Berkeley Data Analytics Stack (BDAS)

Ion Stoica
Databricks / UC Berkeley

What is Big Data used For?

Reports, e.g.,

» Track business processes, transactions

Diagnosis, e.g.,

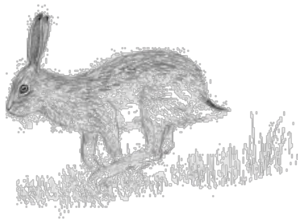
» Why is user engagement dropping?
» Why is the system slow?
» Detect spam, worms, viruses, DDoS attacks

Decisions, e.g.,

» Personalized medical treatment
» Decide what feature to add to a product
» Decide what ads to show

Data is only as useful as the decisions it enables

Data Processing Goals

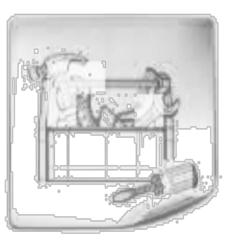


Low latency (interactive) queries on historical data: enable faster decisions

» E.g., identify why a site is slow and fix it

Low latency queries on live data (streaming): enable decisions on real-time data

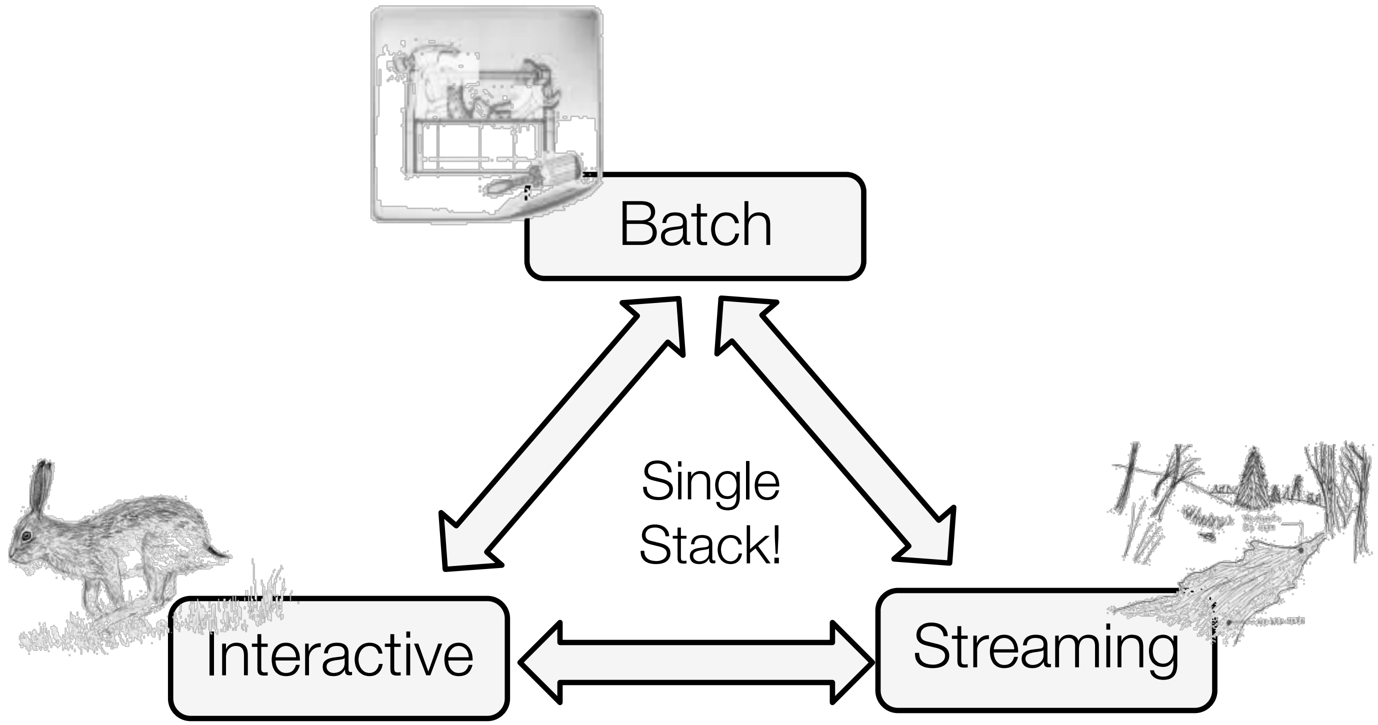
» E.g., detect & block worms in real-time (a worm may infect **1mil** hosts in **1.3sec**)



Sophisticated data processing: enable “better” decisions

» E.g., anomaly detection, trend analysis

Our Goal



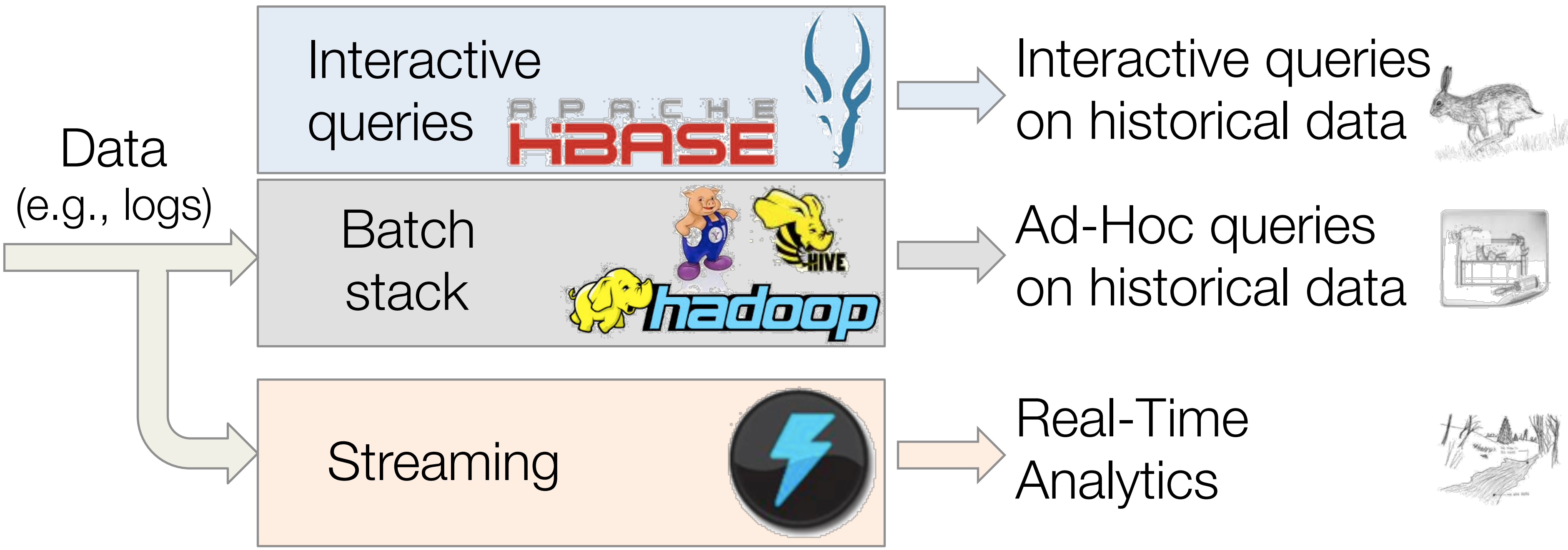
Support *batch*, *streaming*, and *interactive* computations…

… in a unified framework

Easy to develop *sophisticated* algorithms (e.g., graph, ML algos)

The Need For Unification (1/2)

Today's state-of-art analytics stack

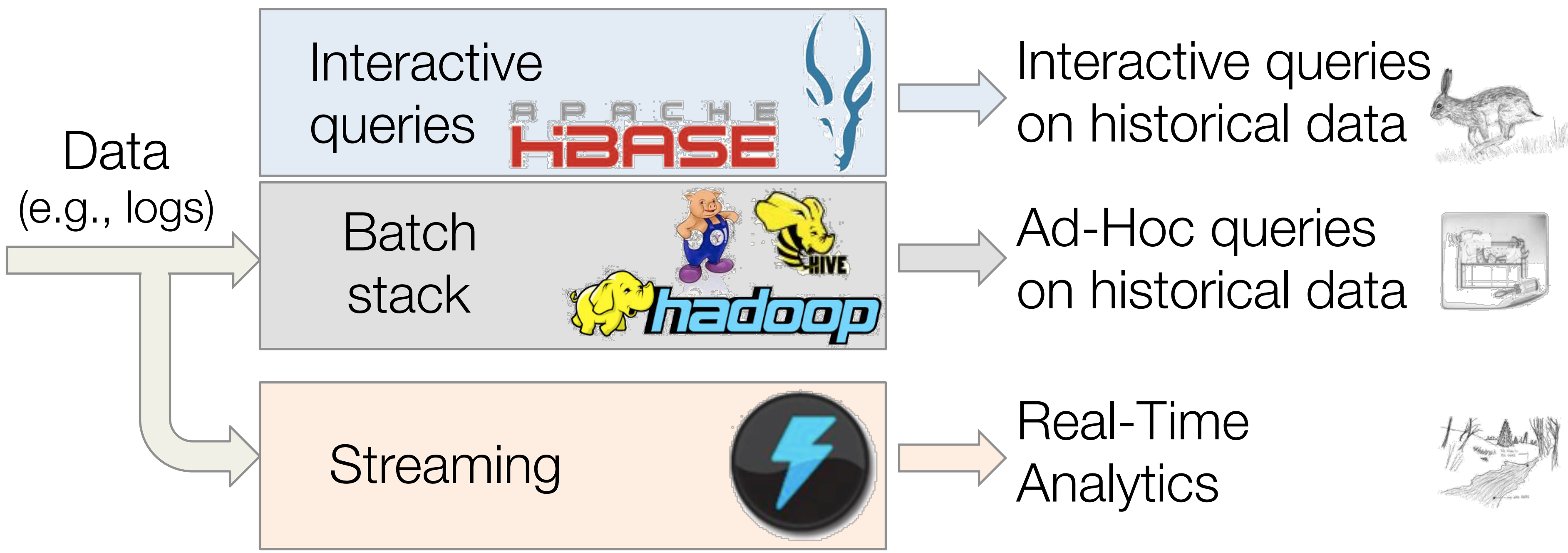


Challenge 1: need to maintain three stacks

- Expensive and complex
- Hard to compute consistent metrics across stacks

The Need For Unification (1/2)

Today's state-of-art analytics stack



Challenge 2: hard/slow to share date

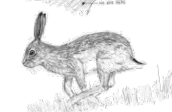
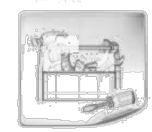
» E.g., cannot perform interactive queries on streamed data

The Need for Unification (2/2)

Make real-time decisions

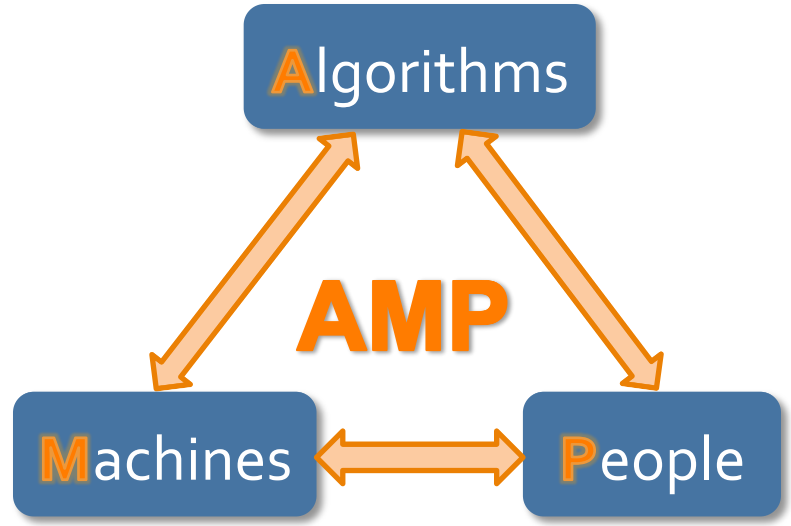
» Detect DDoS, fraud, etc

E.g.,: what's needed to detect a DDoS attack?

1. Detect attack pattern in real time → streaming 
2. Is traffic surge expected? → interactive queries 
3. Making queries fast → pre-computation (batch) 

And need to implement complex algos (e.g., ML)!

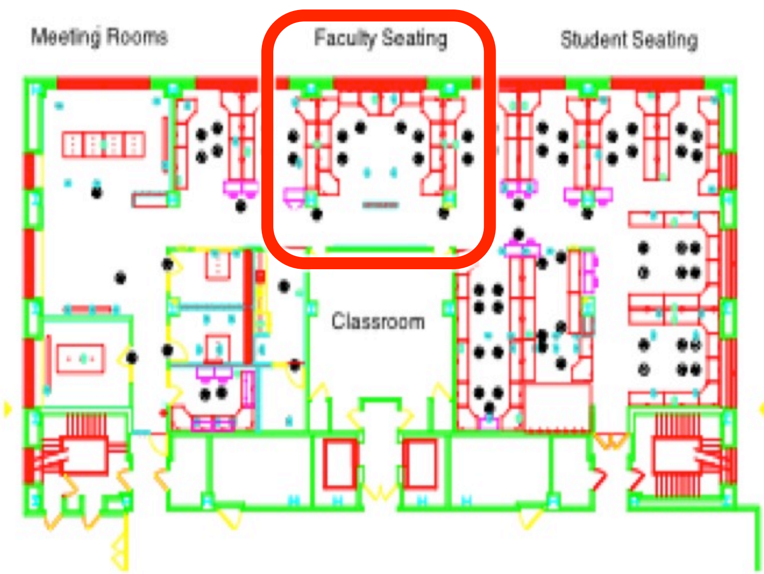
The Berkeley AMPLab



January 2011 – 2017

» 8 faculty
» > 40 students
» 3 software engineer team

Organized for collaboration



3 day retreats
(twice a year)

AMPCamp3
(August, 2013)

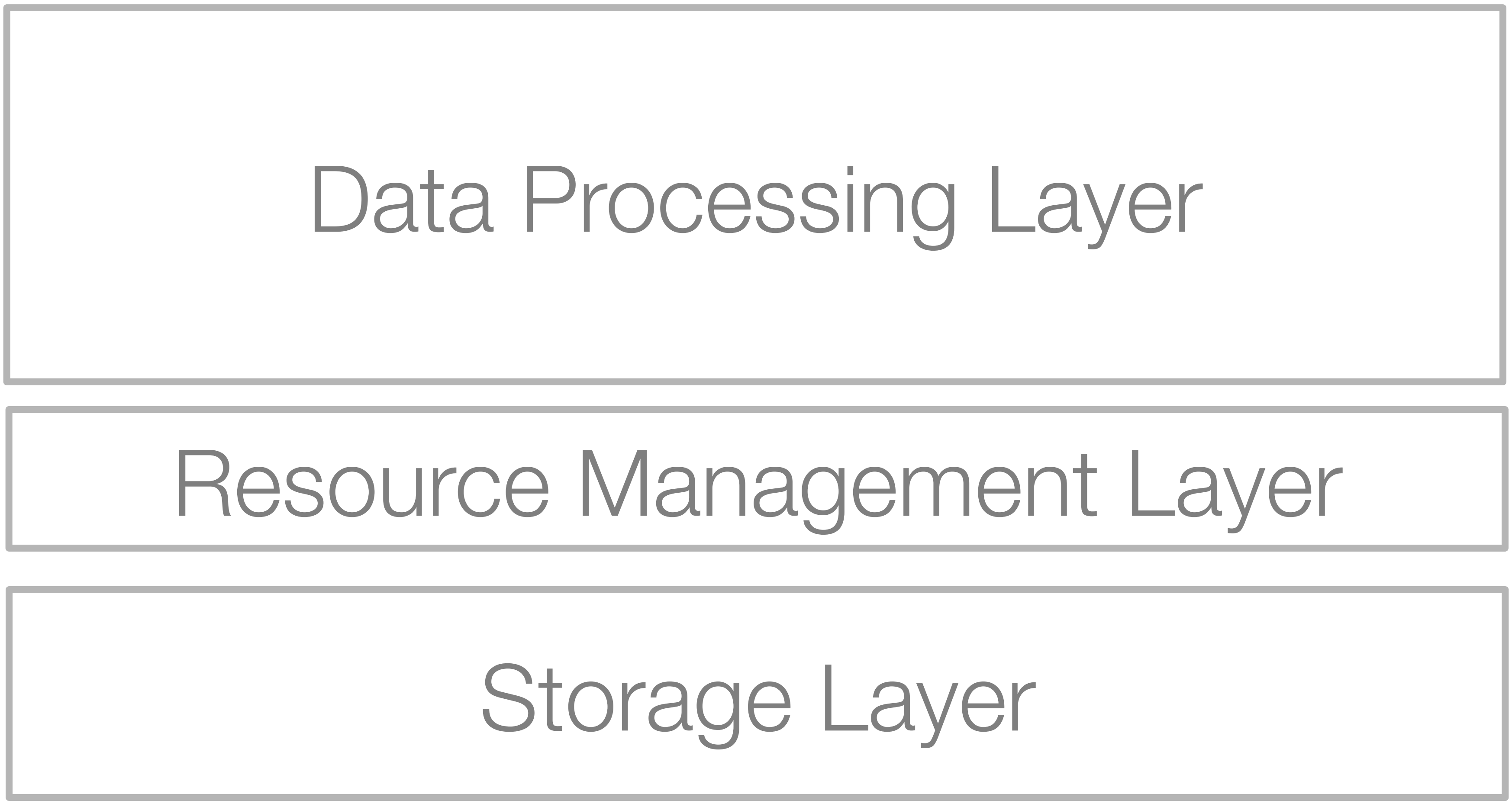
220 campers
(100+ companies)

The Berkeley AMPLab

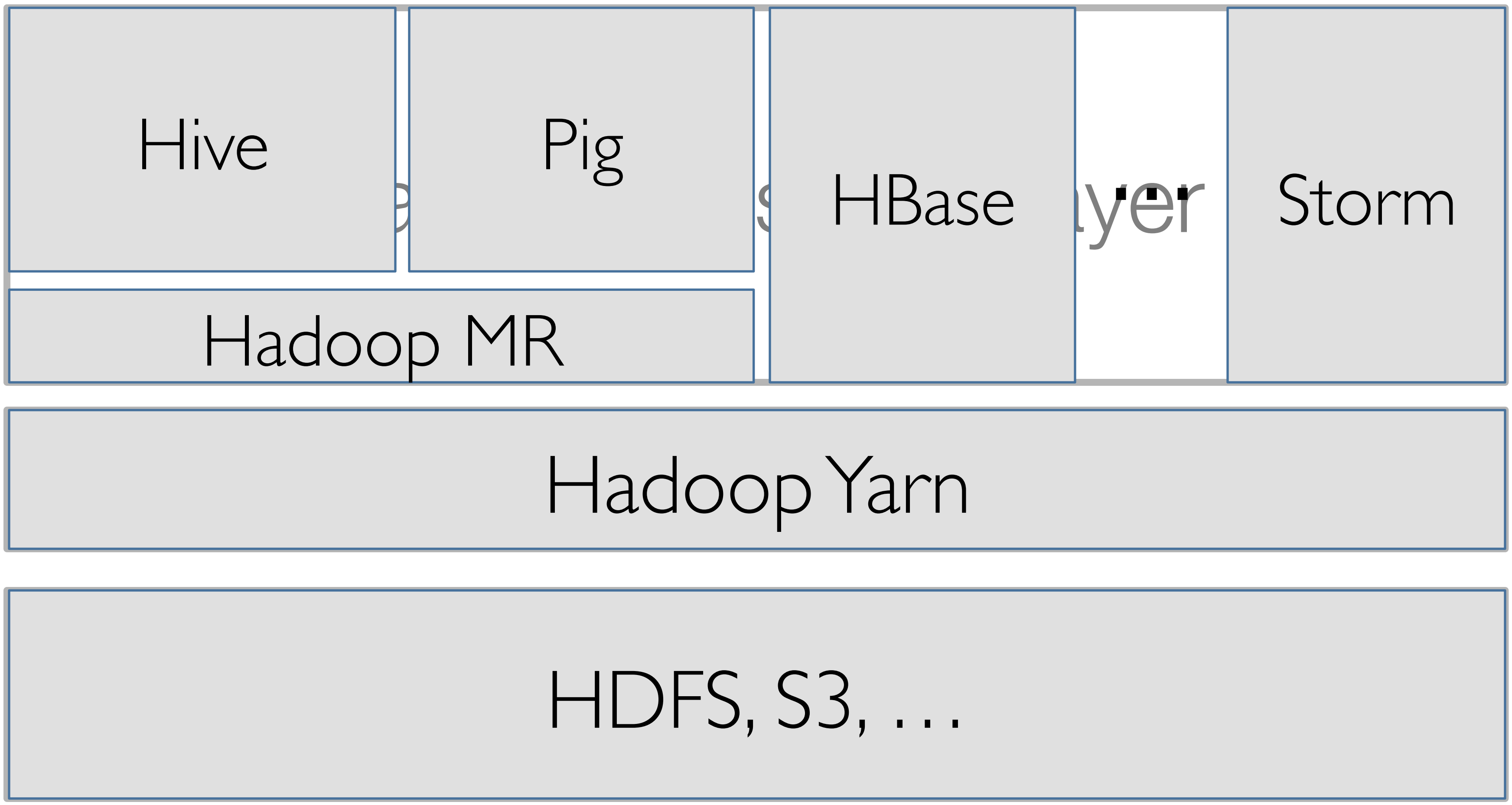
Governmental and industrial funding:

Goal: Next generation of open source data analytics stack for industry & academia: **Berkeley Data Analytics Stack (BDAS)**

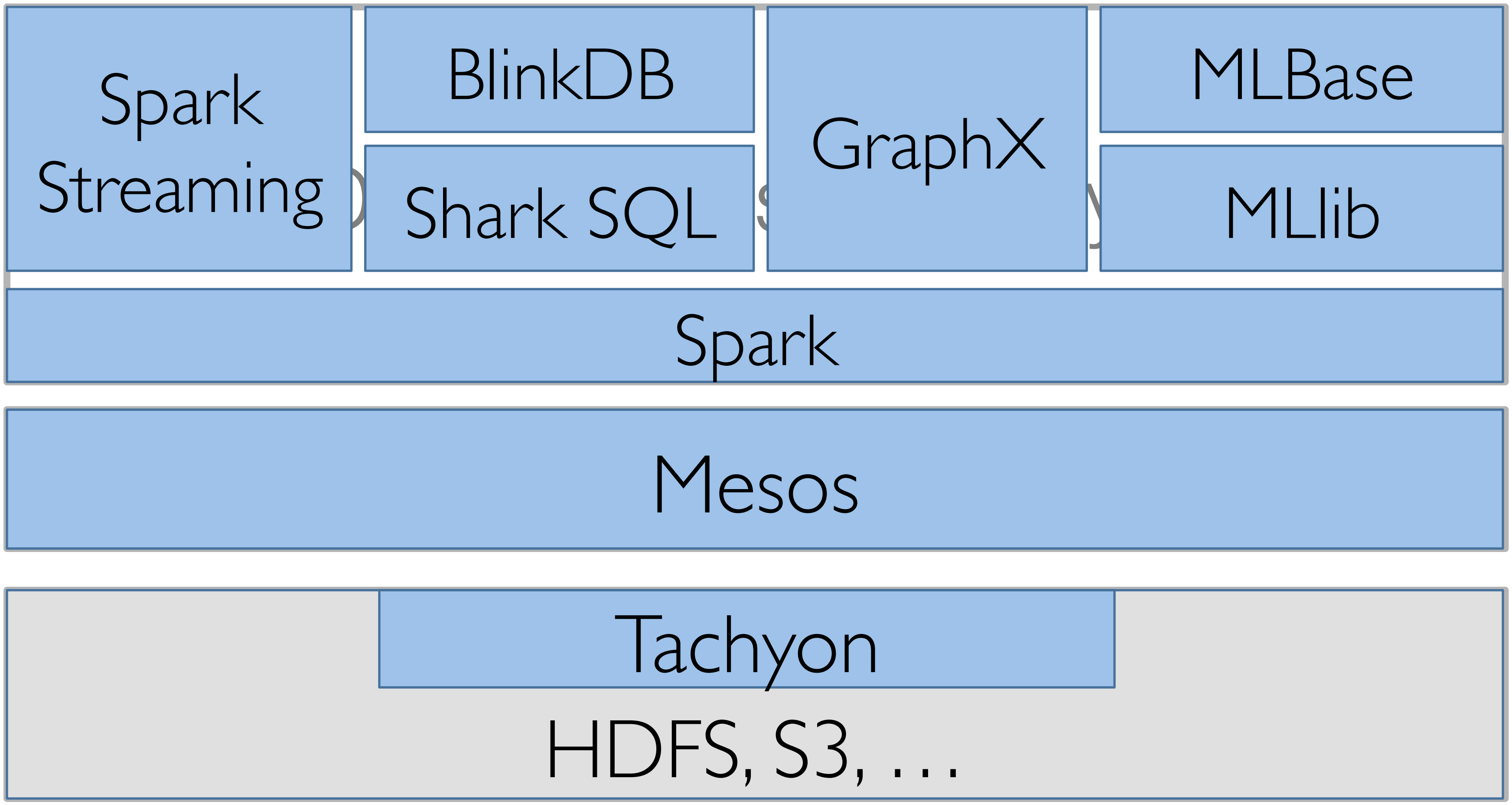
Data Processing Stack



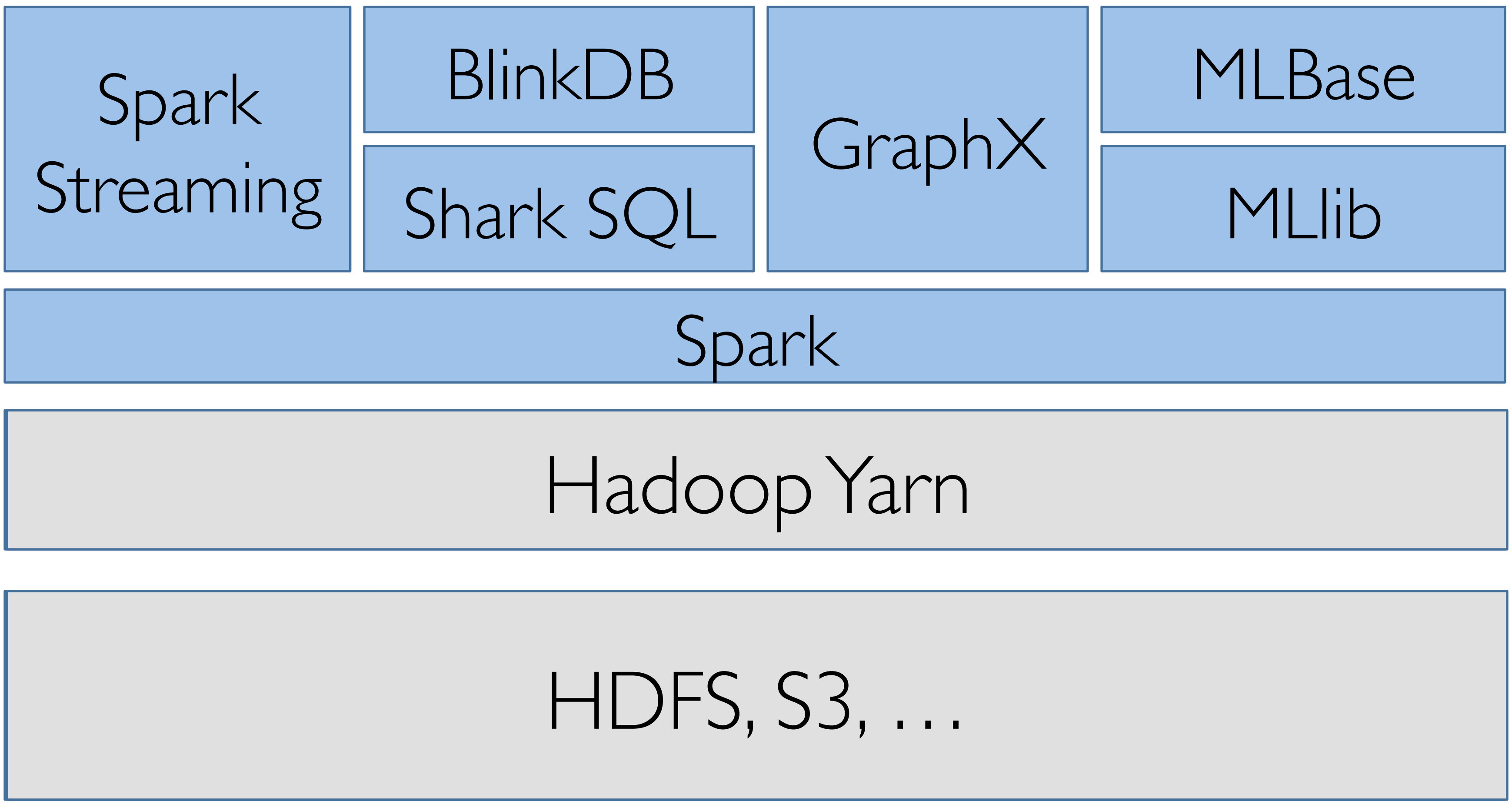
Hadoop Stack



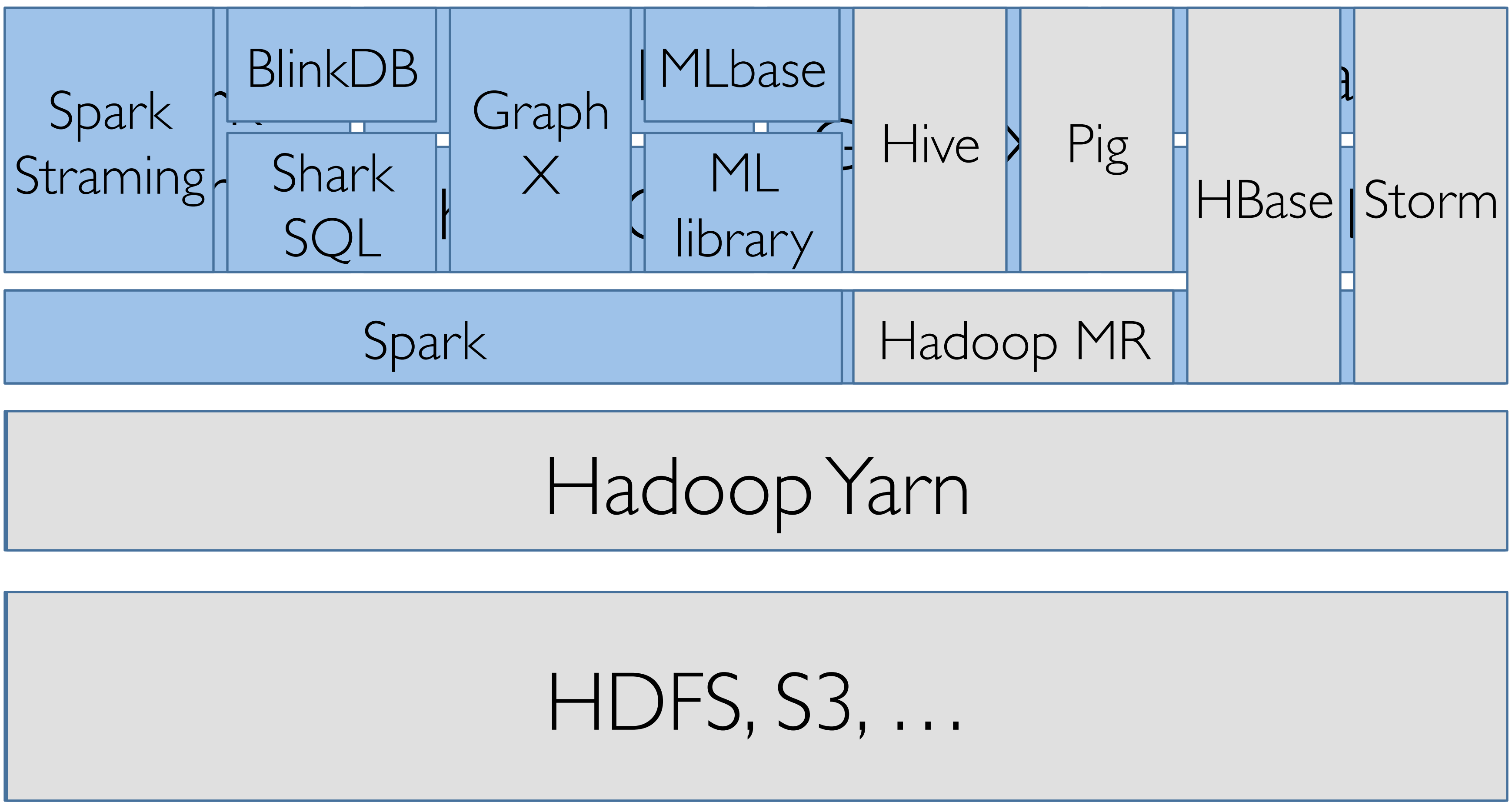
BDAS Stack


Spark Streaming
BlinkDB
Shark SQL
GraphX
MLBase
MLlib
Spark
Mesos
Tachyon
HDFS, S3, …

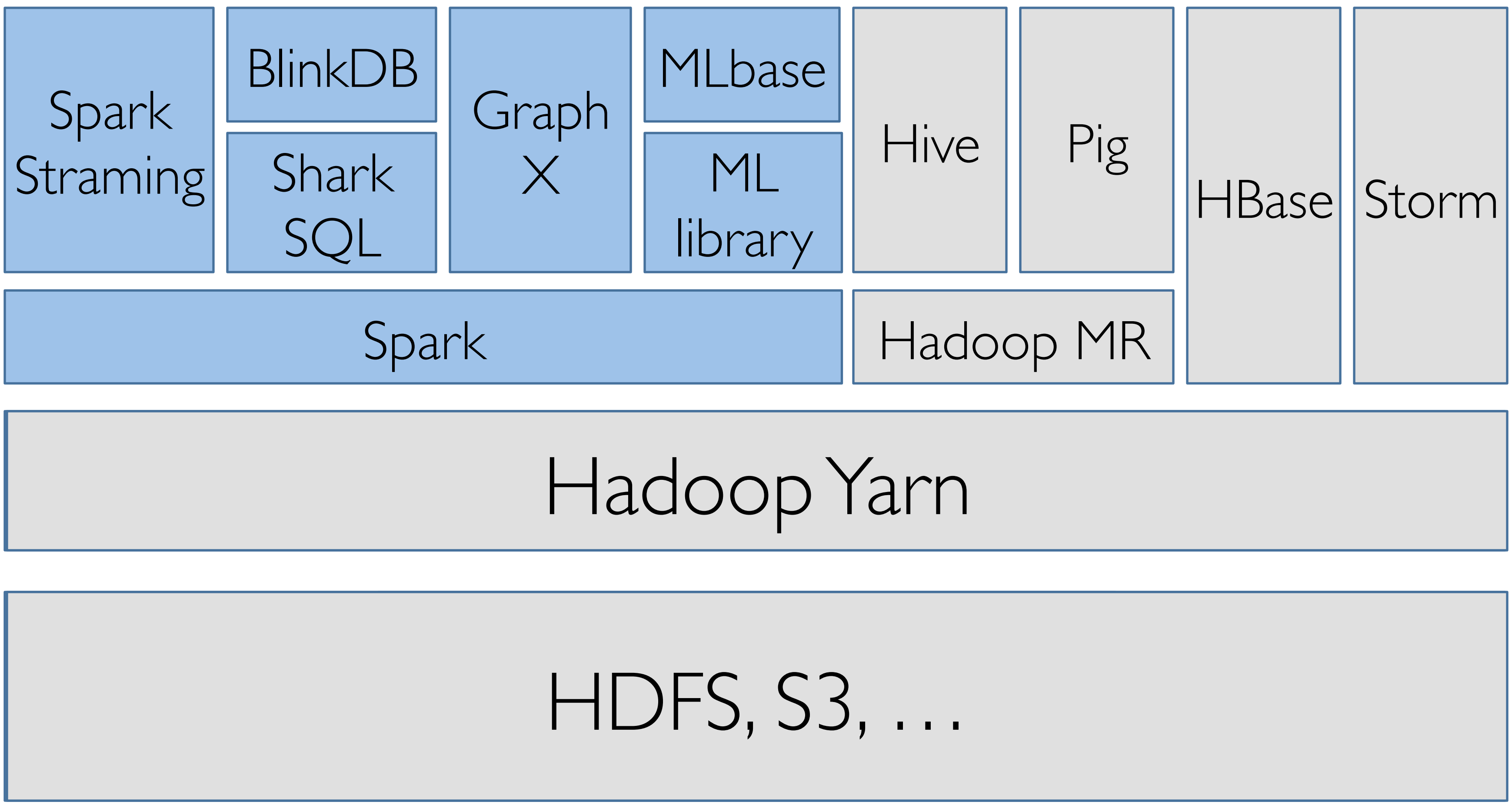
How do BDAS & Hadoop fit together?



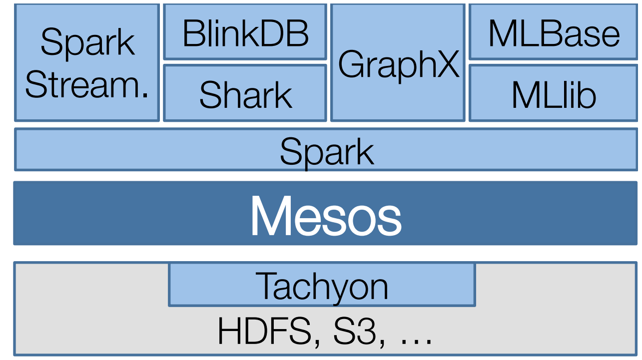
How do BDAS & Hadoop fit together?



How do BDAS & Hadoop fit together?



Apache Mesos



Enable multiple frameworks to share same cluster resources (e.g., Hadoop, Storm, Spark)

Twitter's large scale deployment

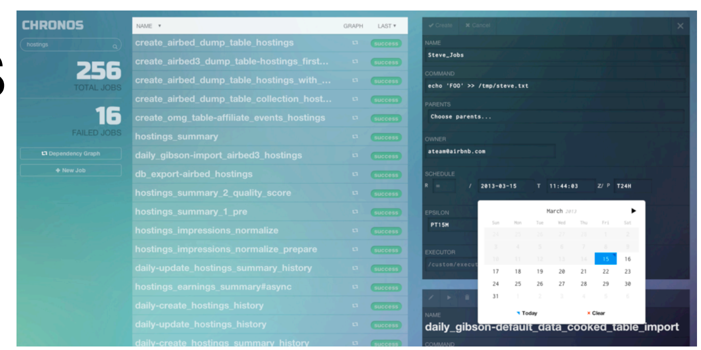
» 6,000+ servers,

» 500+ engineers running jobs on Mesos

Third party Mesos schedulers

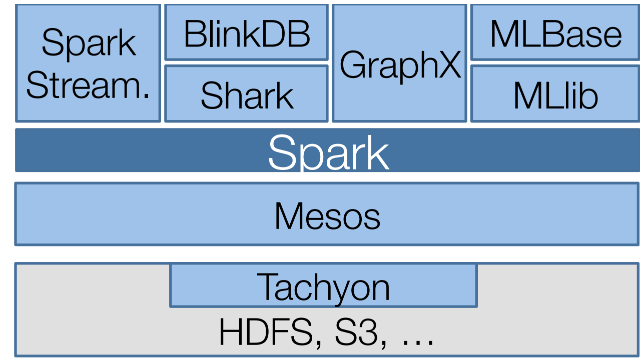
» AirBnB's Chronos

» Twitter's Aurora



Mesospehere: startup to commercialize Mesos

Apache Spark



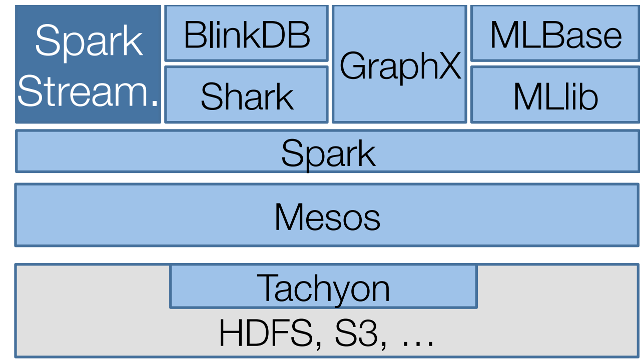
Distributed Execution Engine

- » Fault-tolerant, efficient in-memory storage
- » Powerful programming model and APIs (Scala, Python, Java)

Fast: up to 100x faster than Hadoop MR

Easy to use: 2-5x less code than Hadoop MR

General: support interactive & iterative apps



Spark Streaming

Large scale streaming engine

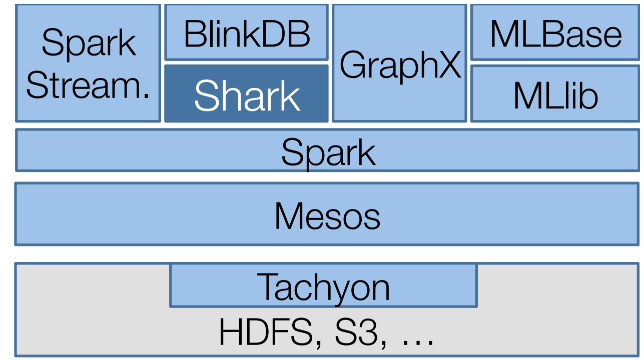
Implement streaming as a sequence of <1s jobs

» Fault tolerant
» Handle stragglers
» Ensure exactly one semantics

Integrated with Spark: unifies batch, interactive, and streaming computations

Alpha release (Spring, 2013), Beta release (Nov.)

Shark



Hive over Spark: full support for HQL and UDFs

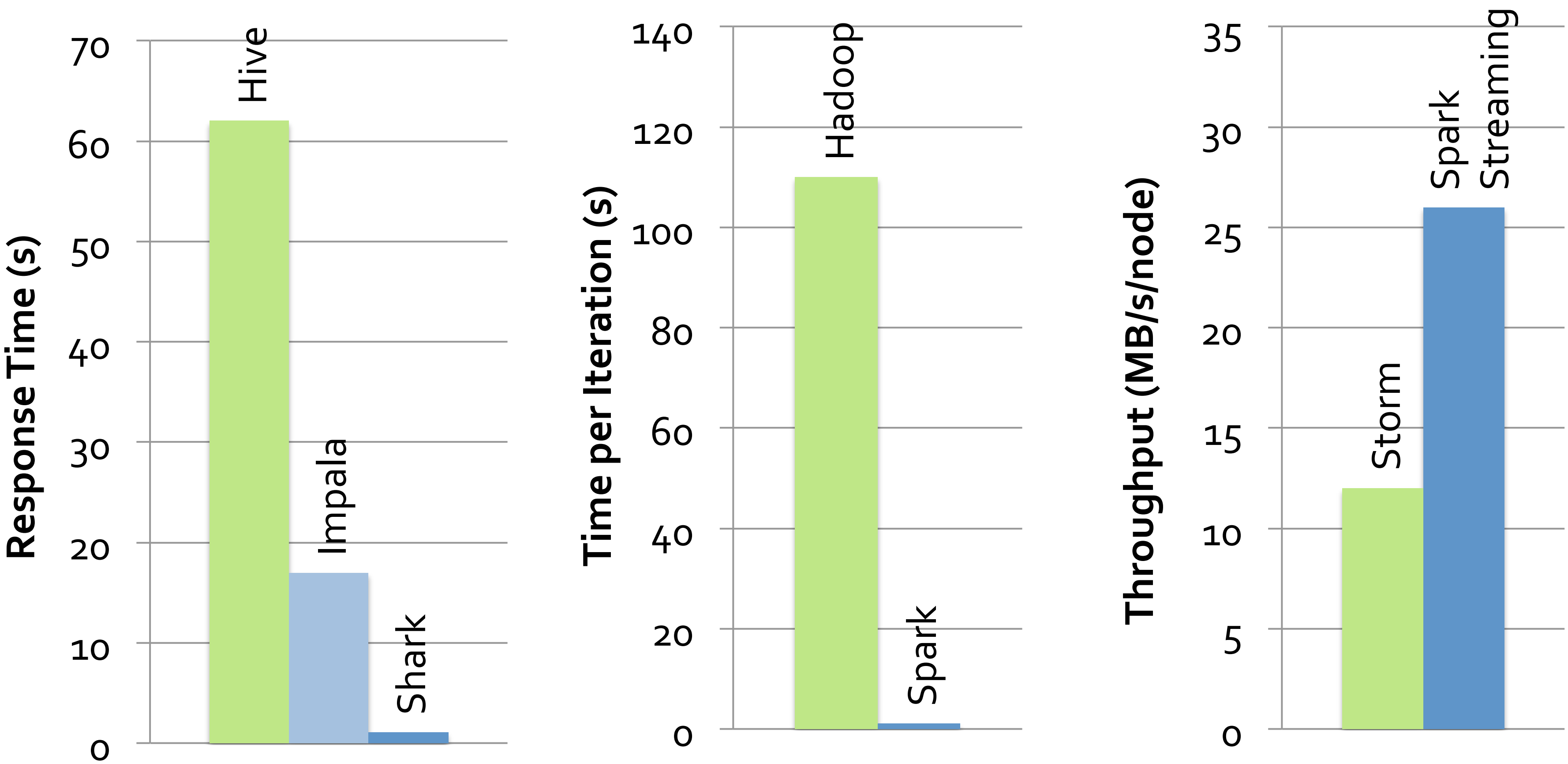
Up to 100x when input is in memory

Up to 5-10x when input is on disk

Running on hundreds of nodes at Yahoo!

Three major releases along Spark

Performance and Generality (Unified Computation Models)



Interactive (SQL, Shark)

Batch (ML, Spark)

Streaming (SparkStreaming)

Unified Programming Models

Unified system for SQL, graph processing, machine learning

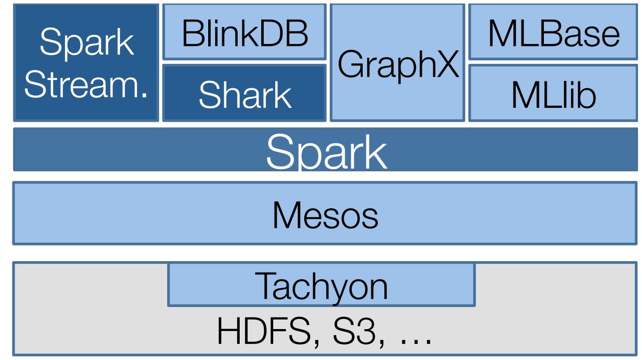
All share the same set of workers and caches

```
def logRegress(points: RDD[Point]): Vector {
  var w = Vector(D, _ => 2 * rand.nextDouble - 1)
  for (i <- 1 to ITERATIONS) {
    val gradient = points.map { p =>
      val denom = 1 + exp(-p.y * (w dot p.x))
      (1 / denom - 1) * p.y * p.x
    }.reduce(_ + _)
    w -= gradient
  }
  w
}

val users = sql2rdd("SELECT * FROM user u
   JOIN comment c ON c.uid=u.uid")

val features = users.mapRows { row =>
  new Vector(extractFeature1(row.getInt("age")),
             extractFeature2(row.getStr("country")),
             ...)}
val trainedVector = logRegress(features.cache())
```

Gaining Rapid Traction



Sold out AMPCamps and Strata tutorials

1,300+ Spark meetup users

20+ companies contributing code

Gaining Rapid Traction

amazon web services

AWS Products & Solutions

Databricks aims to build next-generation analytic tools for Big Data

A new startup will accelerate the maturation of the Berkeley Data Analytics Stack

a | @bigdata | Comment | September 25, 2013

New Cloudera Partner Program Harnesses Power of Innovative Startups Databricks, the Inaugural Partner of Cloudera Connect: Innovators, Teams With Cloudera for High-Speed Data Analytics

© Marketwire 2013

2013-10-28 12:10:03 -

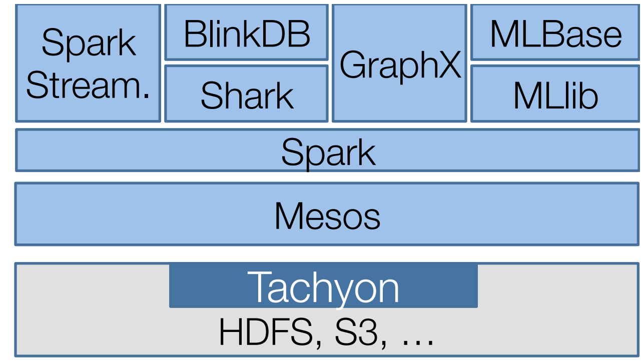
Print article

WANdisco Announces Support for In-Memory Data Processing Technologies, Spark and Shark

MARKET WIRED

Press Release: WANdisco, Plc. – Wed, Jun 26, 2013 9:00 AM EDT

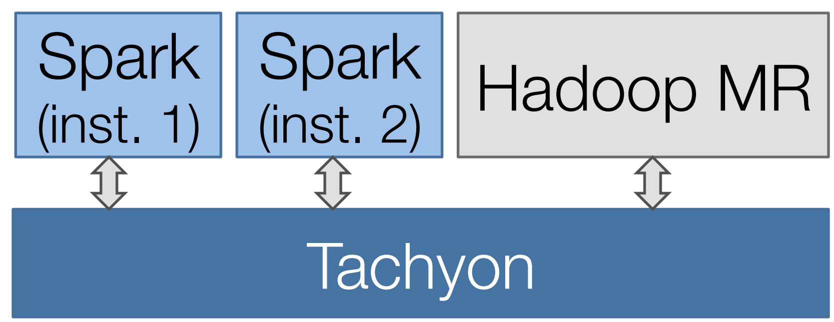
Tachyon



In-memory, fault-tolerant storage system

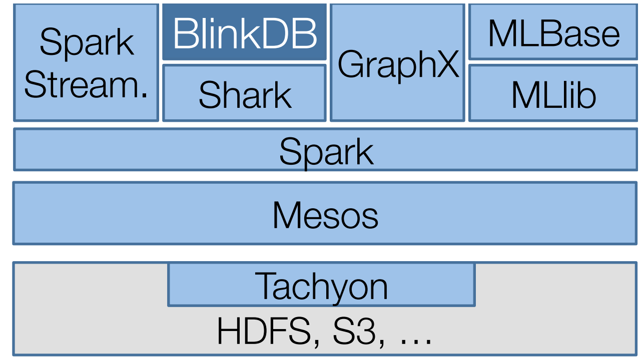
Flexible API, including HDFS API

Allow multiple frameworks (including Hadoop) to share in-memory data

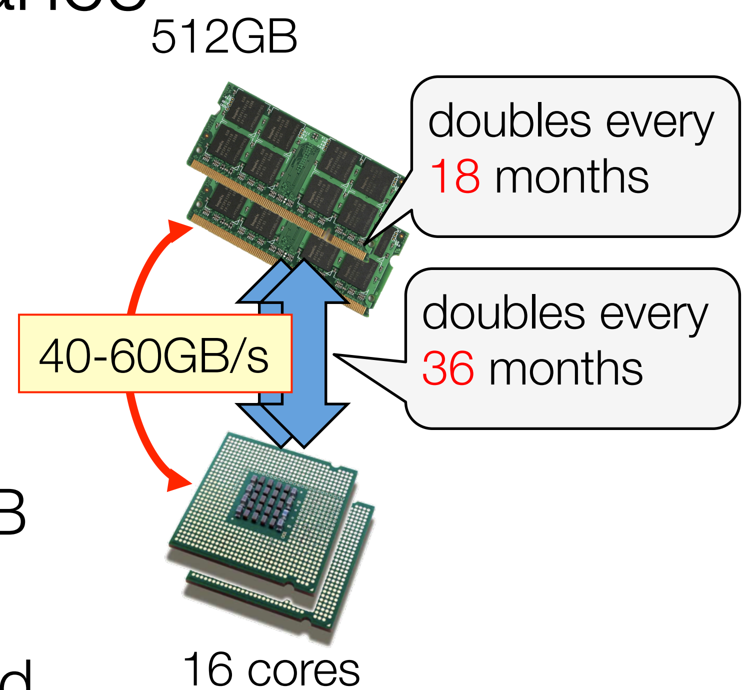


Alpha release (June, 2013)

BlinkDB



Trade between query performance and accuracy using sampling

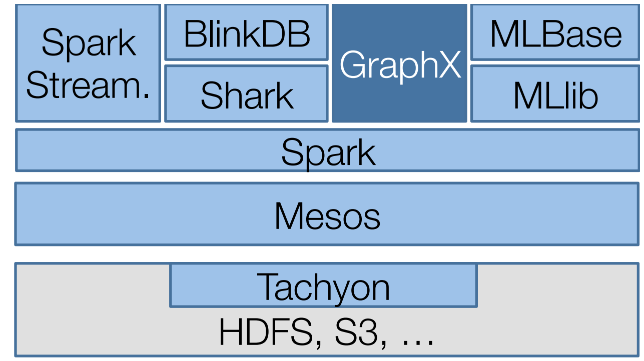


Why?

» In-memory processing doesn't guarantee interactive processing

- E.g., ~10's sec just to scan 512 GB RAM!
- Gap between memory capacity and transfer rate increasing

GraphX



Combine data-parallel and graph-parallel computations

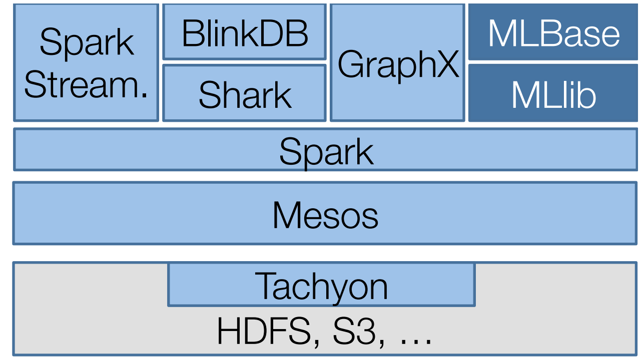
Provide powerful abstractions:

» PowerGraph, Pregel implemented in less than 20 LOC!

Leverage Spark's fault tolerance

Alpha release (Nov., 2013)

MLlib and MLbase



MLlib: high quality library for ML algorithms

» Released with Spark 0.8 (Sept., 2013)

MLbase: make ML accessible to non-experts

» Declarative API: allow users to say what they want

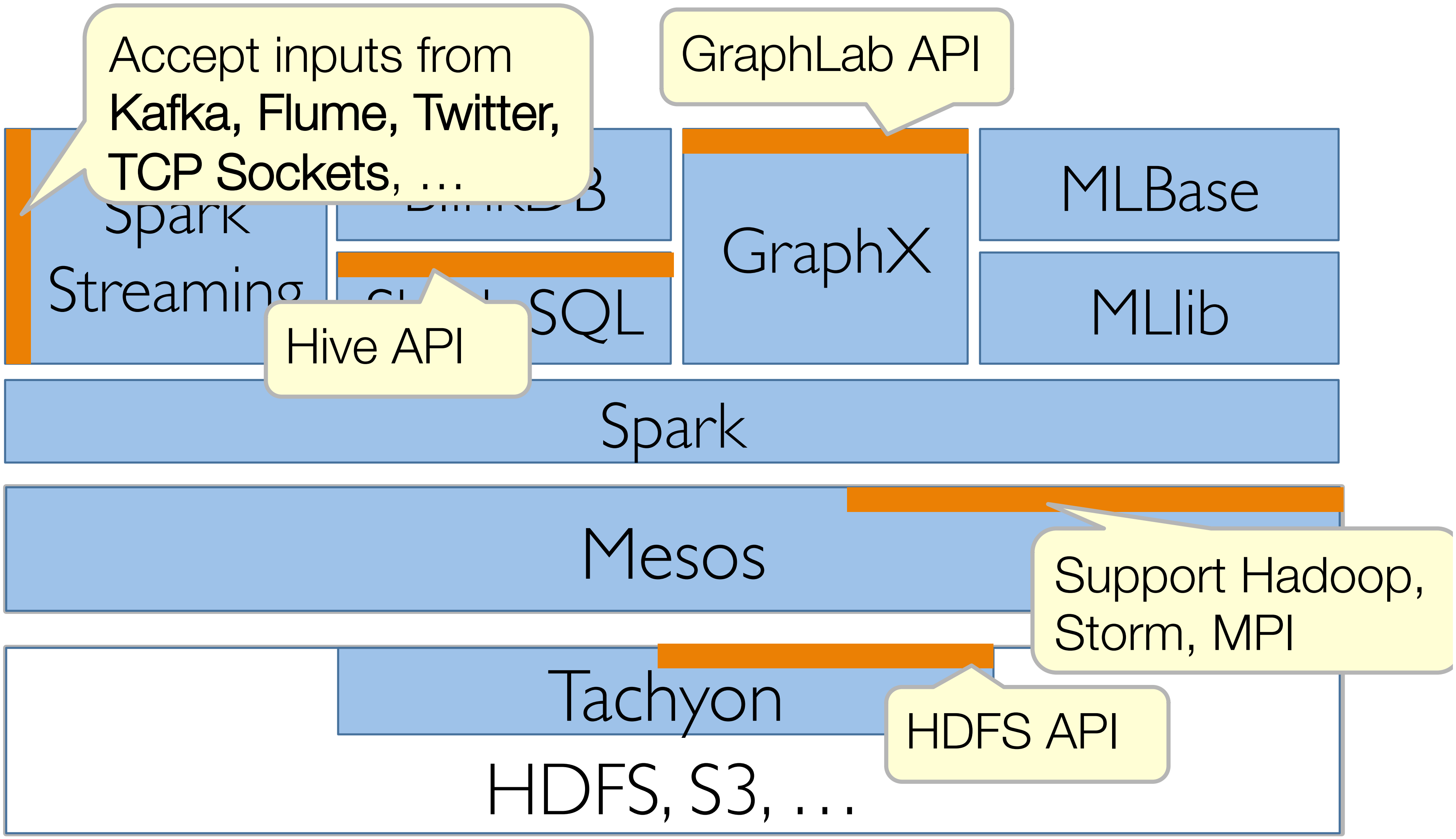
- E.g., classify(data)

» Automatically pick best algorithm for given data, time

» Allow developers to easily add and test new algorithms

» MLI, first component, Alpha release (Sept., 2013)

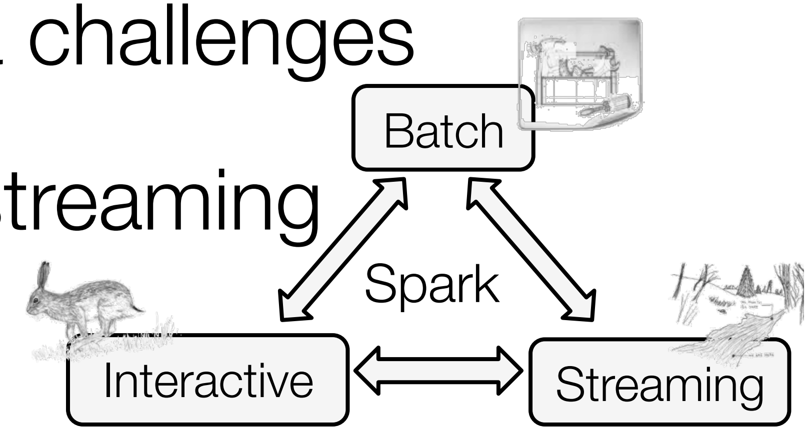
Compatibility to Existing Ecosystem



Summary

BDAS: address next Big Data challenges

Unify batch, interactive, and streaming computations



Easy to develop sophisticate applications

- » Support graph & ML algorithms, approximate queries

Witnessed significant adoption

- » 20+ companies, 90+ individuals contributing code

Exciting ongoing work

- » MLbase, GraphX, BlinkDB, …

Spark Summit

The first event that brings together the Apache Spark community.

December 2: main conference

December 3: hands-on training

www.spark-summit.org

What's Next?

9:00 - 9:25: BDAS overview

9:25 - 10:30: Spark + Shark

10:30 - 11:00: Break

11:00 - 11:30: BlinkDB

11:30 - 12:00: SparkStreaming

12:00 - 12:30: Tachyon