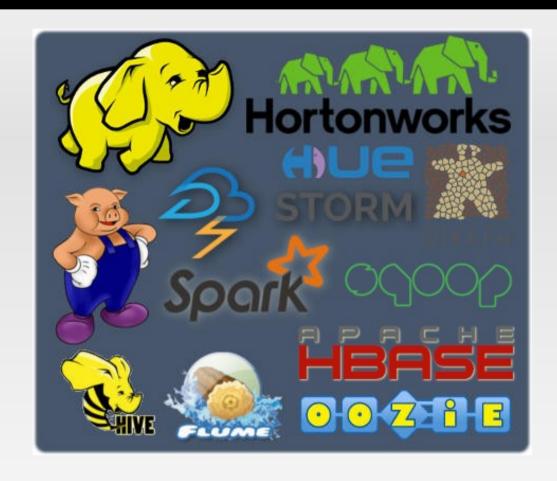


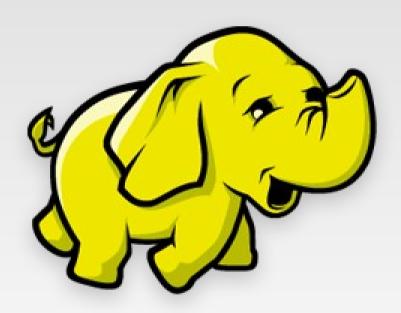
#### Big Datas Økosystem



Terje Berg-Hansen Stavanger 22. mai 2019



# Hadoop





#### Historien





- Det var en gang et firma som het Google, som ville laste ned hele internett og kjøre analyser på det. Derfor lagde de:
  - Google FS og MapReduce
  - ... og på den måten tjente de alle pengene i hele verden
- Google vil sjelden open-source det de lager, men de vil gjerne beskrive det, så: «Google wrote a paper» er blitt en gjennomgående inspirasjon for nye open source teknologier.
- Doug Cutting (Yahoo!) leste Googles papers i 2004 og skrev en Java-app med et distribuert filsystem og en MapReduce-implementering sammen med Mike Cafarella ved Univ. Of Michigan, kalte det Hadoop etter sin sønns gule tøyelefant og etablerte det som en del av Apache Nutch og Apache Lucene.
- (Prof. Cafarella is known for his smooth, baritone speaking voice wikipedia)



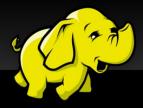
### Hva er Hadoop?



- Hadoop er et system for å prosessere «mind-boggingly large amounts of data» - også kalt «Big Data»
- Kjernen i Hadoops økosystem var opprinnelig to komponenter:
  - HDFS Hadoop Distributed File System et distribuert filsystem som kan spenne over tusenvis av noder med petabytes av data.
  - MapReduce parallell prosessering av data på de maskinene hvor data er lagret. Jobber mappes ut, prosesseres lokalt og resultatene stokkes sammen og reduseres til et sluttresultat
- I 2009 kunne Hadoop sortere en terrabyte med data på 62 sekunder.



# Hvem bruker Hadoop?

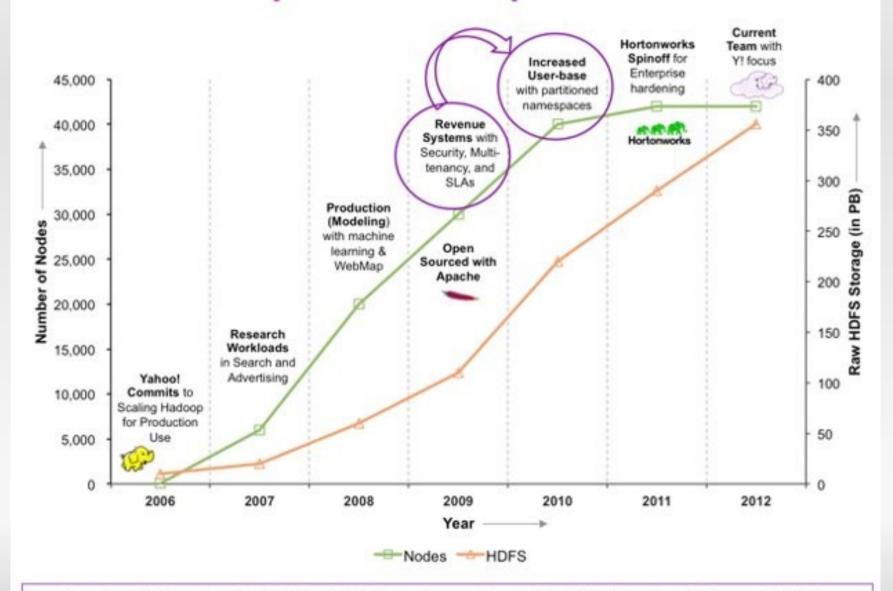


- Hadoop brukes nå av «alle», f.eks. :
- Yahoo!
  - More than 100,000 CPUs in >40,000 computers running Hadoop
  - Our biggest cluster: 4500 nodes
  - >60% of Hadoop Jobs within Yahoo are Apache Pig jobs.



### Hadoop-utviklingen hos Yahoo!

#### Multi-tenant Apache Hadoop at Yahoo!



Footprint: 42,000 nodes, 365PB HDFS, 10M daily slot hours | Usage (Apr'13): 411 projects, 805 users, 13.2M jobs



#### **HDFS 101**

 HDFS er Hadoops filsystem, og oppfører seg på mange måter likt et Linux-filsystem:

```
terje@terjelap:~$ hdfs dfs -ls /
Found 6 items
drwxrwxrwx
                                       0 2014-10-24 20:16 /benchmarks
            - hdfs
                    supergroup
            - hbase hbase
                                       0 2014-10-24 20:16 /hbase
drwxr-xr-x
drwxr-xr-x - solr solr
                                       0 2014-10-24 20:16 /solr
drwxrwxrwt - hdfs supergroup
                                       0 2014-10-25 23:05 /tmp
drwxr-xr-x - hdfs supergroup
                                       0 2014-10-24 20:20 /user
drwxr-xr-x
            - hdfs
                    supergroup
                                       0 2014-10-24 20:15 /var
```

+ HDFS er beregnet for filer i størrelsesorden terabytes og petabytes. HDFS splitter filene opp i blokker på eks. 256 Mb hver (konfigurerbart), som så distribueres over klynger med opptil 4500 noder. HDFS lager 3 kopier av hver blokk (konfigurerbart), og lagrer dem på ulike noder i klyngen. En «NameNode» holder orden på blokkene (spof) som fysisk lagres på «DataNoder».



### MapReduce 101

- MapReduce er en modell for å gjøre distribuert programmering håndterbart (ved bl.a. å unngå at man må håndtere distribuert «state»).
- Alle data behandles som nøkler (keys) og verdier (values)
  - Word Count = Hadoop «Hello world»:
    - Vi teller opp hvor mange ganger hvert ord opptrer i en tekst ved å lage «mappers» som teller hvert ord, og sender dette som nøkkel/verdi-par til «reducers» som summerer opp resultatet.
- MapReduce er på mange måter Hadoops «assembly language». Så folk ble etter hvert leie av å skrive tunge MapReduce-jobber.



# MapReduce sucks?

Grise-leie, faktisk....



# Apache Pig





# En gris ankommer



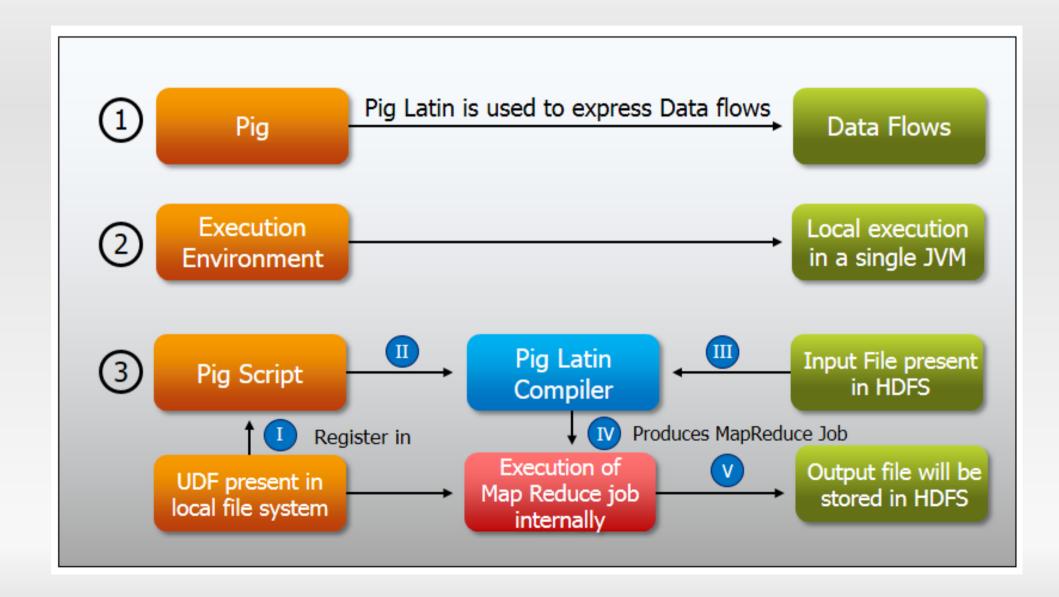
- MapReduce er java-applikasjoner som er tunge og omstendelige å skrive. Derfor lagde Yahoo et enklere språk de kalte «Pig» som forenklet koden betraktelig:
  - «Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs.»
- Istedenfor å skrive to sider med java-kode for MapReduce, kan man nå skrive et par linjer med «Pig Latin» som Pig kompilerer til MapReduce-jobber:
  - A = LOAD 'student' USING PigStorage() AS (name:chararray, age:int, gpa:float);
  - B = FOREACH A GENERATE name;

```
DUMP B;(John)(Mary)(Bill)(Joe)
```



# **Apache Pig**







# Mer griseprat...



- Med Pig kan man bl.a. skrive «Pig Latin» i skallet «Grunt», lagre det som skriptfiler, lage egne funksjoner (UDF) og dele dem gjennom en offisiell «PiggyBank».
- LinkedIn bruker Pig til det meste og har delt sine funksjoner gjennom biblioteket «DataFu»
- Twitter har delt sine UDFer (som de bruker daglig) gjennom: «elephant-bird»
- «Pygmalion» knytter Pig til Apache Cassandra.
- «Piglet» er Pig for Ruby
- «PigPy» er Pig for Python
- «PigPen» er en av flere Eclipse-plugins for Pig.
  - etc....



# **Apache Hive**





# ... og en bikube



- For å lette bruken av Hadoop for de med SQL-kunnskaper, lagde Facebook datavarehus-applikasjonen «Apache Hive», som nå er en de facto standard for Hadoop datalagring.
- «HiveQL» er et SQL-lignende språk som lar deg lagre og hente ut data via MapReduce-jobber (e.l.) på en HDFS-klynge.
  - Eksempel HiveQL:

```
hive> CREATE TABLE pokes (foo INT, bar STRING); hive> SELECT foo FROM invites WHERE ds='2018-08-15';
```

 Mens Pig er en «datafabrikk», ideell til ELT og datamanipulering, er Hive et «datavarehus» ideelt til lagring av data. Data kunne f.eks. opprinnelig ikke endres etter at de ble lagret i Hive, men måtte slettes og lagres på nytt i annen form.



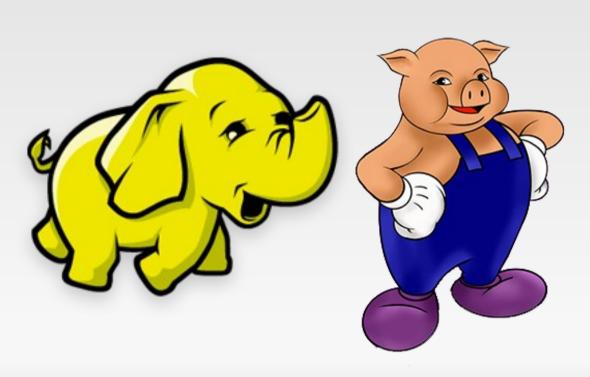
# **Apache HCatalog**

- HCatalog ble laget for å sentralisere et metadatakartotek, slik at samme metadata kan brukes for MapReduce, Streaming, Pig, Hive, Cascading etc.
  - «HCatalog gi meg disse dataene med Pig»
  - «HCatalog gi meg disse dataene med Hive»
    - Osv...
- HCatalog gir en felles mekanisme for schema og datatyper, samt en tabell-abstraksjon, slik at brukerne ikke behøver å tenke på hvor eller hvordan data er lagret
- HCatalog er nå en integrert del av Hive



### ... problem solved?

 Så en elefant, en gris og en bisarr kombinasjon av en elefant og en veps slo seg sammen og løste alle big data-problemene våre?





Vel, nesten ....



# Begresningene til MapReduce

- MapReduce er fantastisk til batch-kjøring av gigantiske jobber hvor du har tid til å vente en halvtime på resultatet.
  - Men ikke til interaktiv jobbing med data.
- Siden både Pig og Hive kompileres ned til MapReduce-jobber, var det behov for mer interaktiv teknologi.
- Interaktive databaser, alternative prosesserings-motorer, streaming-grensesnitt etc. har vokst frem gjennom årene til å bli et stort og omfattende økosystem av programmer som alle bruker (eller kan bruke) HDFS og/eller MapReduce.



# Ressurs-styring

- YARN ble laget for å fordele ressurser mellom ulike programmer i klyngen
- Oozie ble laget for å lage kjøreplaner for jobbene programmene utfører
- Zookeeper ble laget for å ivareta programmenes tilstand og konfigureringsdata



### YARN – MapReduce 2.0

- Apache Hadoop NextGen MapReduce (YARN)
  - «MapReduce has undergone a complete overhaul in hadoop-0.23 and we now have, what we call, MapReduce 2.0 (MRv2) or YARN»
- Hovedideen bak YARN er å splitte de to hovedfunksjonene til Hadoops JobTracker: resource management og jobbscheduling/overvåking, og la dem håndteres av separate daemons.
- Yarn har således én global ResourceManager (RM) og i tillegg en ApplicationMaster (AM) per applikasjon. En applikasjon er enten en enkelt MapReduce jobb eller en DAG av jobber.

YARN = Yet Another Resource Negotiator



### YARN – MapReduce 2.0

#### Arbeidsfordelingen:

- ResourceManager (én per klynge) og NodeManager(s) (én per slavenode), utgjør tilsammen rammeverket for data-prossessering. ResourceManager er autoriteten som mekler ressurser mellom alle applikasjonene i systemet.
- ApplicationMaster (én per applikasjon) er et rammeverkspesifikt bibliotek som har til oppgave å forhandle med ResourceManager om ressurser og jobbe sammen med NodeManager(s) for å eksekvere oppgaver og overvåke dem.



# **Apache Oozie**





## Apache Oozie



- Apache Oozie er et «workflow processing system» som lar brukere definere en serie med jobber skrevet med diverse teknologier – som Map Reduce, Pig og Hive – og så lenke dem sammen i kjeder som utføres til angitte tider.
- Oozie er mao en avansert «scheduler» for Hadoop som f.eks. lar en jobb starte først etter at en annen er avsluttet osv.



# Apache Zookeeper

- Zookeeper er en sentralisert tjeneste for lagring av konfigureringsinformasjon og navn, og som organiserer distribuert synkronisering og gruppe-tjenester for en Hadoopklynge.
- Zookeepers «Raison d'Etre»:
  - «All of these kinds of services are used in some form or another by distributed applications. Each time they are implemented there is a lot of work that goes into fixing the bugs and race conditions that are inevitable. Because of the difficulty of implementing these kinds of services, applications initially usually skimp on them ,which make them brittle in the presence of change and difficult to manage. Even when done correctly, different implementations of these services lead to management complexity when the applications are deployed.»
- MAO Zookeeper er lav-nivå programvare som håndterer kommunikasjonsvanskelighetene i et distribuert system.



# Databaser og lagringsformater





# Databaser og lagringsformater

- Integrert i Hadoop
  - Databaser: HBase / Phoenix, Accumulo, Druid,
  - Lagringsformater: ORCFile, Avro og Parquet
- Kan integreres med Hadoop
  - Cassandra, MongoDB, Hawq, PrestoDB
    - + Tradisjonelle RDBMSer som Microsoft SQL Server m.m.



# Apache HBase





#### **Apache HBase**



- HBase begynte som et prosjekt hos firmaet Powerset (kjøpt av Microsoft for \$100 Mill. i 2008) for så å bli donert til Apache Software Foundation.
- Apache HBase er en open-source, distributert, kolonnebasert ikke-relasjonell database skrevet i Java etter modell av Google's Bigtable.
- «Use Apache HBase when you need random, realtime read/write access to your Big Data. This project's goal is the hosting of very large tables -- billions of rows X millions of columns -- atop clusters of commodity hardware»
  - StumbleUpon driver bl.a. det meste av nettstedet sitt live fra HBase
  - Facebook bruker bl.a. HBase til messages-infrastrukturen sin og likes
  - Adobe, Filmweb, Meetup, Ebay og Twitter er andre store brukere



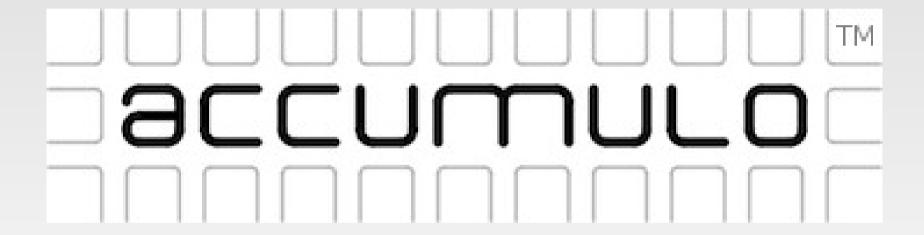
#### **Apache HBase**



- HBase fra kommandolinjen: Opprette en tabell og legge inn data.
  - Opprett tabell med 'create' og spesifiser tabellnavn og kolonnefamilie.
    - hbase> create 'terje', 'kontakter'0 row(s) in 1.2200 seconds
  - List informasjon om Tabellen med 'list':
    - hbase> list 'terje'
       TABLE
       terje
       1 row(s) in 0.0350 seconds
       => ["terje"]
  - Sett inn litt data i tabellen med 'put':
    - hbase> put 'terje', 'row1', 'kontakter:navn', 'Donald Duck' 0 row(s) in 0.1770 seconds
    - hbase> put 'terje', 'row1', 'kontakter:adresse', 'Andebyveien 45' row(s) in 0.0160 seconds
    - hbase> put 'terje', 'row1', 'kontakter:epost', 'donald@disney.com'
       0 row(s) in 0.0260 seconds



#### **Apache Accumulo**





- NSA «forket» HBase i 2008 og la til funksjonalitet som de hadde behov for (row-level sikkerhet osv.). Utviklerne fikk ikke lov til å «commit'e tilbake til community'en», før i 2011, hvor de donerte prosjektet til ASF i 2011 undern navnet «Accumulo».
- Så Accumulo er HBase slik den så ut i 2008 pluss det NSA hadde behov for – videreutviklet parallelt med HBase i 6-7 år.

Fra Accumolo på nett:

«Google published the design of BigTable in 2006. Several other open source projects have implemented aspects of this design including HBase, Hypertable, and Cassandra. Accumulo began its development in 2008 and joined the Apache community in 2011.»



# **Apache Cassandra**





### Apache Cassandra



cassandra

- Apache Cassandra ble utviklet av Facebook og donert til Apache Software Foundation.
- Høy tilgjengelighet, justerbar konsistens
- Alle noder er like ingen er primær/sekundær
- Nodene kommuniserer via «gossip» og «snitch»
- Godt integrert med Hadoop
- Brukes ikke lenger av Facebook, men av bla.
  - Apple (ITunes på 70 000 servere)
  - Netflix og Spotify (til metadata, spillelister osv)
  - Finn.no
  - Norsk Hydro (sensordata fra smelteverk)



### MongoDB



- MongoDB er en dokument-orientert database laget av Dwight Merryman - som nå er verdens femte mest brukte database.
  - Eget distribuert filsystem
  - Databaser med Dokument-samlinger
  - Replika-sett med 3 noder i hvert sett
  - Sharding kan skrus på for hver database
  - Bruker JSON/BSON til dokumenter og lagring
  - Elsket av utviklere, ikke så mye av driftere



# NoSQL-databaser og Hadoop

- Apache Cassandra er godt integrert med Hadoop og Spark bl.a. gjennom et samarbeid mellom DataStax og Hortonworks.
- MongoDB Inc samarbeider både med Cloudera og Hortonworks for integrasjon med CDH og HDP – og har bl.a. laget en mongo-hadoop connector.
- Apache Hawq (PostgreSQL/Greenplum) er integrert bl.a. gjennom Ambari og HDP
- Apache Trafodion (HP), AsterixDB (JSON) m.fl. er integrert via Yarn



# Lagringsformater

- Hadoop støtter en rekke ulike lagringsformater, hvorav noen er utviklet spesielt for Hadoopklynger, som f.eks.:
  - SequenceFile
  - RCFile
  - ORCFile
  - Avro
  - Parquet



# Apache Avro





#### **Apache Avro**



- Apache Avro er et system for data-serialisering, skrevet av Doug Cutting, Hadoops «opprinner». Avro tilbyr:
  - Rike data-strukturer Kompakt, raskt, binært dataformat En container-fil til å lagre store datamengder - Remote procedure call (RPC).
  - Avro har enkel integrasjon med dynamiske språk. Man trenger ikke generere kode for å lese eller skrive datafiler eller for å bruke eller implementere RPC protokoller. Skjemaet som ble brukt da data ble skrevet er lagret i begynnelsen av dokumentet, slik at alle Avro-filer er selvbeskrivende.
  - Avro skiller seg fra Thrift og Protocol Buffers ved bl.a:
     Dynamiske datatyper, utaggede data (siden skjema er tilstede), ingen manuelle felt IDer: Når et skjema endres vil både det gamle og det nye være tilgjengelige når data prossesseres.



# Apache Parquet





## Apache Parquet



- Parquet er et kolonne-orientert lagringsformat for alle komponenter i Hadoops økosystem.
- Parquet er utviklet for komplekse nøstede datastrukturer, basert på Dremels «record shredding» og «assembly algorithm»
- Parquet er laget for å støtte svært effektive komprimerings- og «encoding»-schemes.
- ... og ikke minst har formatet raskt blitt en de facto standard innenfor mange teknologier og organisasjoner.



### Migrering av Data

- Teknologier for å overføre data til Big Data klynger fra:
  - Relasjonelle databaser
  - Loggfiler
  - Datastrømmer, f.eks. fra sensorer
  - Dokumenter Regneark, Tekstdokumenter, Pdf
  - Mediafiler audio, video



## **Apache Flume**





#### **Apache Flume**



- Flume er en distribuert tjeneste for effektiv innsamling, aggregering og flytting av store mengder logg-data til Hadoop for videre analyse.
  - Flume har en enkel og fleksibel arkitektur basert på «streaming data flows».
  - Flume bruker en enkel og utvidbar datamodell som er tilrettelagt for online analytiske applikasjoner
  - Flume-agenter settes opp gjennom redigering av en enkelt konfig-fil.



# Apache Sqoop







## Apache Sqoop



- Sqoop er et verktøy for å importere data fra relasjonelle databaser / datavarehus til Hadoop og evt. eksportere data fra Hadoop til relasjonelle databaser / datavarehus.
- Sqoop kan migrere data til/fra HDFS (csv, Avro osv), Hive og Hbase/Accumulo og til/fra de fleste kjente RDBMSer.
- Eks: Importer fra Oracle til Avro-fil (med 6 map-prossesser): sqoop import –connect "jdbc:oracle:thin:@(description= (address= (protocol=tcp)(host=orahost)(port=1521)) (connect\_data= (service\_name=myser))) --username SCOTT --password tiger --table MYTABLE -m 6 --as-avrodatafile --warehouse-dir /staging/sqoop/MYTABLE



#### **Apache Crunch**

- Apache Crunch er et bibliotek som kan kjøres på toppen av Hadoop MapReduce og Apache Spark
- Crunch er et enkelt Java API for oppgaver som sammenslåing og aggregering av data – oppgaver som er kronglete gjøre i ren MapReduce.
- Crunch er spesielt nyttig når man skal prosessere data som ikke passer naturlig inn i den relasjonelle modellen – som f.eks. tidsserier, serialiserte objektformater som protocol buffers eller Avro-records, eller HBase -rader og koloner.
- For Scala-brukere finnes Scrunch API, som er bygget på toppen av Java APIer og inkluderer en REPL (read-evalprint loop) for å lage MapReduce pipelines.



## Strømming av Big Data

- Parallell-prosessering av datastrømmer øker i bruk med økende datamengder og behov for sanntids-analyser.
- Noen mye brukte teknologier for dette er:
  - Apache Kafka
  - Apache Flume
  - Apache Storm
  - Apache Flink
  - Apache Spark Streaming



## Apache Kafka

# 8 kafka



### Apache Kafka

- «Apache Kafka is publish-subscribe messaging rethought as a distributed commit log»
- Kafka gir funksjonalitet som et «messaging system», men har et unikt design..



- Kafka håndterer strømmer med beskjeder i kategorier som kalles emner (topics).
- Prossesser som publiserer beskjeder til et Kafka-emne kalles produsenter.
- Prossesser som abonnerer på emner og prossesserer strømmer med publiserte beskjeder kalles konsumenter
- Kafka kjøres som en klynge bestående av en eller flere servere som alle kalles meklere.
- Kommunikasjon mellom servere og klienter gjøres via en språkagnostisk TCP-protokoll



#### Apache Kafka

#### Kafkas økosystem:

- Prossessering av datastrømmer
  - Storm A stream-processing framework.
  - Samza A YARN-based stream processing framework.
  - Storm Spout Konsumerer beskjeder fra Kafka og emitter dem som Storm tuples
  - Kafka-Storm Kafka 0.8, Storm 0.9, Avro-integrasjon
  - SparkStreaming Kafka reciever supports Kafka 0.8 and above
- Integrasjon med Hadoop
  - Camus LinkedIn's Kafka=>HDFS pipeline. «This one is used for all data at LinkedIn, and works great».
  - Kafka Hadoop Loader En annerledes implementering av Hadoops loadingfunksjonalitet .
  - Flume Inneholder Kafka Source (konsument) og Kafka Sink (produsent)









- Apache Storm er et open source «distributert sanntidsprossesseringssystem», opprinnelig utviklet av Twitter.
- Storm gjør det enkelt å prossessere uendelige strømmer med data, og gjør for sanntids-prossessering det Hadoop gjør for batch prossessering.
  - «Storm is simple, can be used with any programming language, and is a lot of fun to use!»
- Storm har mange bruksområder: realtime analytics, online machine learning, continuous computation, distributed RPC, ETL etc..
- Storm er rask: En benchmark har klokket den til over en million tuples prossessert per sekund per node.
- Storm er skalerbar, fault-tolerant, garanterer at data blir prossessert og er lett å sette opp og drifte.

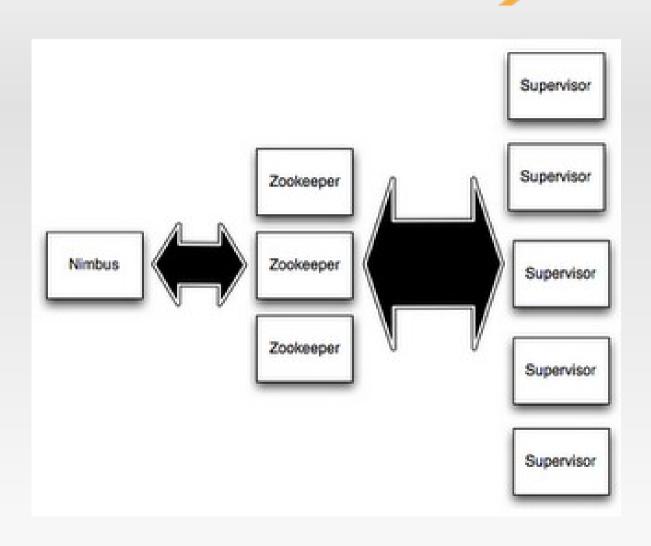




- Istedenfor MapReduce-jobber kjører man "topologier" med Storm. En «topologi» skiller seg fra en «jobb» bl.a. ved at MapReduce-jobber avsluttes, mens topologi-prossesser fortsetter å sende beskjeder til evig tid ( ... or until you kill it).
- En Storm-klynge har to typer noder: master og worker. Master noder kjørere en daemon som kalles "Nimbus" som ligner på en Hadoop "JobTracker". Nimbus distribuerer kode rundt i klyngen, tildeler maskiner jobber og overvåker systemet.
- Hver worker kjører en daemon kalt "Supervisor". Supervisor lytter etter arbeid tildelt maskinen og starter og stopper prossesser basert på det Nimbus har tildelt den.









#### **Apache Flink**



- Apache Flink er et rammeverk for prosessering av datastrømmer
- Kjernen i Apache Flink er en distribuert «streaming dataflow engine» skrevet i Java og Scala.
- Flink kjører valgte dataflow-programmer med parallell-prosessering av data i en «pipeline».



## Apache Flink

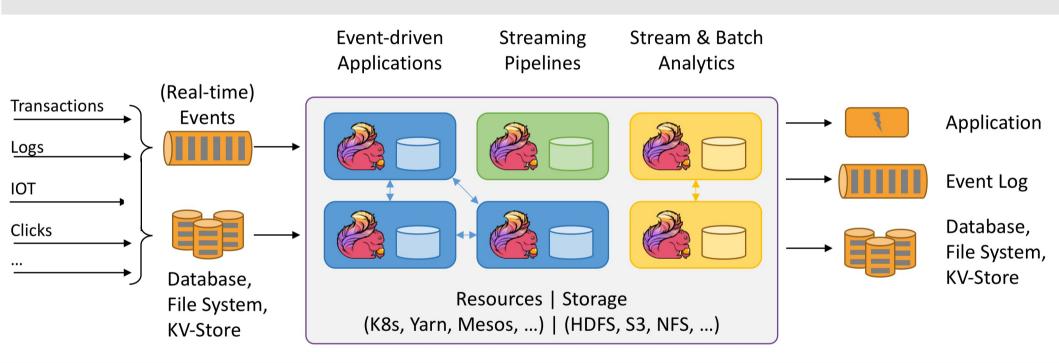


- Flink kan kjøres på Hadoop-klynger med YARN, på Apache Mesos eller Kubernetes, men også som en stand-alone klynge på egne servere.
- Flink er konfigurert for høy tilgjengelighet, uten noen «single point of failure».
- Flink har skalert til flere tusen kjerner og terabytes med data, med høy gjennomstrømningshastighet og lavt etterslep, og brukes av noen av verdens mest krevende strømme-prosesserings-applikasjoner. (eBay, AWS, Huawei, Uber osv)



## Apache Flink







## Maskinlæring

- Maskinlæring i distribuerte klynger er raskt og kan håndtere store datamengder.
- Teknologier for Big Data Maskinlæring inkluderer bla.:
  - Apache Mahout
  - Apache Spark ML
  - Apache HiveMall



#### **Apache Mahout**





#### **Apache Mahout**



- Mahout er en samling maskinlærings-algoritmer for Hadoop.
- Mahout har primært tre hovedområder:
  - Collaborative filtering
    - Anbefalelses-motorer: Andre som så denne videoen og også så andre vidoer du liker så også disse videoene
  - Clustering
    - Gruppering basert på felles kriteria: Videoer som ligner på denne..., Google news groups etc
  - Classification
    - Spam / ikke-spam basert på kriteria



## **Apache Hivemall**





## **Apache Hivemall**



- Apache Hivemall er hovedsakelig designet for å kjøre maskinlærings-algoritmer på Apache Hive, men støtter også Apache Pig og Apache Spark.
- Dermed kan Hivemall betraktes som et kryssplattforms-bibliotek for maskinlæring;
  - Prediksjonsmodeller bygget fra en batch-spørring mot Apache Hive kan bruker på Apache Spark/Pig, og omvendt kan modeller bygget med Apache Spark brukes fra Apache Hive/Pig.

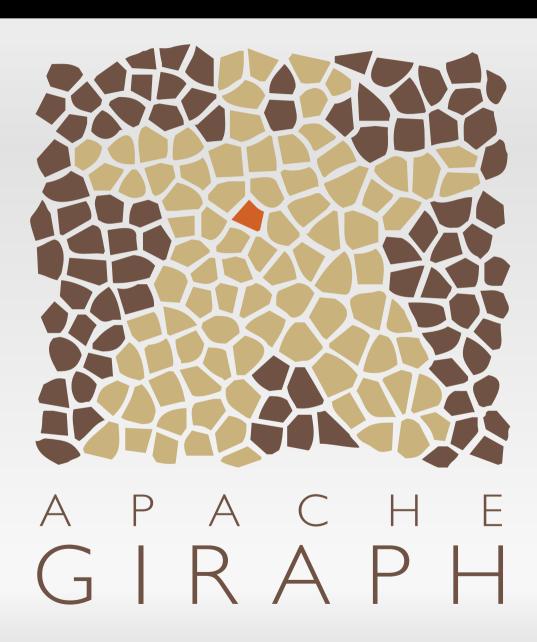


## Prosessering av data

- Det finnes en rekke Big Data-teknologier for prosessering av data i batch eller som stream.
  - Prosessering foregår i distribuerte server-klynger
  - Prosessering foregår parallellt
  - Har APIer mot mange av de mest brukte programmerings-språkene (Java, Scala, Python, SQL, etc)

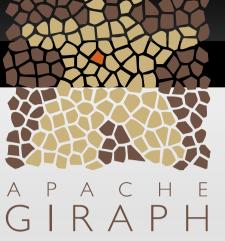


## Apache Giraph





### Apache Giraph



- Apache Giraph er et system for iterativ graph-prossessering bygget for høy skalerbarhet.
- For eksempel brukes den av Facebook for å analysere den sosiale grafen som produseres av brukere og deres forbindelser.
- Giraph ble laget som en open-source motpart til Googles «Pregel», som de beskrev i et paper i 2010. Både Pregel og Giraph er inspirert av «Bulk Synchronous Parallel»-modellen for distributert prosessering introdusert av Leslie Valiant.
- Giraph inkluderer bl.a.:
  - Master computation, Sharded aggregators, Edge-oriented input, Out-of-core computation, m.m.



# Apache Tez





#### **Apache Tez**



- Apache Tez er et prosjekt med formål å bygge et applikasjons-rammeverk som bruker en kompleks directed-acyclic-graph av oppgaver for å prossessere data. Tez bygges på toppen av Apache Hadoop YARN
- Apache Tez lar Hive og Pig kjøres via en kompleks DAG med oppgaver. Dermed kan en enkelt Tez-jobb brukes til å prossessere data som tidligere krevde flere MapReduce-jobber.
- = Betraktelig raskere enn MapReduce



## Stinger-initiativet

#### The Stinger Initiative: Making Apache Hive 100x Faster

#### **Apache Hive**

#### **Base Optimizations**

Generate simplified DAGs In-memory Hash Joins

#### **Deep Analytics**

SQL Compatible Types
SQL Compatible Windowing
More SQL Subqueries

Apache Hadoop

#### **Vector Query Engine**

Optimized for modern processor architectures

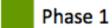
#### **Query Planner**

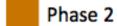
Intelligent Cost-Based Optimizer

#### **Hive Query Server**

Pre-warmed Containers Low-latency dispatch

#### Hive/Stinger Phases





Phase 3

#### **Buffer Caching**

Cache accessed data Optimized for vector engine

#### **ORCFile**

Column Store
High Compression
Predicate / Filter Pushdowns

#### Tez

Express data processing tasks more simply Eliminate disk writes

#### YARN

Next-gen Hadoop data processing framework



#### Apache Beam



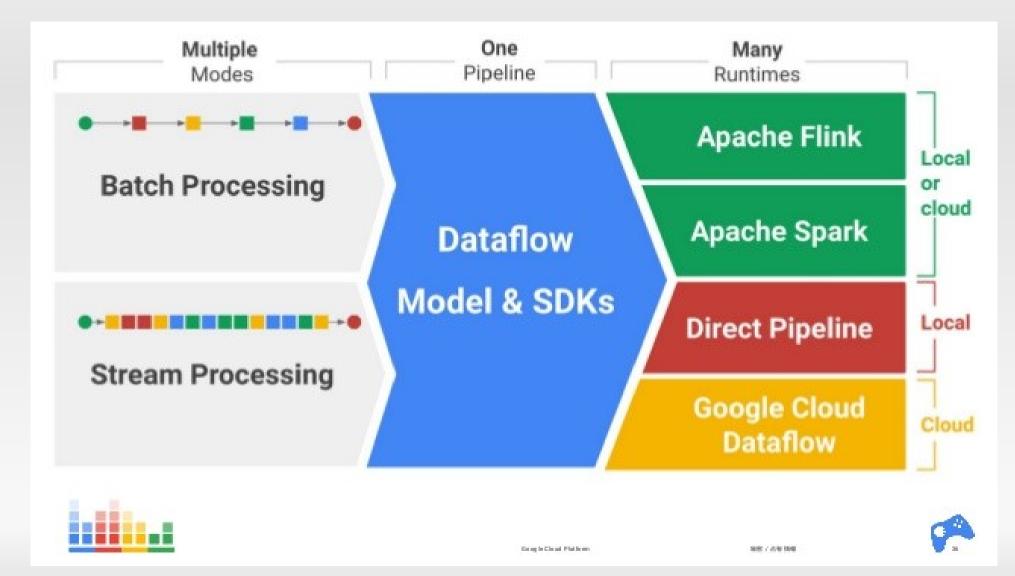
#### «An advanced unified programming model»

- Batch og datastrømmer kan prosesseres på en mengde «execution engines»:
- DirectRunner: kjøres lokalt på en maskin for utvikling og testing.
- ApexRunner: Kjører på Apache Apex.
- FlinkRunner: Kjører på Apache Flink.
- SparkRunner: Kjører på Apache Spark.
- DataflowRunner: Kjører på Google Cloud Dataflow
- GearpumpRunner: Kjører på Apache Gearpump (incubating).
- SamzaRunner: Kjører på Apache Samza.
- NemoRunner: Kjører på Apache Nemo



## Apache Beam







## Apache Spark





#### Apache Spark



Apache Spark er et helhetlig system for parallell-prossesering som kan kjøres sammen med eller istedenfor Hadoop.

- Spark kan kjøre programmer opptil 100x raskere enn Hadoop MapReduce «in memory» eller 10x raskere fra disk.
- Spark har en avansert «DAG execution engine» som støtter «cyclic data flow» og «in-memory computing».
- Spark kommer med en ny versjon hver 3. måned





## **Apache Spark**



- Spark gjør det enkelt å bygge parallelle applikasjoner og kan brukes interaktivt fra Scala-, R- eller Python-skall.
   Applikasjoner kan bygges i Java, Scala, Python eller R.
- Spark innheolder en rekke høy-nivå-verktøy, inkludert Spark SQL, Spark MLlib for maskinlæring, GraphX og Spark Streaming. Disse rammeverkene kan integreres sømløst innen samme applikasjon.
- Spark kan kjøres på Hadoop 2 klynger via YARN og kan lese alle eksisterende Hadoop datakilder, så som Avro, Parquet, CSV, HBase, MongoDB eller Cassandra.
- Med en eksisterende Hadoop 2 klynge, kan Spark kjøres uten noen spesiell installasjon. Eller man kan kjøre Spark standalone, evt. på EC2 eller Mesos.



### Cloudera HUE





### **Hadoop User Experience**

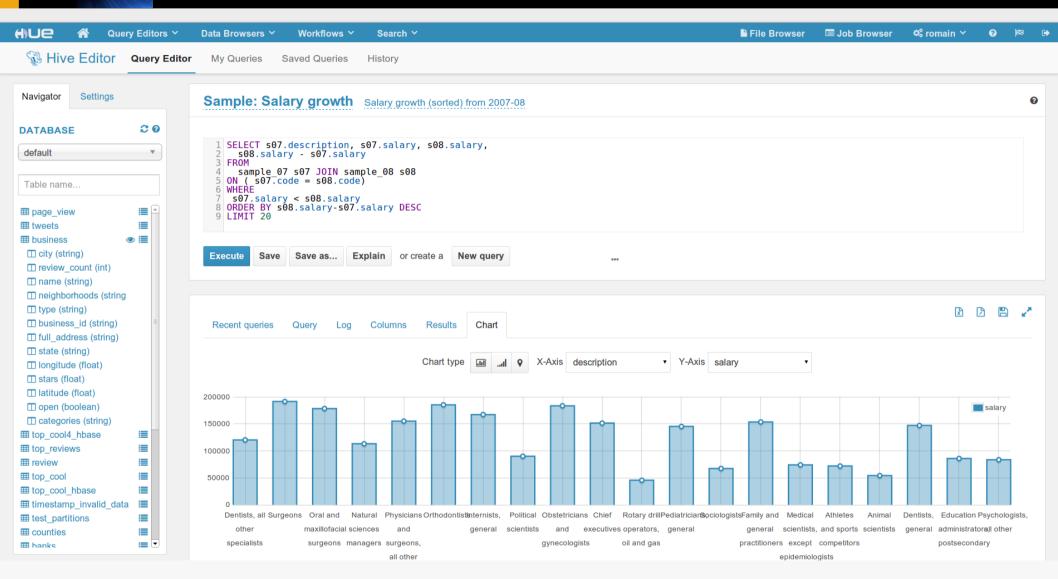


- Hue er et Webgrensesnitt for analyse av data med Apache Hadoop, utviklet av Cloudera for Hadoop-distroen CDH og «open-sourcet» slik at den kan brukes med alle Hadoop-distroer.
- Sjekk http://demo.gethue.com for en interaktiv demo.



### **Hadoop User Experience**







### Sikkerhet og «Governance»

- In-house Big Data-klynger er ofte laget med hovedsakelig skall-sikring som beskyttelse mot omverden.
- Nyere versjoner av bla. Hadoop-klynger inneholder også gode mekanismer for autentisering og autorisering av brukere, f.eks. gjennom Kerberos, Ldap/Active Directory osv.



## Apache Ranger

### Apache Ranger

- Apache Ranger er et rammeverk for data sikkerhet, som bl.a. tilbyr:
  - Sentralisert sikkerhets-administrasjon via et UI eller ved å bruke REST APIs.
  - Administrering av finmaskede tilgangsrettigheter
  - Standardiserte metoder for tilgangsrettigheter for alle Hadoop-komponentene.
  - Støtte for ulike rettighets-metoder: Rollebasert, attributtbasert etc.
  - Sentralisert auditing av brukertilgang for alle Hadoops komponenter



### **Apache Knox**



- Knox tilbyr skallsikring av Hadoop-klynger- en «REST API Gateway for the Apache Hadoop Ecosystem»
  - Authentication (LDAP and Active Directory Authentication Provider)
  - Federation/SSO (HTTP Header Based Identity Federation)
  - Authorization (Service Level Authorization)
- Knox sikrer også Hadoops web Ul'er





## Apache Ambari

http://incubator.apache.org/ambari

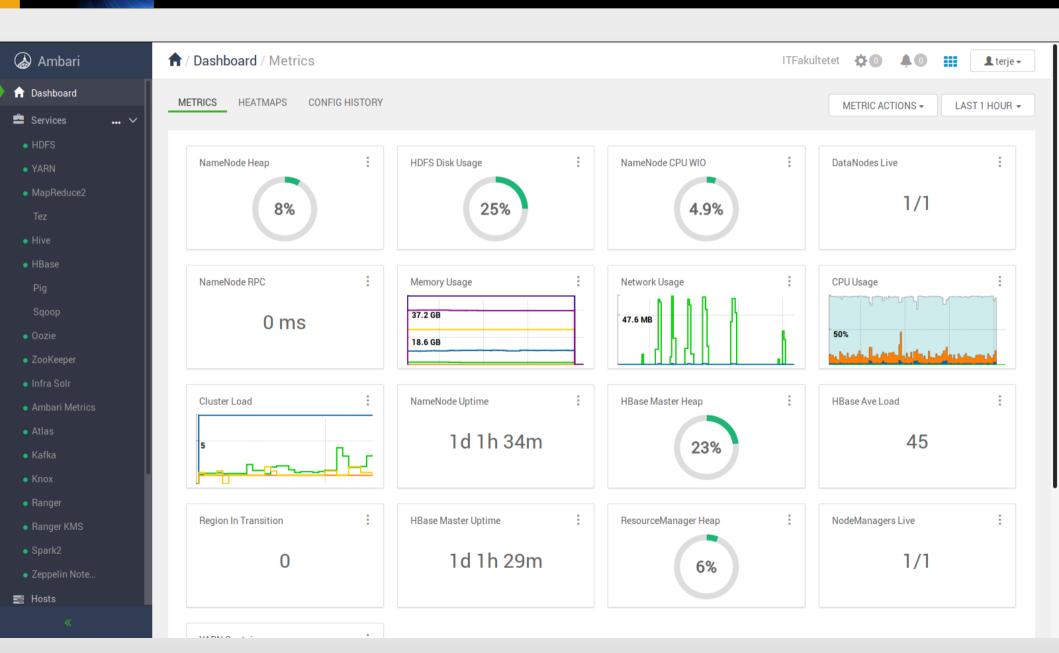




- Ambari er et web-basert verktøy laget av Hortonworks for å provisjonere, styre og overvåke Apache Hadoop-klynger, med støtte for bl.a. HDFS, MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig og Sqoop.
- Ambari inneholder også et dashbord for overvåking av klyngens helsetilstand – med bl.a. heatmaps og muligheter for å vise MapReduce-, Pig- og Hive-applikasjoner visuelt sammen med deres ytelses-mål på en brukervennlig måte.
- Ambari lar System Administratorer:
  - Provisjonere en Hadoop-klynge med trinnvise guider for å installere ulike Hadoop-tjenester på alle verter i klyngen.
  - Konfigurere, starte, stoppe, rekonfigurere og restarte Hadoop-tjenester for klyngen fra ett webgrensesnitt.
  - Overvåke status og helsetilstanden til Hadoop-klynger via et intuitivt dashbord – integrert med Ganglia for sanntids datainnsamling og Nagios til varsling når f.eks. en node går ned eller det er lite lagringsplass igjen etc.

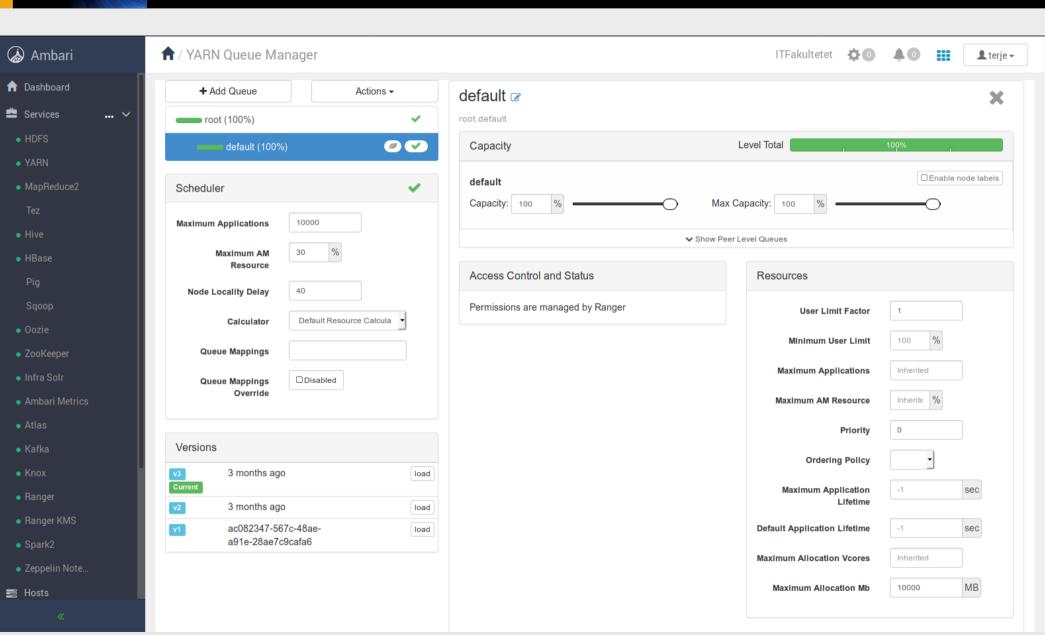














### **Hadoop Distroer**















# **Apache Bigtop**





## **Apache Bigtop**



Apache Bigtop har samlet kompatible versjoner av en rekke Hadoop-komponenter:

- ✓ Apache Crunch
- Apache Flume
- Apache Giraph
- Apache HBase
- Apache HCatalog
- Apache Hive
- Apache Mahout
- Apache Oozie

- Apache Pig
- Apache Solr
- Apache Sqoop
- Apache Whirr
- ✓ Apache Zookeeper
- Cloudera Hue
- ✓ LinkedIn DataFu

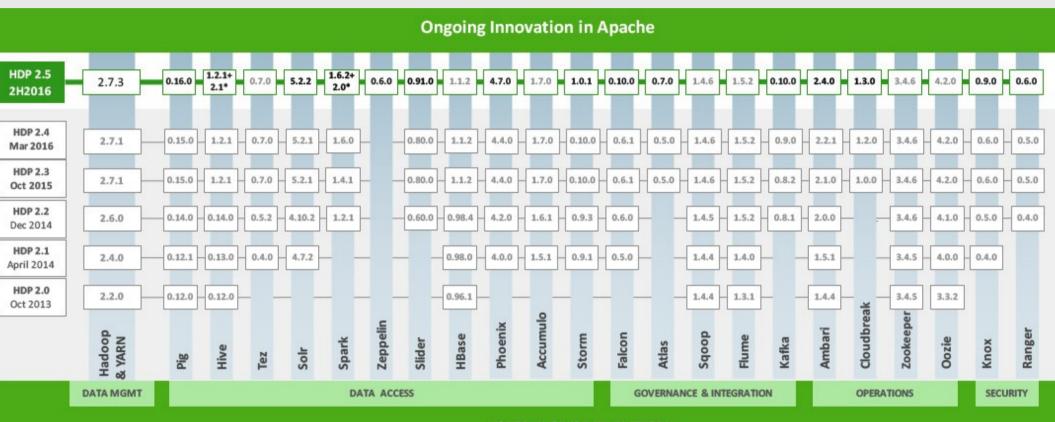


### Hortonworks





### **Hortonworks Data Platform**

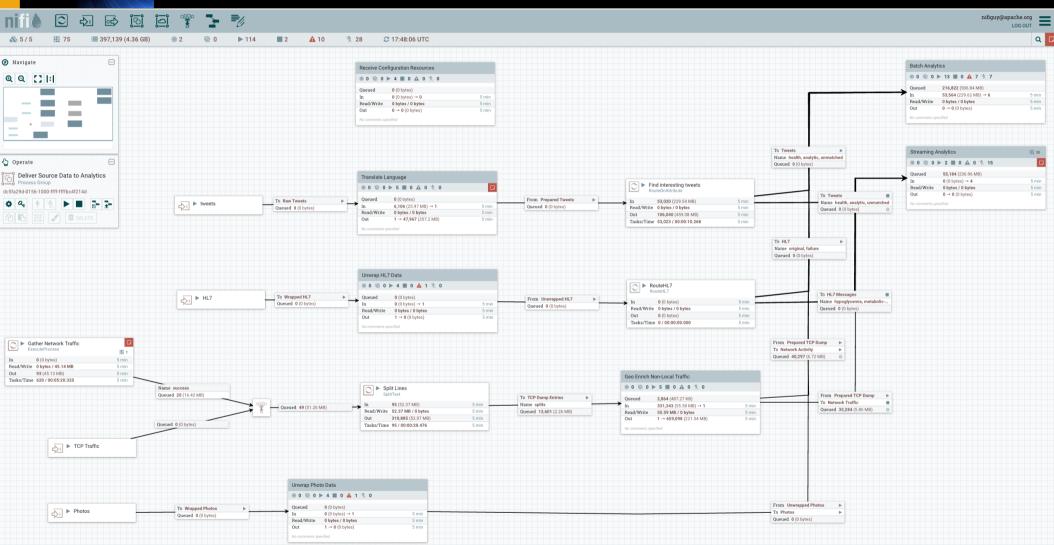


### HORTONWORKS DATA PLATFORM

\* Spark 1.6.2+ Spark 2.0 – HDP 2.5 support installation of both Spark 1.6.2 and Spark 2.0. Spark 2.0 is Technical Preview within HDP 2.5. Hive 1.2.1+ Hive 2.1 – Hive 2.1 is Technical Preview within HDP 2.5.



### **Hortonworks Data Flow**



NiFi 1.0.0 » Deliver Source Data to Analytics



### Cloudera





### Cloudera Distribution for Hadoop

### CDH

BATCH PROCESSING (MapReduce, Hive, Pig) ANALYTIC SQL (Impala) SEARCH ENGINE (Cloudera Search) MACHINE LEARNING (Spark, MapReduce, Mahout) STREAM PROCESSING (Spark) 3RD PARTY APPS (Partners)

WORKLOAD MANAGEMENT (YARN)

STORAGE FOR ANY TYPE OF DATA

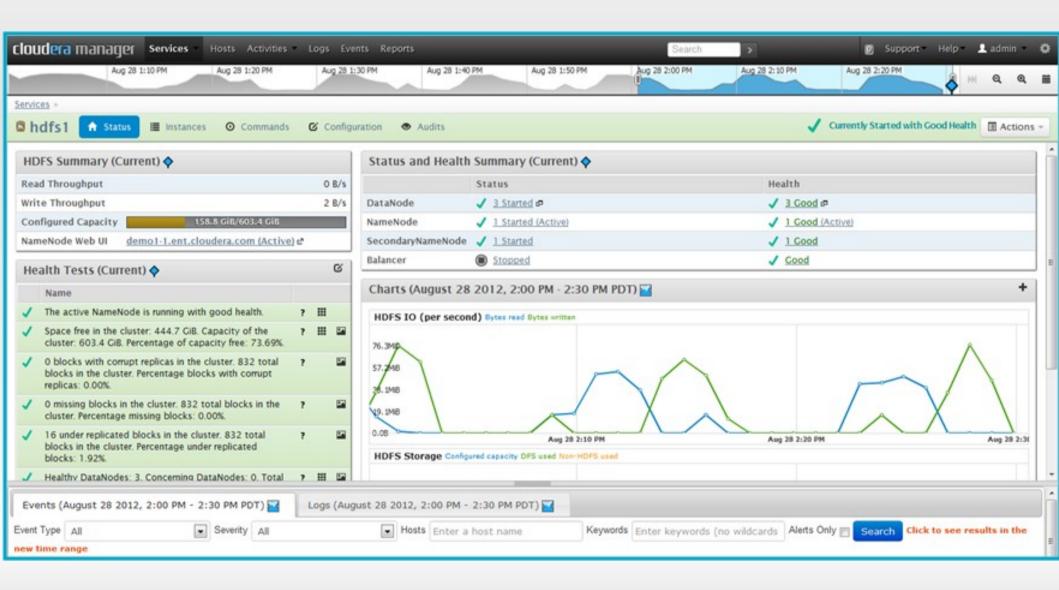
UNIFIED, ELASTIC, RESILIENT, SECURE (Sentry)

Filesystem (HDFS) Online NoSQL (HBase)

DATA INTEGRATION (Sqoop, Flume, NFS)



### Cloudera Manager



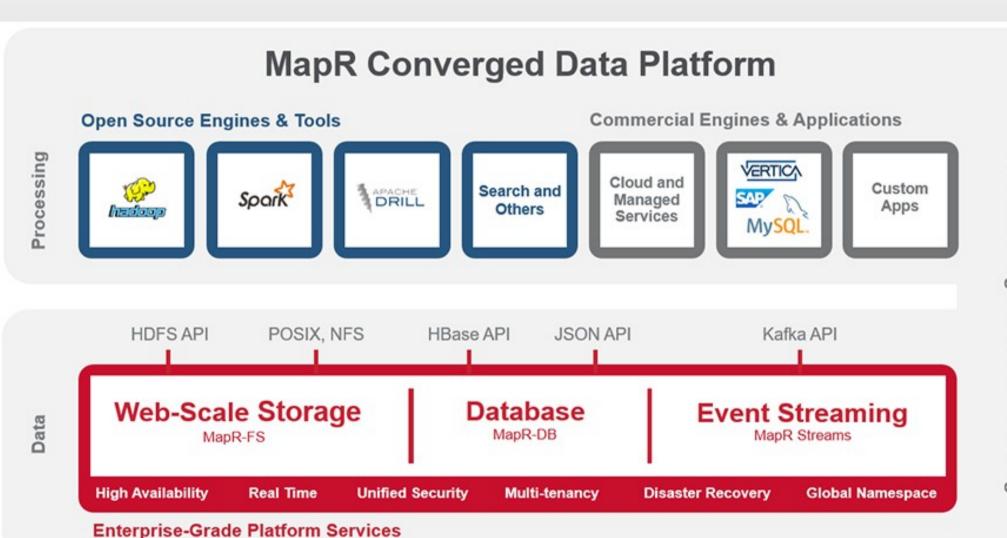


### MapR





### MapR Data Platform



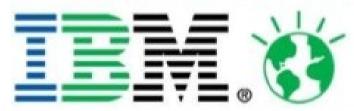
**Unified Management and Monitoring** 



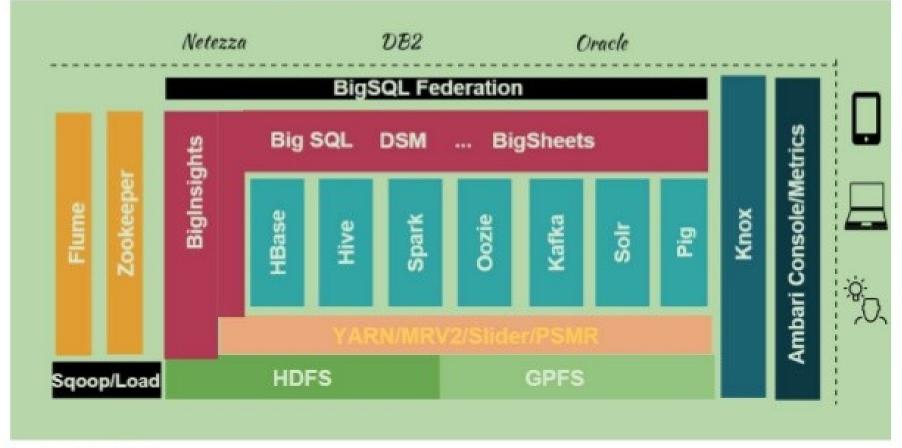
### IBM

### Architecture Overview - IBM Open Platform





dibinização





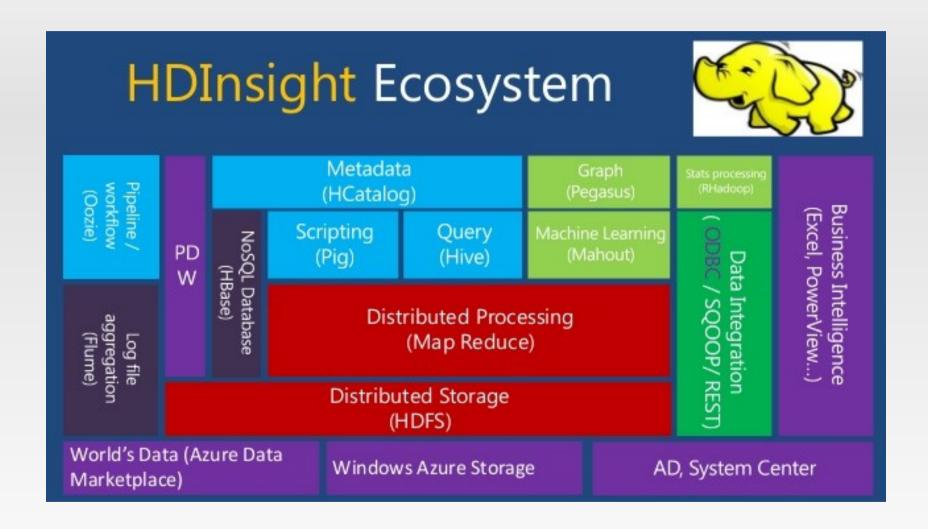
### **IBM**

- In 2015, IBM® released the IBM Open Platform (IOP) with Apache Hadoop and Spark.
- IOP is IBM's big data platform, built on 100% open source Apache ecosystem components as if you had downloaded components from Apache.org and built it yourself.



## Microsoft HDInsight







### De som ikke ble glemt...

- Det er en rekke teknologier som ikke er nevnt i denne enkle presentasjonen, f.eks.:
  - Cloudera Impala sql-basert database
  - Apache Slider «slide inn» din applikasjon på Yarn
  - Apache Drill drill down into stuff with sql
  - Apache Ranger sikkerhet, gjerne koblet med Apache Knox
  - Apache Flink ref Spark
  - Apache Phoenix SQL for HBase
  - Apache Samza Stream Processing Framework
  - Apache Lens og Apache Kylin OLAP basert på Hive
  - Apache Zeppelin Online Notebook for bl.a. Spark



### De som ikke ble glemt...

### ...forts...:

- Apache Trafodion SQL over HBase, inkludert støtte for transaksjoner, laget av HP
- Apache Hawq Distribuert SQL-løsning basert på Pivotals Greenplum/PostgreSQL
- Apache Atlas Data Governance og Metadata framework
- Falcon Feed management og data processing platform (Onboarding av data, Policybasert Livs-syklus Management)
- Apache Arrow Kolonnebasert in-memory lagring for kryssbruk mellom applikasjoner
- Apache Beam «...unified programming model that you can use to create a data processing pipeline».
- AsterixDB JSON-basert database for yarn/HDFS



## **Apache Software Foundation**

- Det dukker opp nye Hadoop-relaterte prosjekter annenhver uke (ca) – og de kan deles i to kategorier:
  - De som administreres av ASF
  - De som kommer til å bli det.
- Først undergår de ASF incubation, så løftes de til hovedprosjekter.
  - http://incubator.apache.org
  - http://projects.apache.org