workloads
- Digital Telephony
- Bio-simulations...

Flume



Timestamp difference between sink and source transaction helps you measure the throughput of a flume node

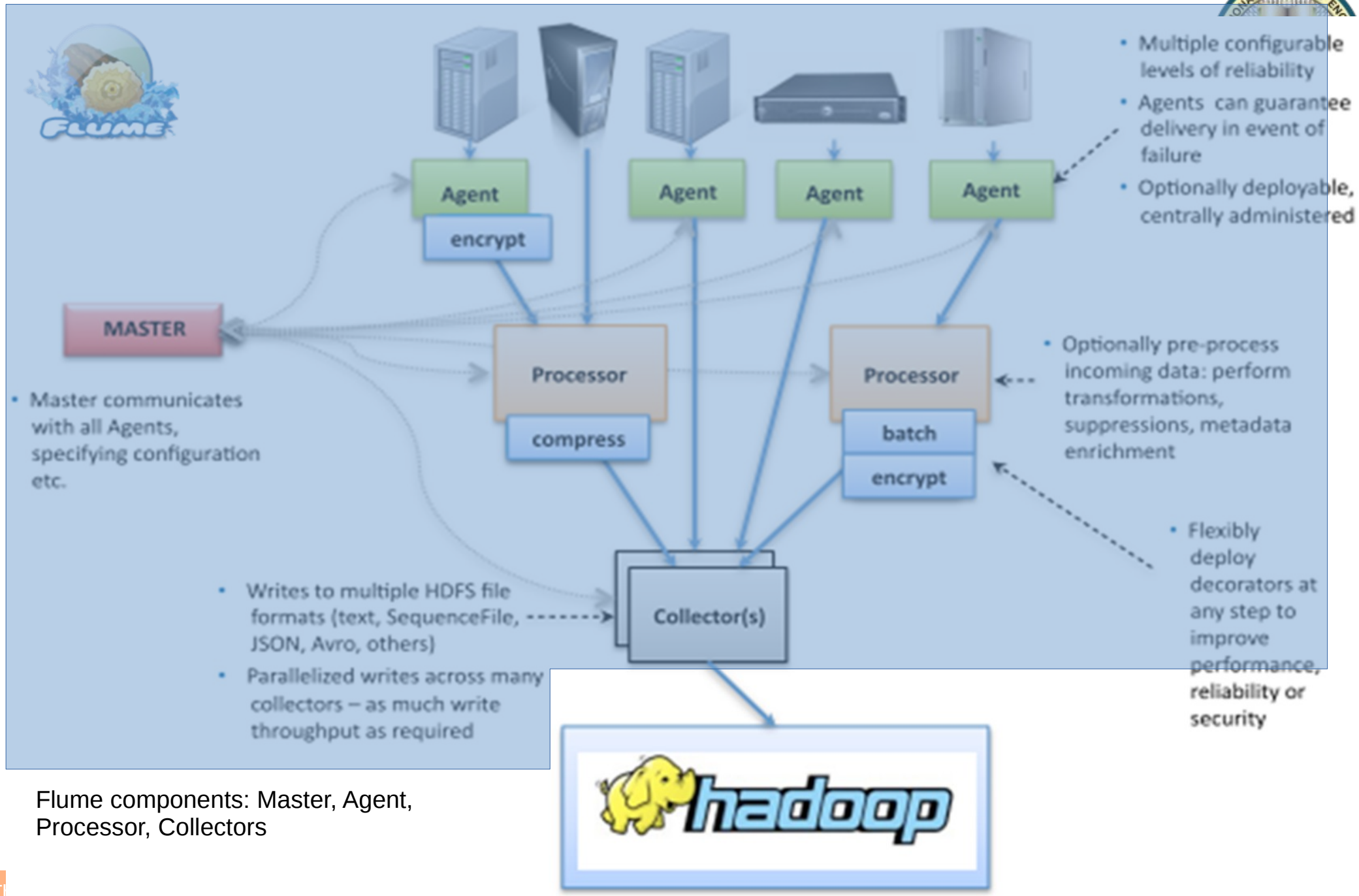
- Flume agent/node: A container that can host flume components (Master, Agent, Processor, Controller) to enable high speed data & event movement in Hadoop systems. Each flume agent has following parts:
- Source:
  - Connected to the client/another sink from where data needs to obtained
  - Require at least one channel to function; Transactions are used to write to channel
- Channel:
  - Intermediate store of data. Can be in memory (Can be lost if JVM or system reboots) or HDD (Made persistent using Write Ahead Logs/WALs)
  - Designed to work with any number of sinks and sources
- Sink:
  - Tells Flume agent/node where to send data
  - Uses transactions to remove data from channel
  - Sink transaction commits only after receiving data ack from downstream process. This ensures data is not lost in Flume
  - Sink can have one or more decorators to perform simple processing on data as it passes through:
    - Compression
    - Encryption
    - awk, grep-like functionality
    - Serialization

Flume Components

- Agent:
  - May physically be close to client end; Spec may be controlled by client
  - The source may be specifically designed for a client
  - Throughput form agent may be monitored by client or by the system operator in collaboration with client
  - Note: every system in flume is technically an agent (even master). But here, we refer to a Source-Channel-Sink combination
- Processor:
  - Source and Channel in a processor may be un-important; the sinks need perform additional processing (re-routing specific data, adding meta-data tags)
  - Physical machine and spec controlled by system operator
  - Initial spec is an educated guess
  - After some data is collected from agent, spec may be modified
- Collectors:
  - Consolidate the data and write to external system
  - Channels and sinks are likely to be critical in collectors
  - Output from the sink of a collector may be to another system like HDFS, Avro, Spark, Elastic search, Solr etc
- Master:
  - Remember your friend the heartbeat?
  - Master gets heartbeat from all three types of systems
  - It is replicated and can be HA enabled

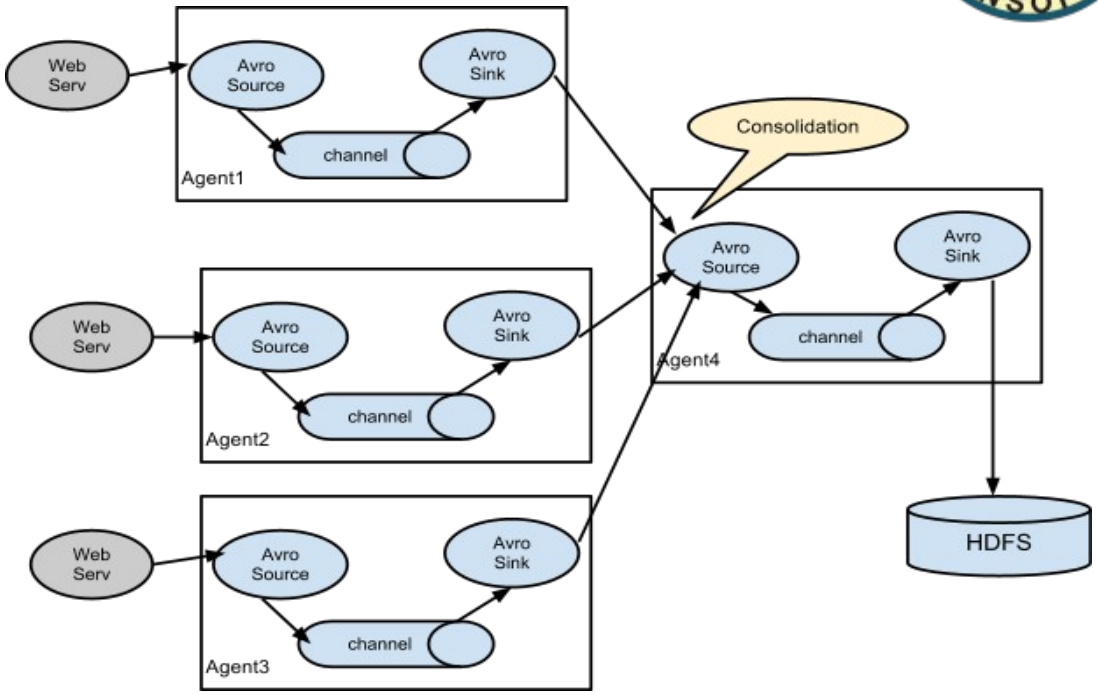
All these flume components can also be seen as agents having Source, channel and sink. They are listed differently because given the type of work an agent does, it may need different configurations. Flume lets you configure an agent; E.g.: Bigger resource for a large source (like a fast twitter/high bandwidth video feed)

Comprehensive Flume guide: http://archive.cloudera.com/cdh/3/flume/UserGuide/

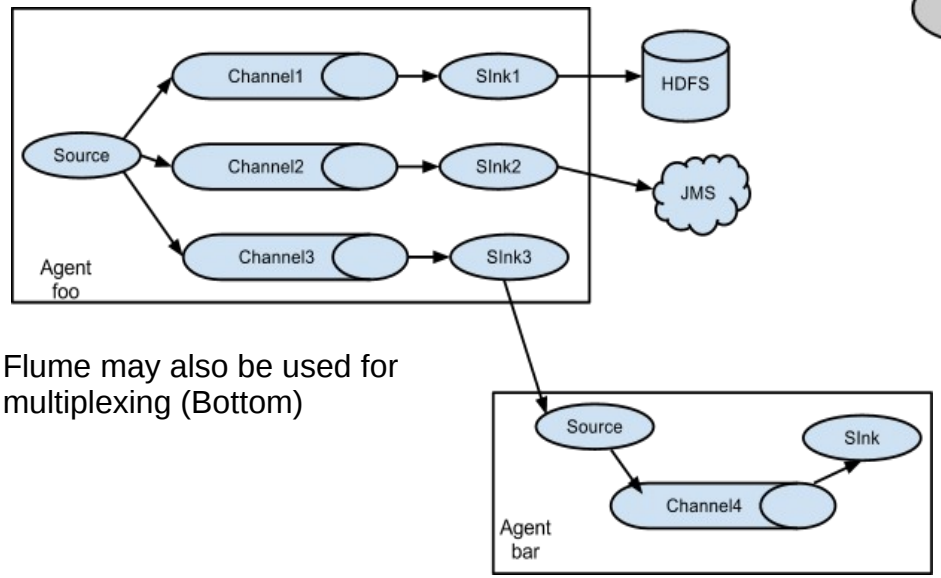


Flume components: Master, Agent, Processor, Collectors

Common Deployments of Flume



Collecting logs or other information from hundreds of application servers (Left)

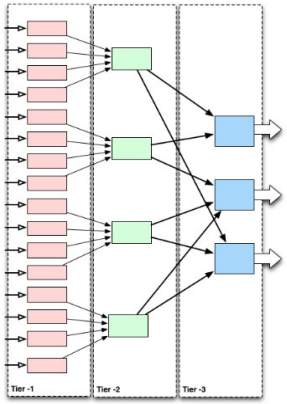


Flume may also be used for multiplexing (Bottom)

How to setup and deploy flume: https://flume.apache.org/FlumeUserGuide.html

Planning Flume

- Channels are key to ensure no data is lost when:
  - Source sends a sudden burst of data
  - Systems downstream to a sink report failures
- Correct sizing of channel is key



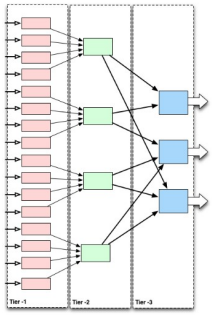
Flume employs tiers where number of agents reduce from outermost tier to inner tier

Any temporary imbalance in event volume of one tier is absorbed by channel(s)

When the traffic ebbs the Channel(s) drains out extra events to reach steady state

http://tinyurl.com/za9vphl

Planning Flume



- Number of tiers is calculated by taking upstream to downstream agent ratio:4:1, 8:1, 16:1 etc
  - 4:1 implies 4 upstream agents at client to 1 agent after the last sink
- Exit batch size: $\frac{\text{steady state data volume exiting the tier}}{\text{number of agents in tier}}$

- Channel capacity: Worst case data injest sustained from clients when faced with worst case downstream data rate after last sink. Depends on number of discs, write speed etc

For detailed planning, see slides 32 to 36 of: http://tinyurl.com/za9vphl

Flume - Summary

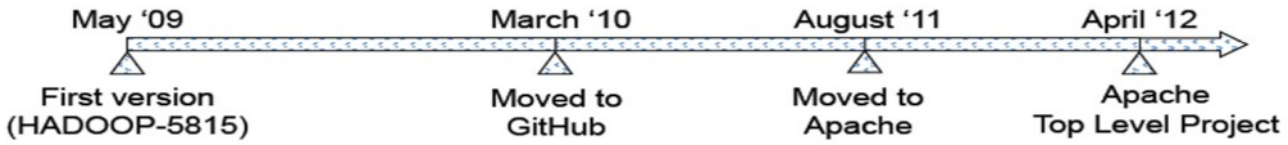
- Suitable for large volume data injestion, especially if data comes from multiple locations
- Once planned and sized, will run with least operational involvement
- Transactional and write ahead logs ensures data is rarely lost between agents
- Has rich out of box features for contextual routing, support for popular client and destination systems
- Failure of master: Flume network operations will continue, but automated resource allocation will stop; Typically, major issues do not occur before standby takes over
- Kafka is also great for data in motion, but is more complex
  - We will study this later

Agenda

- Data Ingestion
  - Flume
  - Sqoop
  - Kafka
- Stream Processing
  - Storm
  - Spark Stream
- Big Data Review
- Lambda Architecture

SQOOP (Born 2009, Apache-ed in 2012)

- RDBMS-Hadoop interoperability is key to Enterprise Hadoop adoption
- SQOOP provides a good general purpose tool for transferring data between any JDBC database and Hadoop
- SQOOP extensions can provide optimizations for specific targets
- Why not Flume?
  - Flume does not have a way to import the DB structure



Sqoop: Quick Overview

- Enables Bulk Transfer:
  - Import/Export from/to relational DBs, enterprise warehouses, and NoSQL
  - Populate HBase, Hive and HDFS
  - Integrate with Oozie as an action
  - Support plugins via connector-based architecture

Sqoop

Implemented in Map-Reduce

- Import & Export for ODBC, JDBC Data Sources
- CSV Files in HDFS

SQOOP

Export from HDFS to RDBMS

- Read files in HDFS directory via MapReduce
- Bulk parallel insert into database table

Import from RDBMS into HDFS

- Divide table into ranges using primary key max/min
- Create mappers for each range
- Mappers write to multiple HDFS nodes
- Creates text or sequence files
- Generates Java class for resulting HDFS file
- Generates Hive definition and auto-loads into HIVE

E.x.: You have a crawler that populated HDFS; You need to export some critical counts or list of links to RDBMS

Export from Hive/HBase to RDBMS needs to be done via HDFS. Direct transfer is not enabled

A part of SQOOP is on the RDBMS to do this

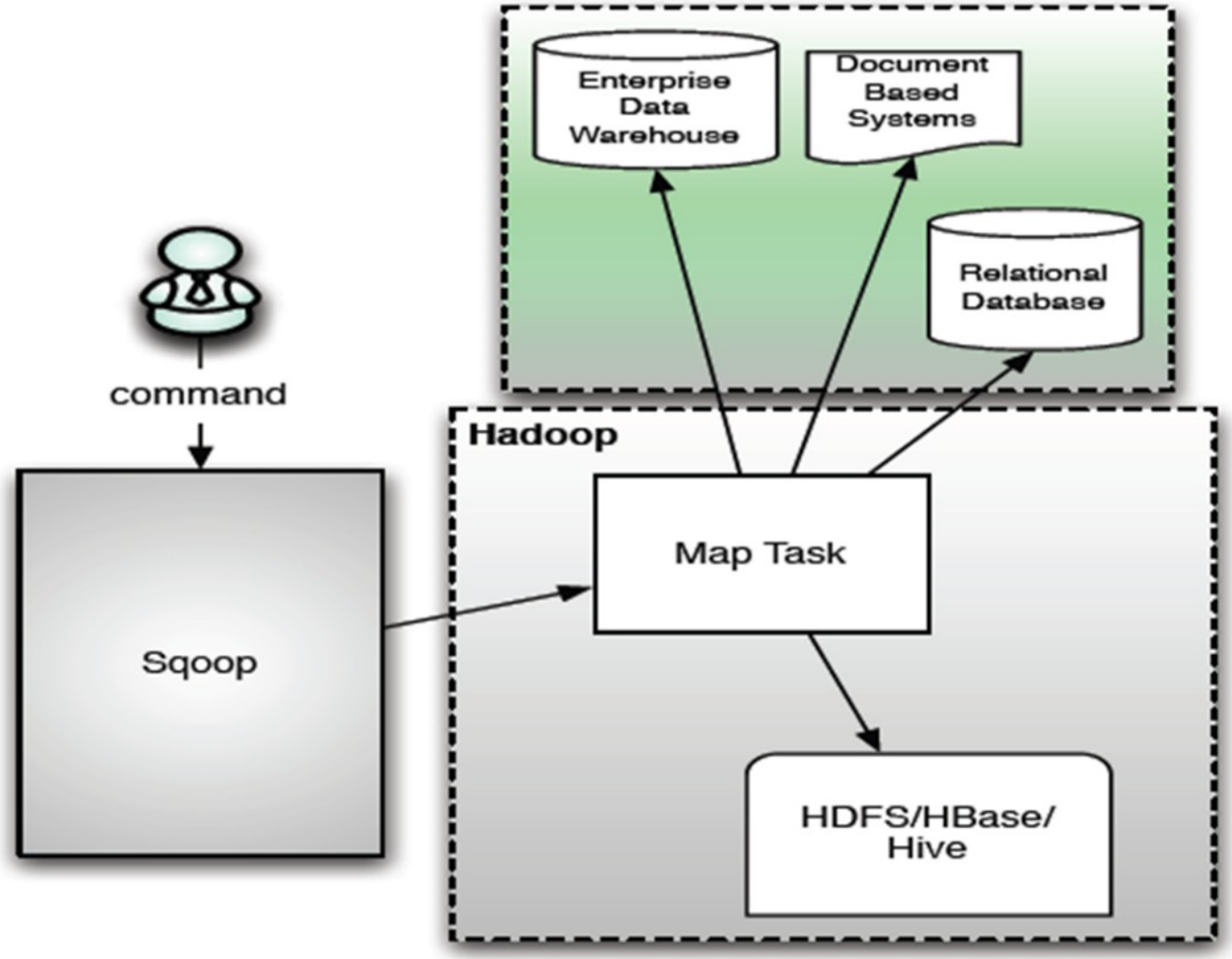
You need to store the RDBMS structure somewhere; A java class is created out of schema so that when the data needs to be used in MR later on, the schema is available

The Schema is stored into HDFS or into a Hive definition (Very similar to Data Definition Language)

SQOOP

- SQOOP features:
  - Compatible with almost any JDBC enabled database
    - Many plugins available for commercial RDBMS systems
  - Auto load into HIVE
  - HBase support
  - Job management
  - Cluster configuration (jar file distribution)
  - WHERE clause support
  - Open source, and included in most Hadoop distributions

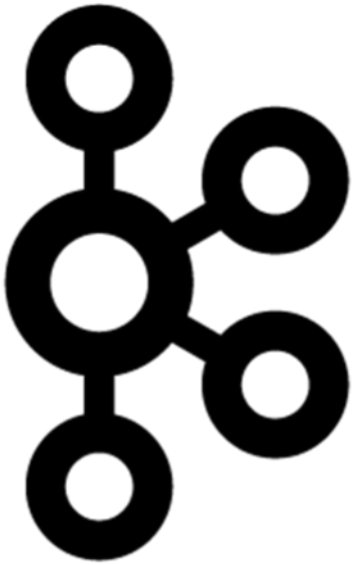
Sqoop Architecture



Automatic monitoring or scheduling is not present

Agenda

- Data Ingestion
  - Flume
  - Sqoop
  - Kafka
- Stream Processing
  - Storm
  - Spark Stream
- Big Data Review
- Lambda Architecture



kafka

Some of the slides taken from Akash Vacher's Dec 2015 talk at Kafka first Meetup, Bangalore

Kafka: Overview

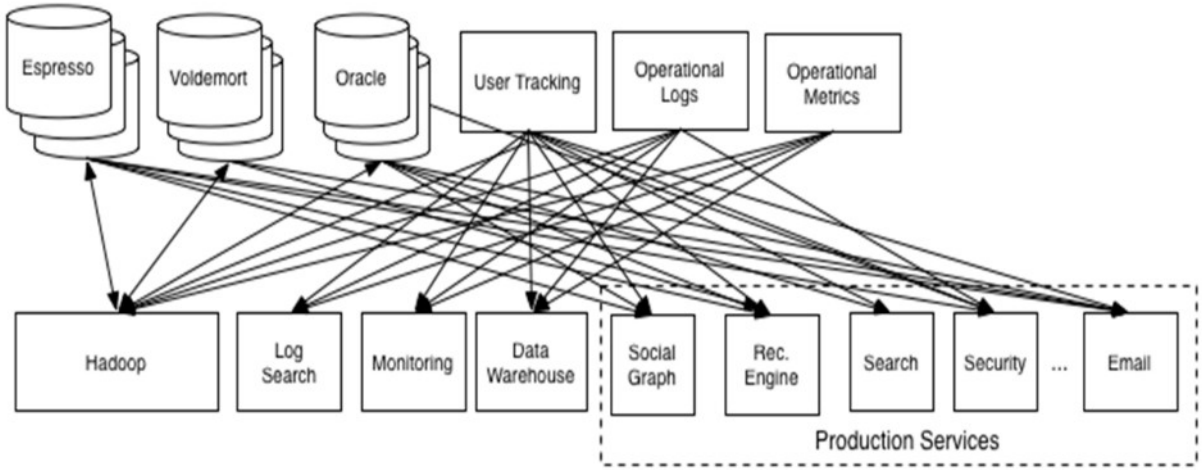
- High-throughput distributed messaging system
- Kafka guarantees:
  - At least once delivery
  - Strong ordering
- Developed at Linkedin and open sourced in early 2011
- Implemented in Scala and Java

https://gigaom.com/2014/11/06/the-team-that-created-kafka-is-leaving-linkedin-to-build-a-company-around-it/

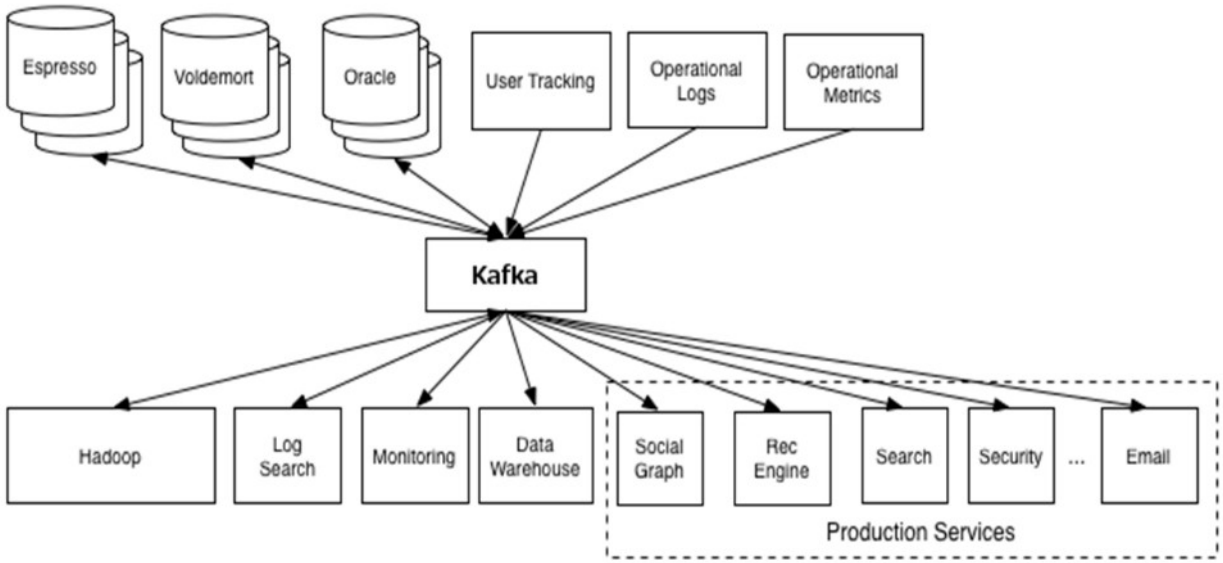
Attributes of a Kafka Cluster

- Disk Based
- Durable
- Scalable
- Low Latency
- Finite Retention
- Motivation:
  - Unified platform to handle all real time data feeds
  - High throughput
  - Stream Processing
  - Horizontally scalable

Kafka Architecture

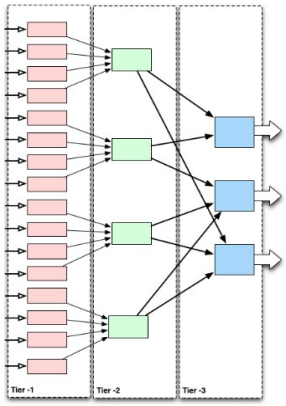


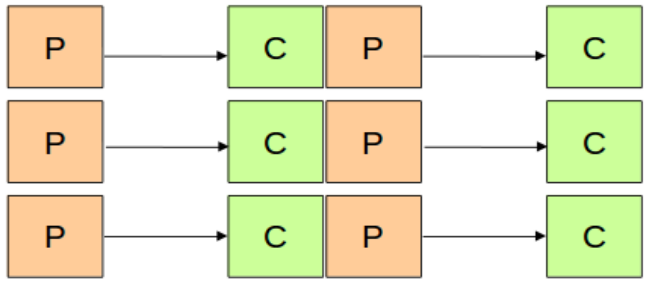
Without and With Kafka



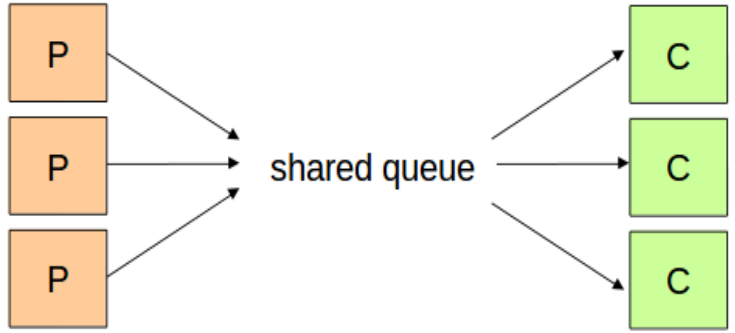
Kafka vs Flume: Notice the core concepts

Flume

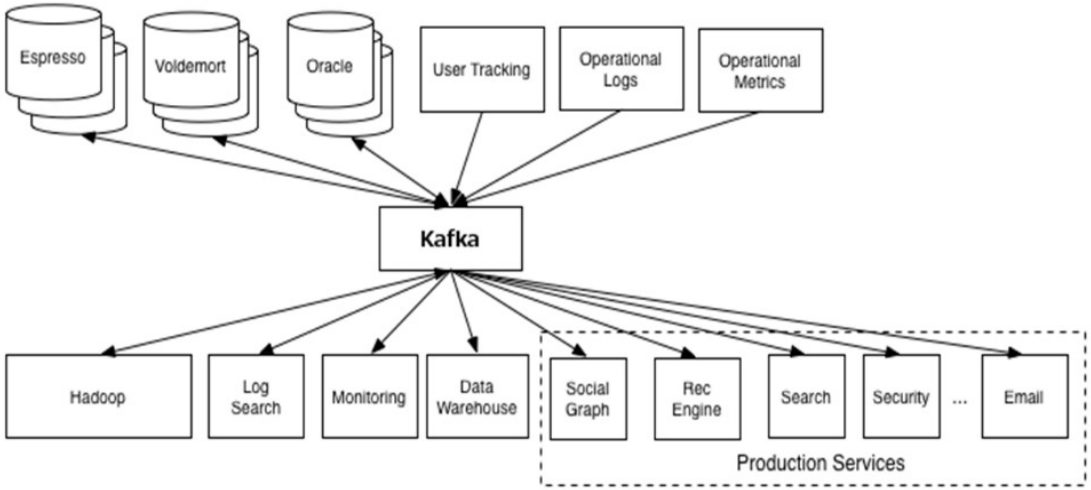




Producer-consumer architecture



Shared-queue architecture



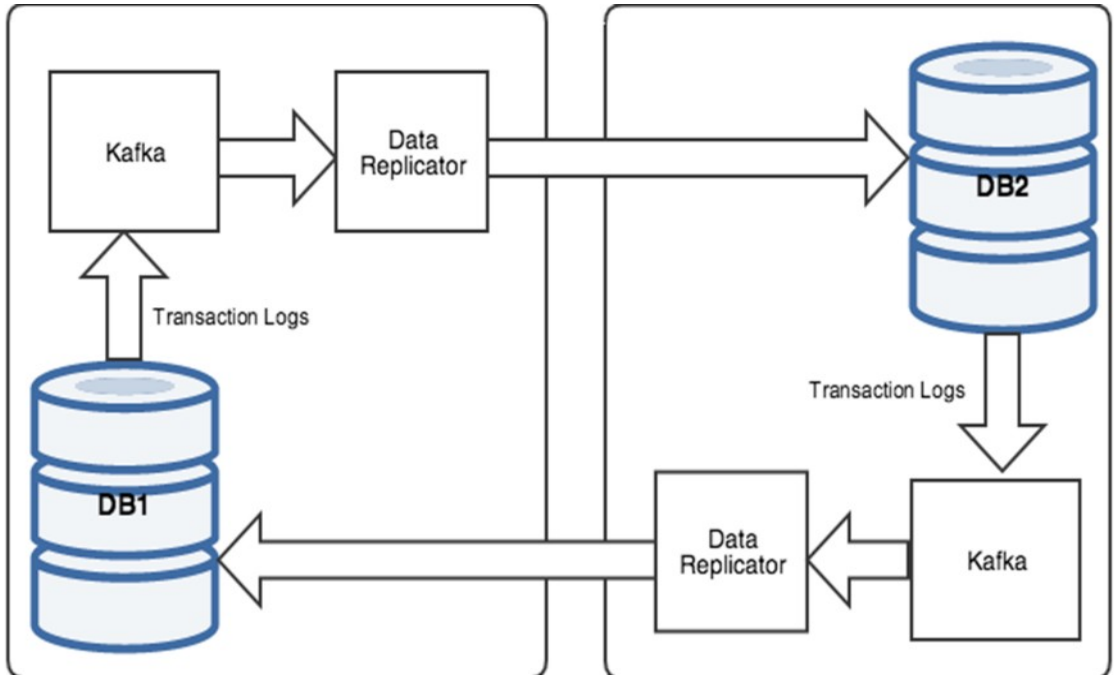
Kafka

Think core concepts!!! Kafka was developed as a shared-queue, and flume is producer-consumer. Later, flume also started incorporating concepts of shared queue

How is Kafka Used at LinkedIn

- Monitoring (inGraphs)
- User tracking
- Email and SMS notifications
- Stream processing (Samza)
- Database Replication
- How scalable is Kafka?
  - Over 1,300,000,000,000 messages are produced to Kafka everyday at LinkedIn
  - 300 Terabytes of inbound and 900 Terabytes of outbound traffic
  - 4.5 Million messages per second, on single cluster
  - Kafka runs on ~1300 servers at LinkedIn

Kafka @ LinkedIn (DB Replication)



- This also works for Master-Master replication

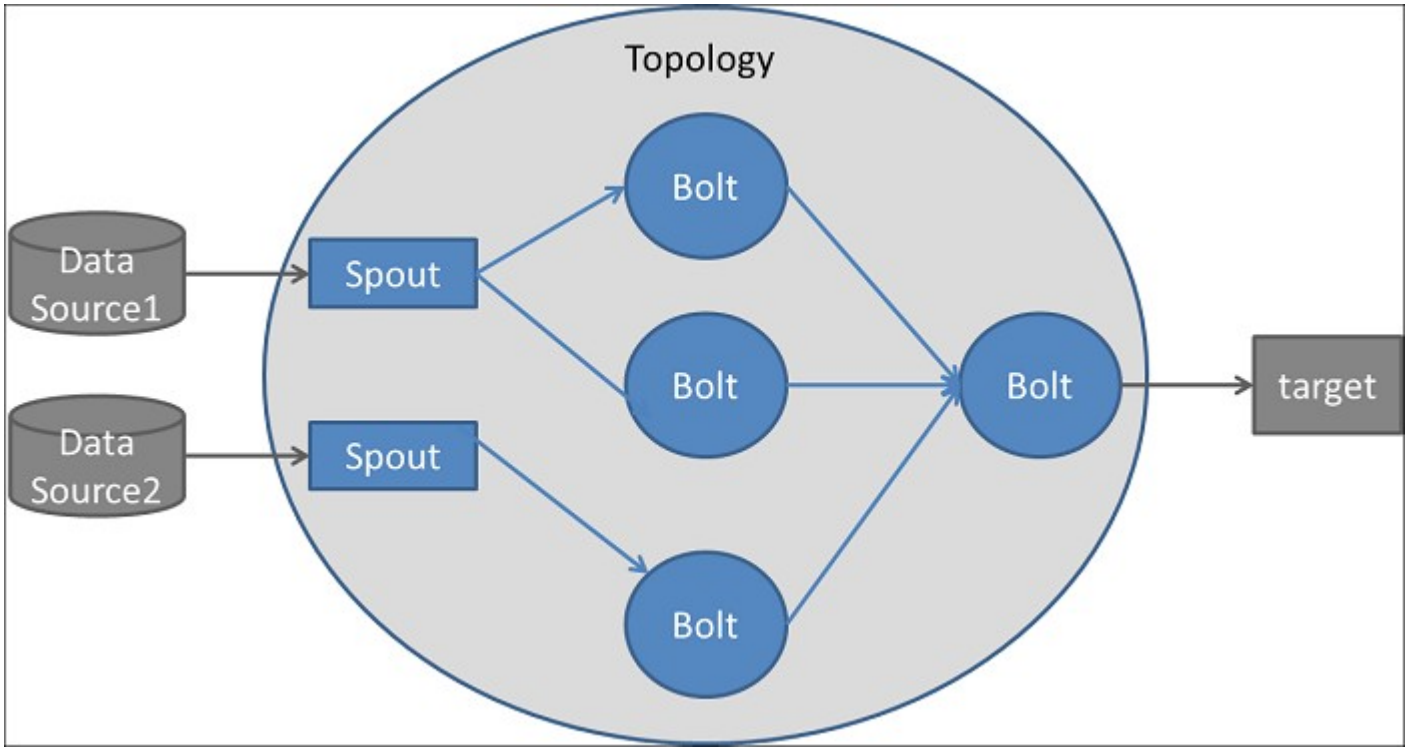
Agenda

- Data Ingestion
  - Flume
  - Sqoop
  - Kafka
- Stream Processing
  - Storm
  - Spark Stream
- Big Data Review
- Lambda Architecture

STORM

Concepts

- Streams
- Spouts
- Bolts
- Topologies

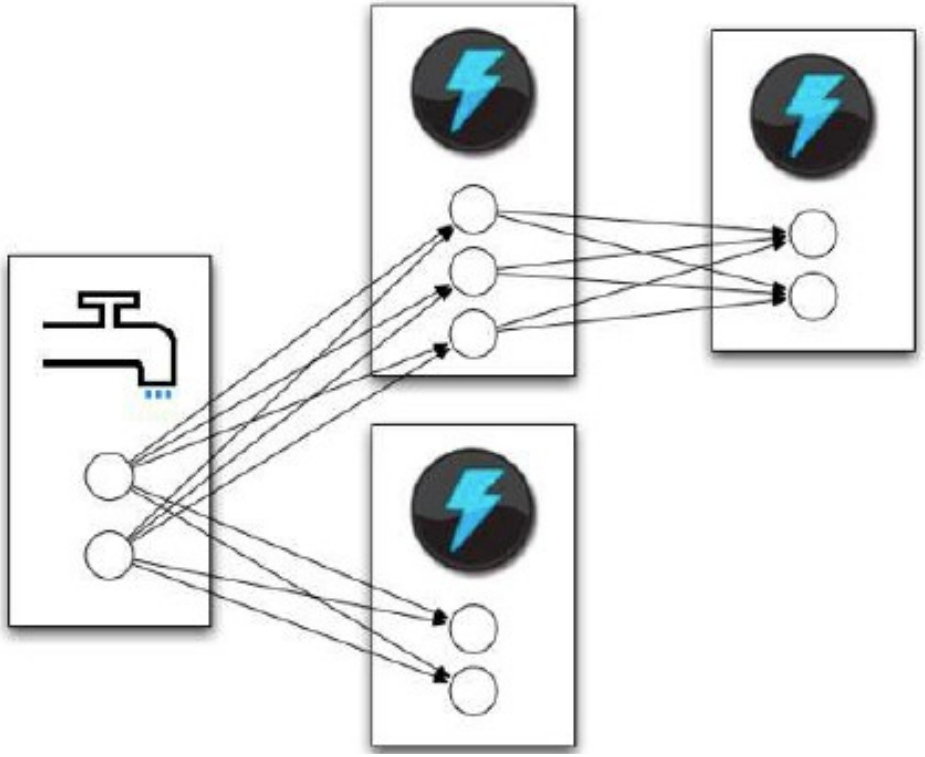


Streams Unbounded sequence of tuples

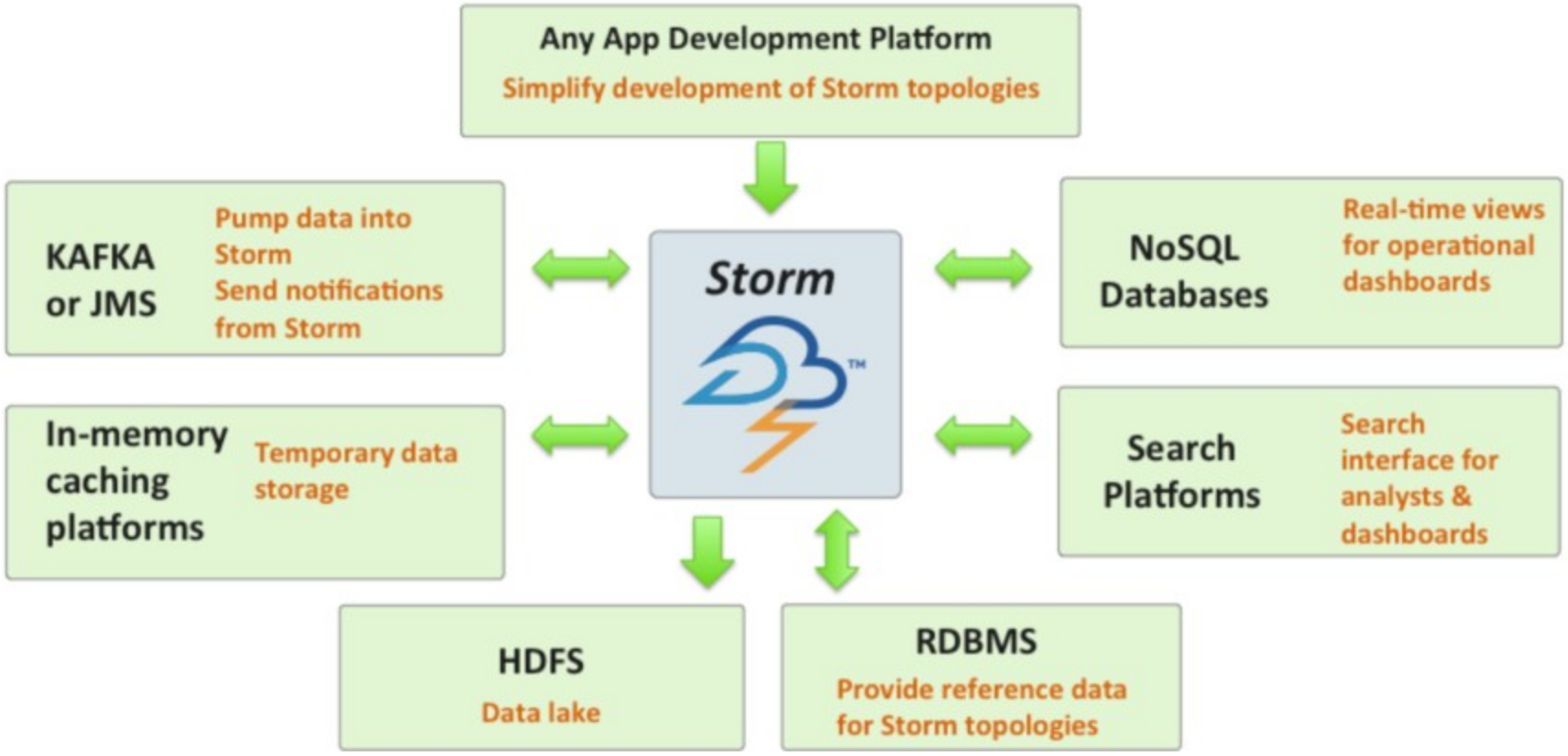
Bolts

- Functions
- Filters
- Aggregation
- Joins
- Talk to databases

Spouts and bolts execute as many tasks across the cluster



Stream Processing: Apache Storm



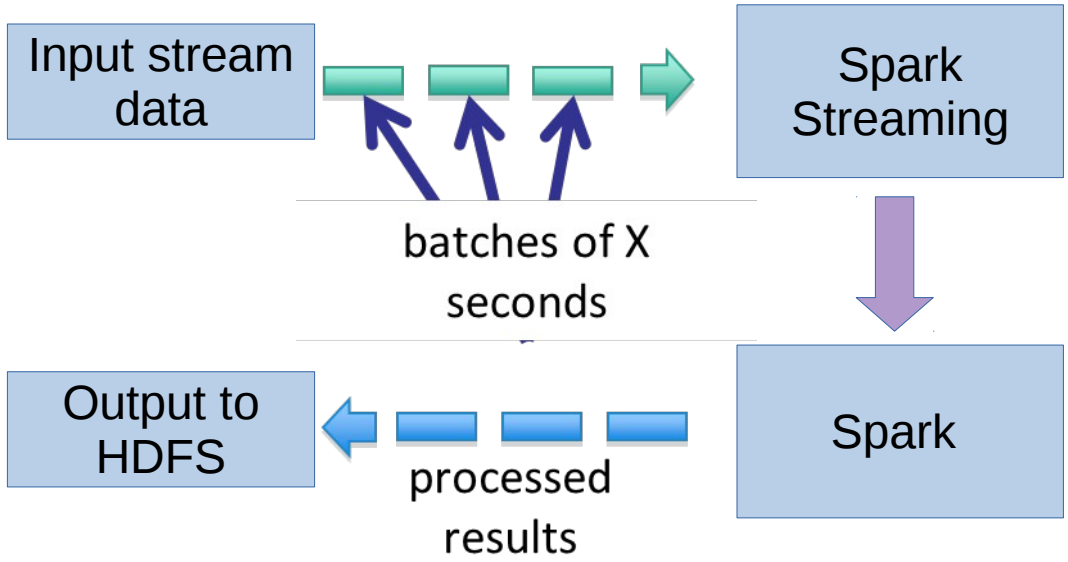
Some slides from Tathagatha Das presentation.

Spark Streaming

Run a streaming computation as a **series of very small, deterministic batch jobs**

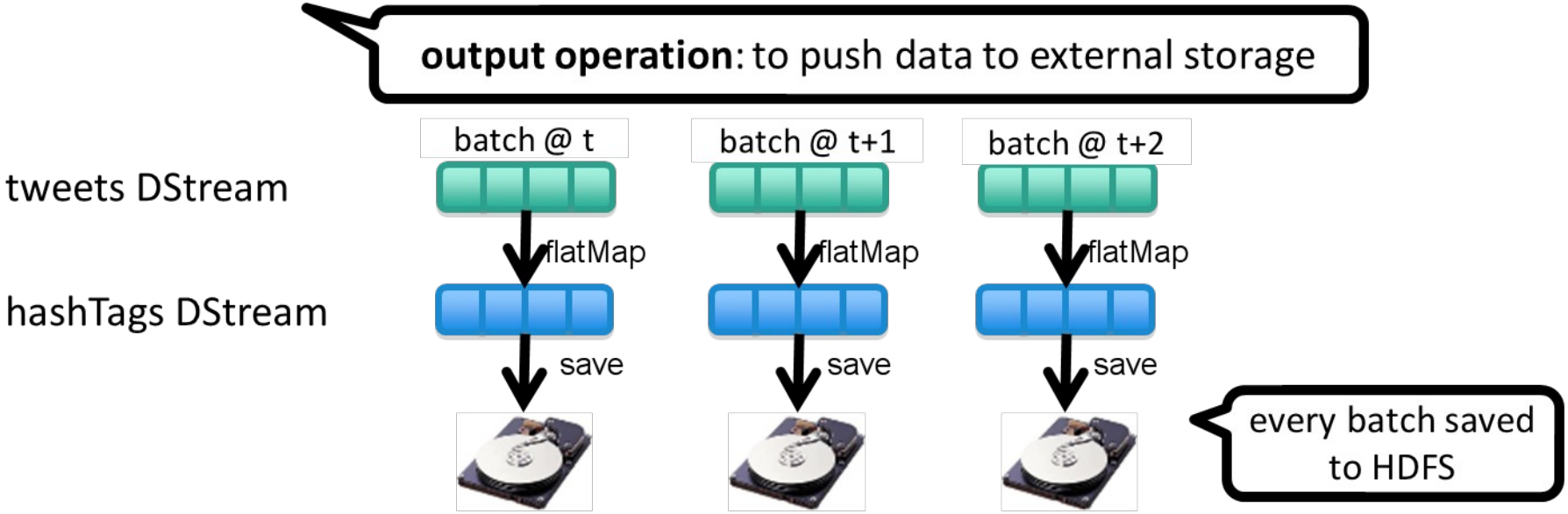
Batch sizes as low as ½ second, latency ~ 1 second

Potential for combining batch processing and streaming processing in the same system



Spark Streaming

```
val tweets = ssc.twitterStream (<Twitter username>, <Twitter password>)
val hashTags = tweets.flatMap (status => getTags(status))
hashTags.saveAsHadoopFiles("hdfs://...")
```



Spark Streaming: **Key concepts**

DStream – sequence of RDDs representing a stream of data
Twitter, HDFS, Kafka, Flume, ZeroMQ, Akka Actor, TCP sockets

Transformations – modify data from one DStream to another through:

- Standard RDD operations – map, countByValue, reduce, join,
- Stateful operations – window, countByValue and Window

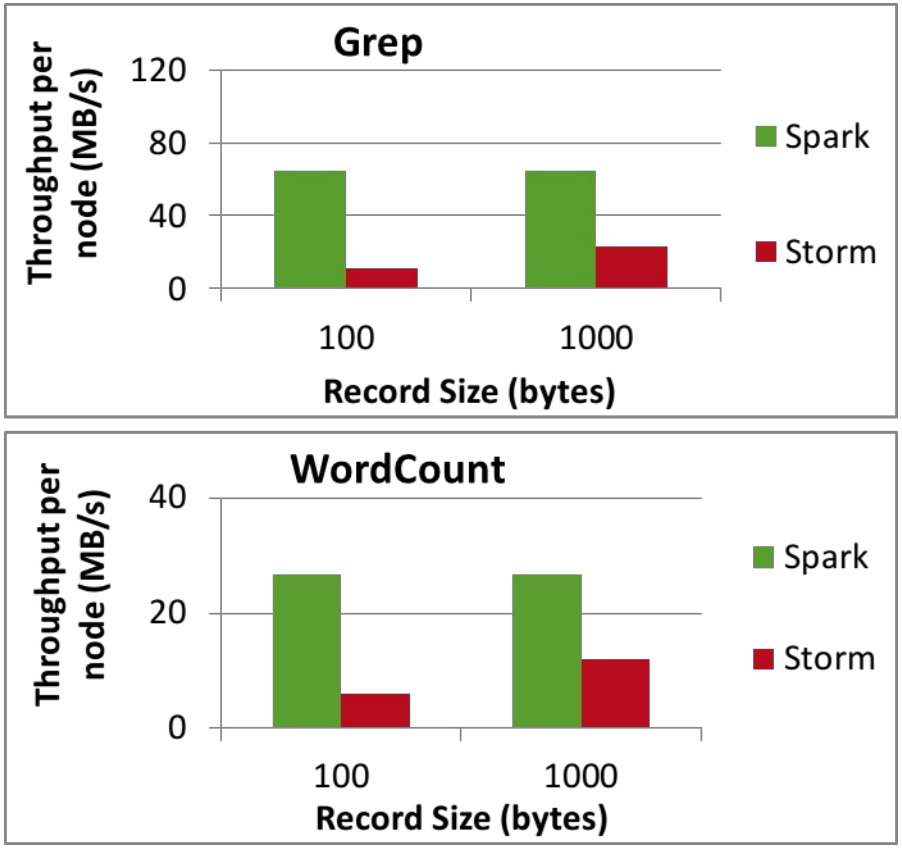
Output Operations – send data to external entity
saveAsHadoopFiles – saves to HDFS
foreach – do anything with each batch of results

Spark Streaming: **Comparison with Storm and S4**

Higher throughput than Storm

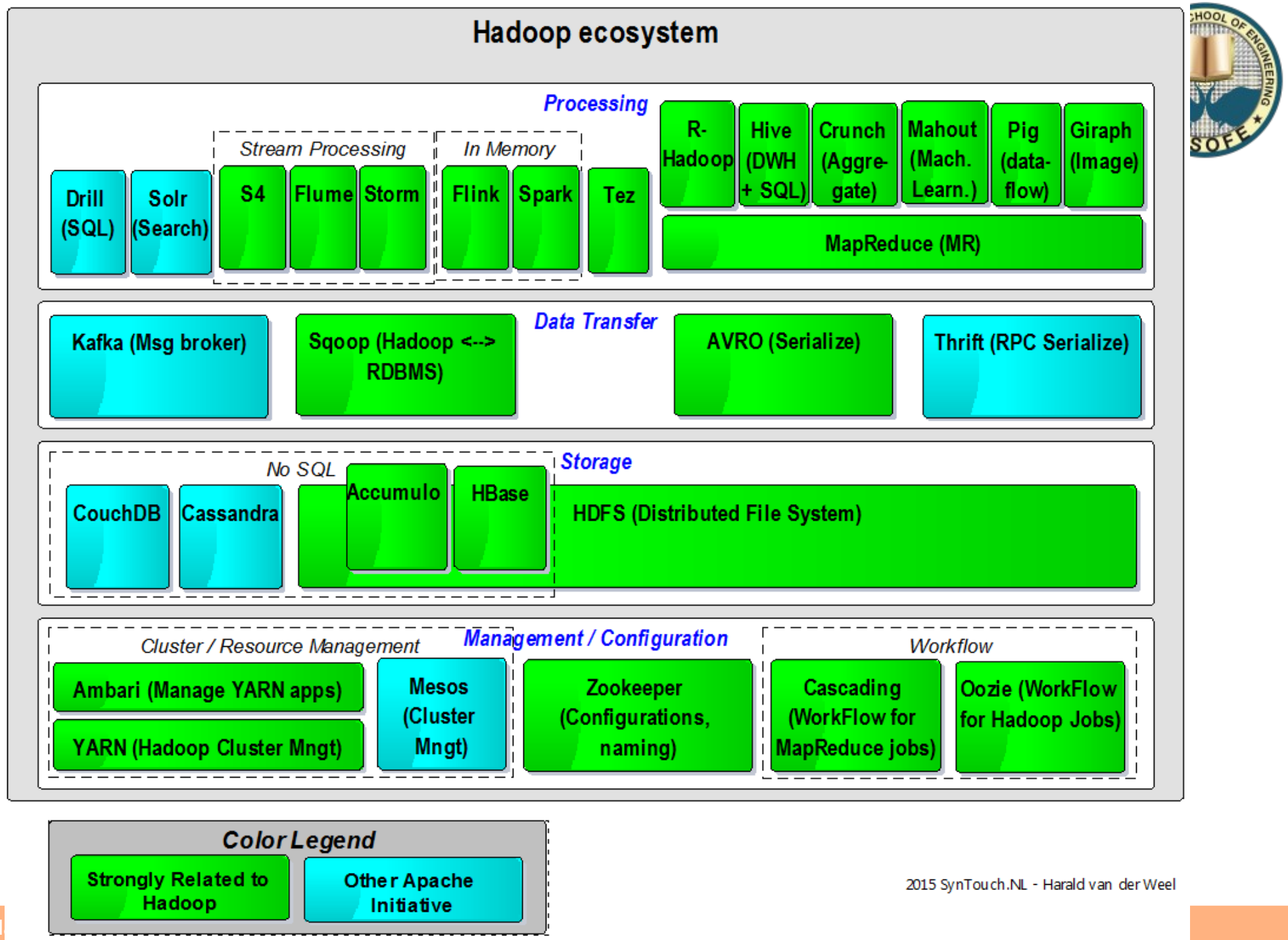
Spark Streaming: **670k** records/second/node

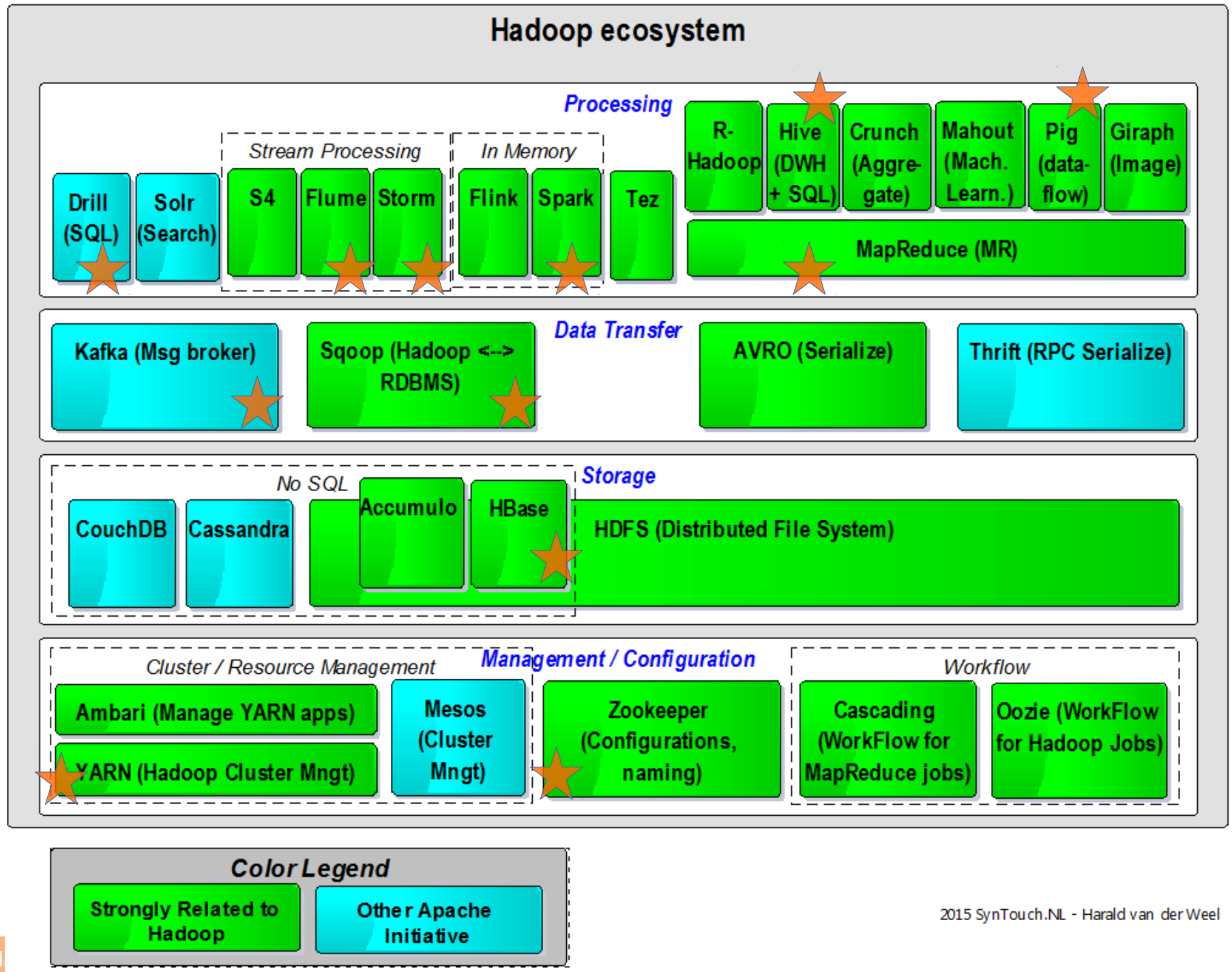
Storm: **115k** records/second/node



Agenda

- Data Ingestion
  - Flume
  - Sqoop
  - Kafka
- Stream Processing
  - Storm
  - Spark Stream
- Big Data Review
- Lambda Architecture





Notice Topics Carefully

- Class 1:
  - Different architectures
  - Transition from Databases to data warehouses and data lakes
  - Thinking of Large Jobs as Task Decompositions
  - How BigData is changing IT and business operations
- Class 2:
  - Hadoop: Storage
- Class 3:
  - Map-Reduce, YARN
  - Spark
- Class 4:
  - NoSQL:
    - Hbase
  - Scripting in Big Data:
    - Hive / Pig
    - Other tools
      - Impala
      - Spark SQL
      - Apache Drill
- Class 5:
  - Data Ingestion
    - Flume
    - Sqoop
    - Kafka
  - Stream Processing
    - Storm
    - Spark Stream
  - Big Data Review
  - Lambda Architecture

Fundamentals more important than tools!!!

Trends and tools WILL CHANGE VERY RAPIDLY

Deep ML & fundamental knowledge more valuable in long term

Select 1-2 tools, and be on top of these.

Open source blogs, sourceforge, and hands-on is the best teacher

Keep each other informed, and share!!!

HDFS	Cost-effective, replicated, distributed disk storage	Key aspects: NameNode, DataNode, Secondary Name Node, Blocks. Shadow Name Node may be present in conjunction with zookeeper. Heartbeats every 3 seconds help systems to keep track of each other
MapReduce	Distributed computing	Key aspects: Job tracker, task tracker. MapReduce2: MR application Master, Container
YARN	Resource manager (YARN acronym expands to Yet Another Resource Negotiator)	Resource Manager, Node Manager, Container, Application Master. Physical machines or parts of physical machines are allotted by YARN to specific app masters. App masters then request RM for containers and mange the application on these containers
Spark	Creates Resilient Distributed Datastructures (RDDs) to keep results of MR in memory	Avoids writing intermediate MR results to disk. Key components are drivers and workers. Allows objects to be shared across DataNodes, persisted on disk. Replication of RDDs in RAM and disk. Write-ahead logs to recreate RDDs
Flink	Distributed Stream processing	Can merge or split stream data. Run aggregates on a time window of streaming data
Flume	Unstructured data ingestion. By unstructured, we typically mean data that does not follow a schema	Source, Channel, Sink. Tiered architecture to absorb variable incoming and outgoing data throughputs.
Avro	Places metadata on top of unstructured stream data	
Chukwa	Injest logs Hadoop logs and create dashboards.	Runs the Hadoop Infrastructure Care Centre (HICC): https://chukwa.apache.org/docs/r0.5.0/hicc.html
NoSQL	A term for database technologies that relax ACID (Atomicity, Consistency, Isolation, Durability)	Get more throughput, cost-effectiveness by relaxing ACID. CAP (Consistency, Availability, Partition tolerance) theorem. Key Value pair DBs (Voldemort, DynamoDB), Document DBs (MongoDB), ObjectBased (Virtuoso), Columnar (HBase, BigTable, Cassandra, Vertica), Graph DBs: (Neo4J, flockDB)

Hortonworks vs Cloudera:
http://www.networkworld.com/article/2369327/software/comparing-the-top-hadoop-distributions.html

Hbase	Columnar DB that provides eventual consistency	Hmaster, HregionServer, HRegion, Store, Hlog. Rows, Column families, columns. A set of row keys are handled by HregionServer. A set of column families are handled by Hregion. Columns and time stamps are handled by Store. Hlog is first referred to avoid multiple reads
Sqoop	Injestion tool to import/export relational data into the Hadoop system	Importing of RDBMS done by storing schema as hive-compliant jar file, and data as flat files on HDFS. Exporting is done by using JDBC connections, schema file, and the data file. Sqoop2 is expected to bring: Job management, ODBC connections, security
Hive	SQL-like queries compiled into MR jobs	Allows simpler code, generally suitable for initial test runs. Requires mapping of all data into a schema.
Pig	SQL-like Data flows written in Pig Latin compiled into MR jobs	1.4 times slower than MR code written by humans. But orders of magnitude less code (4 lines of Pig vs 240 lines of MapReduce). Allows schema-less querying
Impala	One can think of Hive that does not use MR, but converts SQL-like queries into parallel operations on top of HDFS	Uses YARN and HDFS while bypassing MR
Spark SQL (formerly known as Shark)	Similar to Hive, but runs on Spark instead of MR	3-10 times faster on some benchmark queries
Drill	SQL Parser + execution engine that works with HDFS, Hbase, RDBMS	Uses Drillbit, YARN, zookeper to execute SQL like queries on multiple platforms
Oozie	Workflow management system	
Rhadoop	R installed on DataNodes	R-like commands written and executed in MapReduce fashion on DataNodes
Mahout		
SparkML		
Security components	Atlas, Kerberos, Knowx, Ranger	

http://www.bigdataanalyst.in/hive-interview-questions/

http://www.bigdataanalyst.in/pig-interview-questions-answers/

- What BigData is:
  - A set of tools to store, process, retrieve large amounts of data
  - Till a few years ago, this was VERY difficult or VERY expensive
  - Most of the components are open source, and you can contribute to it
- What BigData is not:
  - An end-to-end ML tool
  - A complete EDW solution
  - Plug-and-play system that does not require coding
- What you have to do:
  - Learn parts of Big Data that are relevant to your job
  - Understand how to put the parts together efficiently and quickly

Senior Exec Questions:	Where used?
Sales question:	How to sell?
Engineer question:	What gets a job?

- Where is BigData Used:
  - Consultants (Deloitte, HP, IBM, Cloudera, Hortonworks, MapR etc)
  - Tech Startups (Talkdesk, Loggly, Mezi, Myntra)
  - Verticals: Mobile, Retail, Banking, Scientific, Medical
- How BigData sells:
  - Licensing, Easy to use administration tools (IBM, Cloudera, RevolutionR)
  - Pay-As-You-Go or PAAS/SAAS (AWS, HDInsights)
  - Support and service (Hortonworks, Cloudera, MapR)
- What gets the job:
  - Become an open source committer to a feature (ANY feature)
  - Start your own project

ML in Big Data

- Amazon (Or any other retailer)
- Fake Reviews
  - Someone else sells a product close to yours but has way better reviews
- Prevent counterfeit product
  - Inferior product, different branding but same look/packaging
- Hijacked product
  - Copied product, same look, same brand, total copy
  - Can be detected by Amazon, but the copier can open a new store with a different Amazon ID

http://video.cnbc.com/gallery/?video=3000532539

ML in Big Data

- Social media analytics or Mining “Social Noise”
- Collecting social data:
  - Twitter hash tags, Facebook, Likes of a specific product or image
  - Track specific content using twitter, facebook, linked in and similar APIs
- Analyze Comments/content
  - Use ML or NLP tools to analyze the specific content

Lambda Architecture: Good at Batch + Good at Real TIME

Lambda Architecture: Overview

A. All data is sent to **both** the **batch and speed layer**

B. Master data set is an **immutable, append-only** set of data

C. Batch layer **pre-computes** query functions from scratch, result is called Batch Views. Batch layer **constantly re-computes** the batch views.

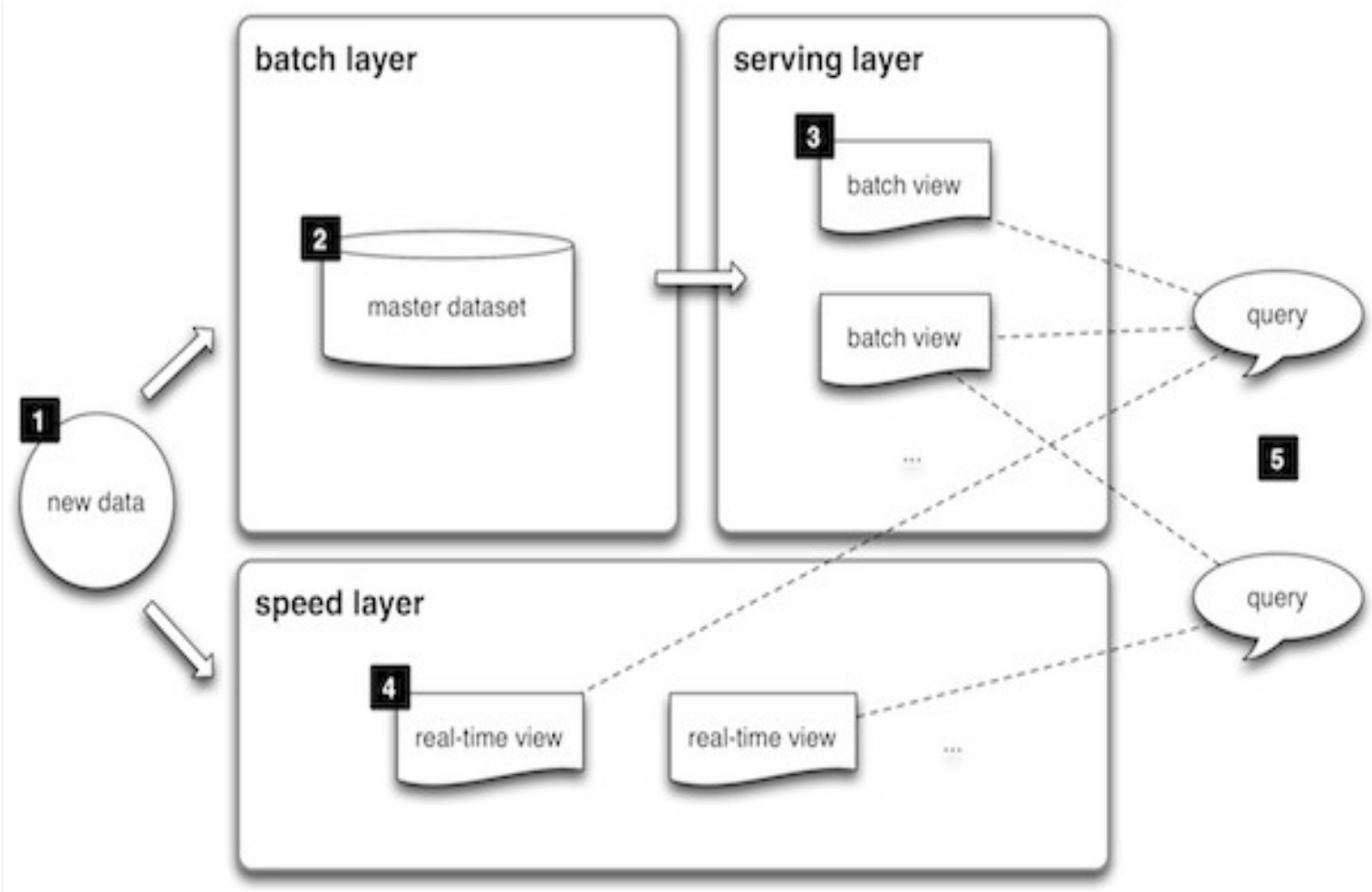
D. Batch views are **indexed** and **stored** in a **scalable database** to get particular values very quickly. Swaps in new batch views when they are available

E. Speed layer **compensates** for the high latency of updates to the Batch Views

F. Uses fast **incremental algorithms** and read/write databases to produce real-time views

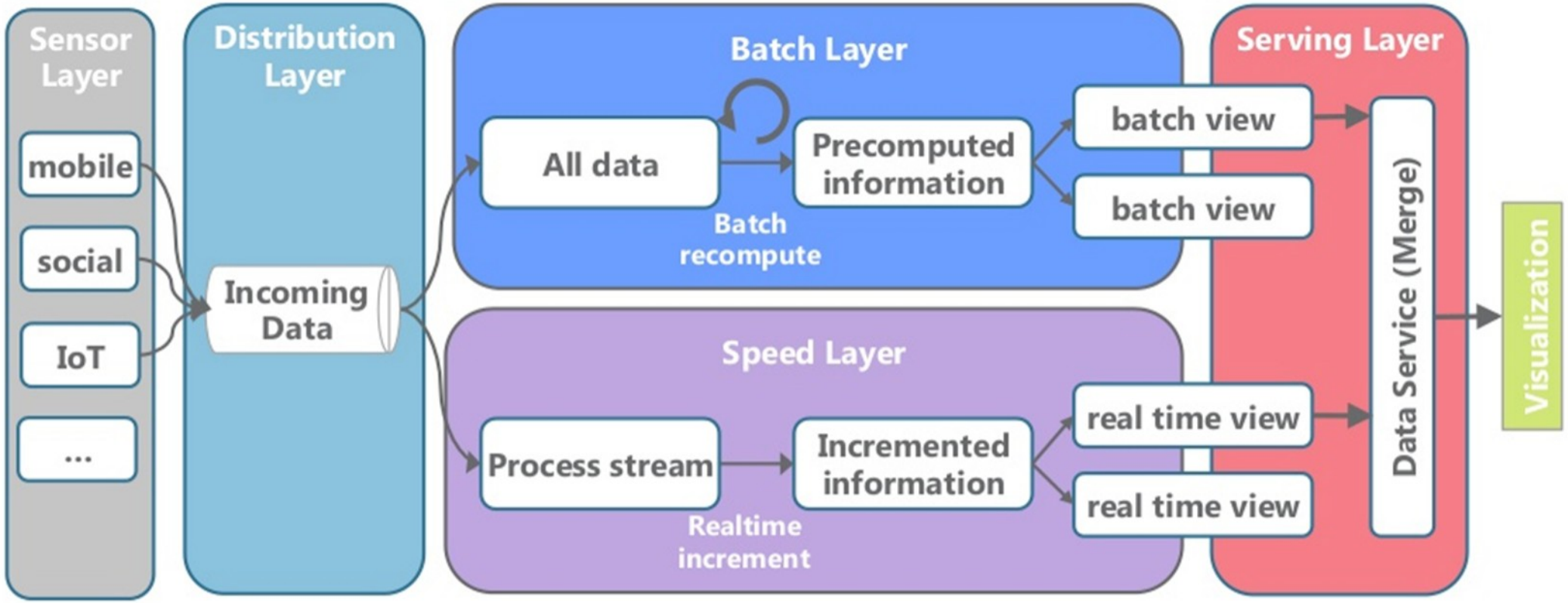
G. Queries are resolved by getting results from **both** batch and real-time views

Lambda Architecture



Not everything is a formula, there are many one-off solutions: http://lambda-architecture.net/

Lambda Architecture

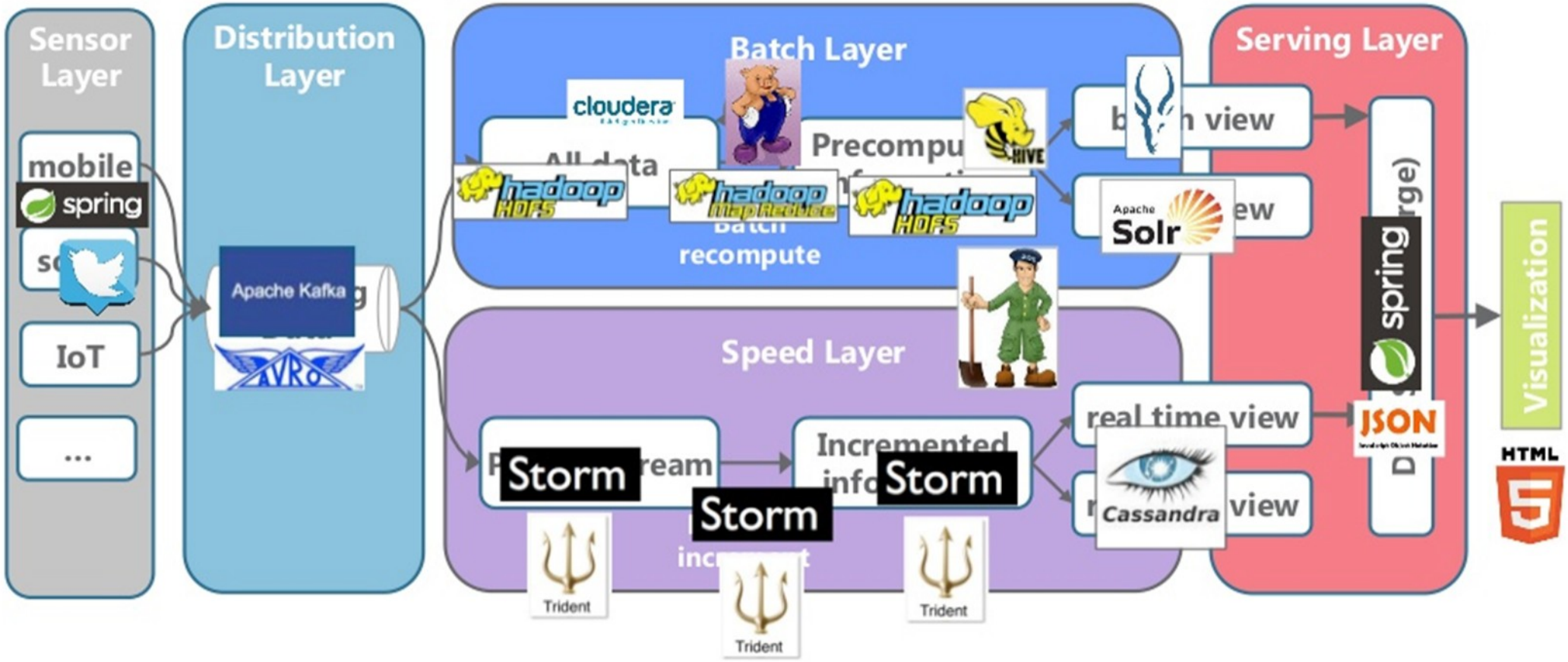


Adapted from: Marz, N. & Warren, J. (2013) Big Data. Manning.

http://lambda-architecture.net/

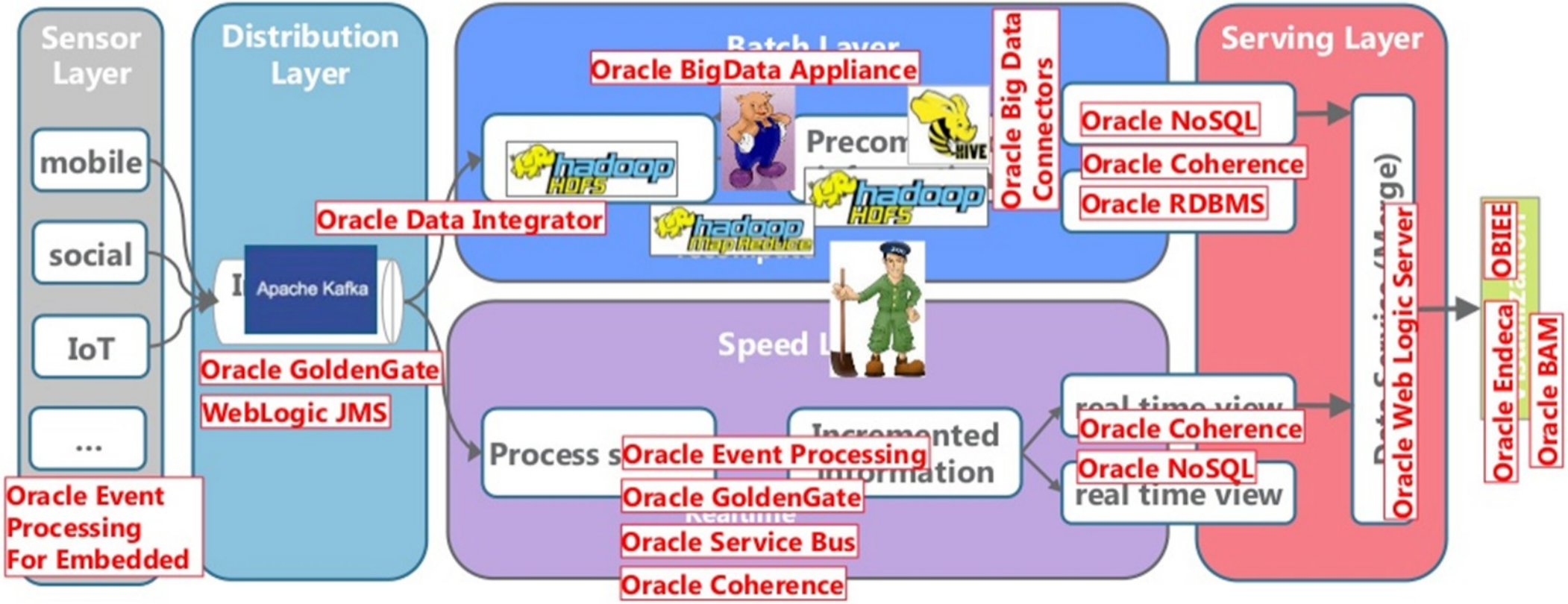
https://spark-summit.org/2014/wp-content/uploads/2014/07/Lambda-Architecture-Jim-Scott..pdf

Lambda Architecture: Open Source Tools



Lambda Architecture: Proprietary Implementations

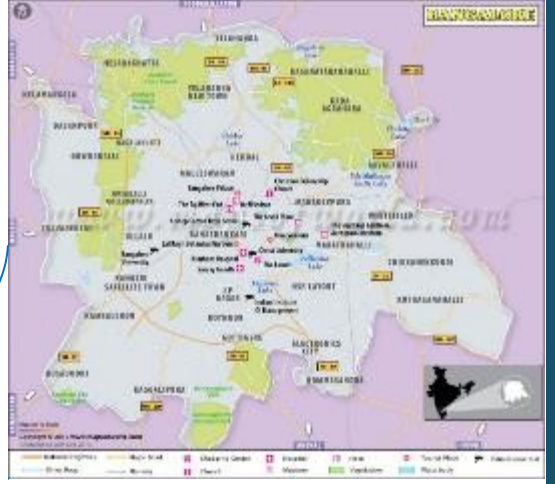
Possible implementation with Oracle Product stack



Other Real-Time Architectures

- Google Millwheel http://research.google.com/pubs/pub41378.html
- Amazon Kinesis http://aws.amazon.com/kinesis
- Azure Stream Analytics http://download.microsoft.com/download/6/2/3/623924DE-B083-4561-9624-C1AB62B5F82B/real-time-event-processing-with-microsoft-azure-stream-analytics.pdf
- Facebook Puma http://www.slideshare.net/cloudera/building-realtime-big-data-services-at-facebook-with-hadoop-and-hbase-jonathan-gray-facebook

Inspire...Educate...Transform.



HYDERABAD

Office and Classrooms

Plot 63/A, Floors 1&2, Road # 13, Film Nagar,
Jubilee Hills, Hyderabad - 500 033
+91-9701685511 (Individuals)
+91-9618483483 (Corporates)

BENGALURU

Office

Incubex, #728, Grace Platina, 4th Floor, CMH Road,
Indira Nagar, 1st Stage, Bengaluru – 560038
+91-9502334561 (Individuals)
+91-9502799088 (Corporates)

Classroom

KnowledgeHut Solutions Pvt. Ltd., Reliable Plaza,
Jakkasandra Main Road, Teacher's Colony, 14th Main
Road, Sector – 5, HSR Layout, Bengaluru - 560102

Social Media

Web:	http://www.insofe.edu.in
Facebook:	https://www.facebook.com/insofe
Twitter:	https://twitter.com/Insofeedu
YouTube:	http://www.youtube.com/InsofeVideos
SlideShare:	http://www.slideshare.net/INSOFE
LinkedIn:	http://www.linkedin.com/company/international-school-of-engineering

This presentation may contain references to findings of various reports available in the public domain. INSOFE makes no representation as to their accuracy or that the organization subscribes to those findings.