GraphX: Graph Analytics in Spark

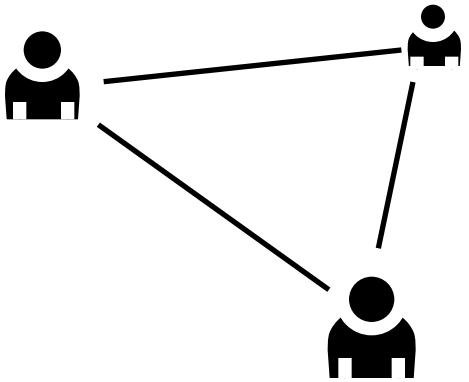
Ankur Dave
Graduate Student, UC Berkeley AMPLab

Joint work with Joseph Gonzalez, Reynold Xin, Daniel Crankshaw, Michael Franklin, and Ion Stoica

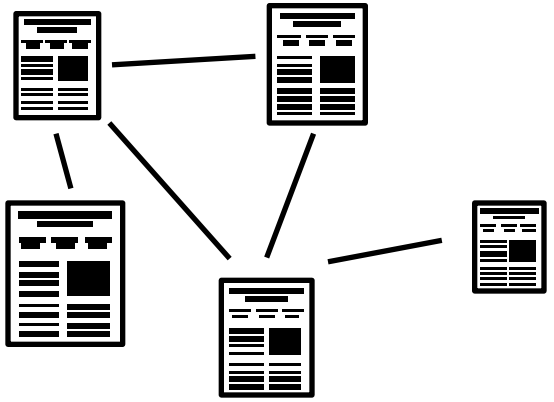
http://www.istc-cc.cmu.edu/

Graphs Are Everywhere

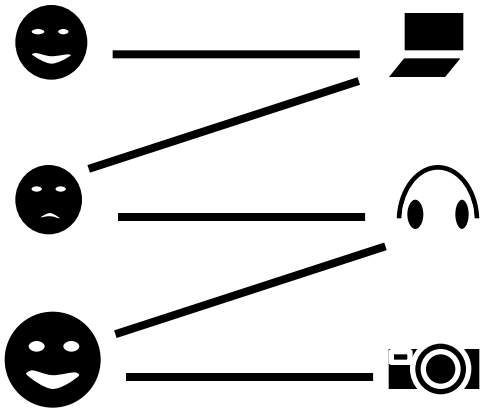
Social Networks



The Web

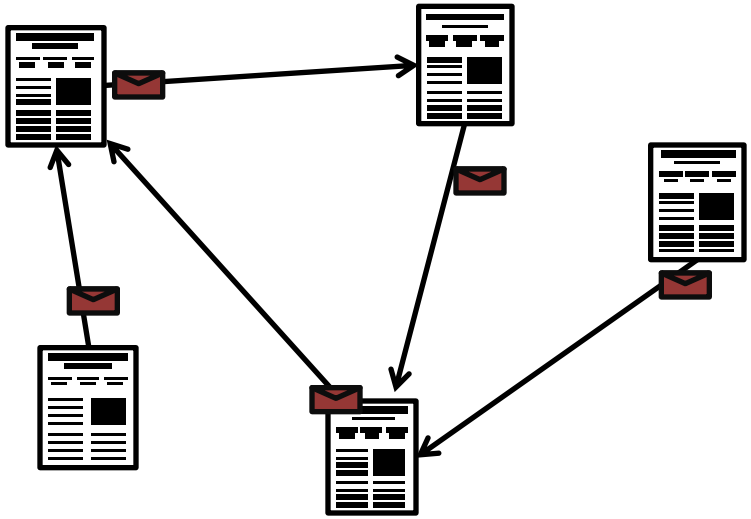


Product Ratings

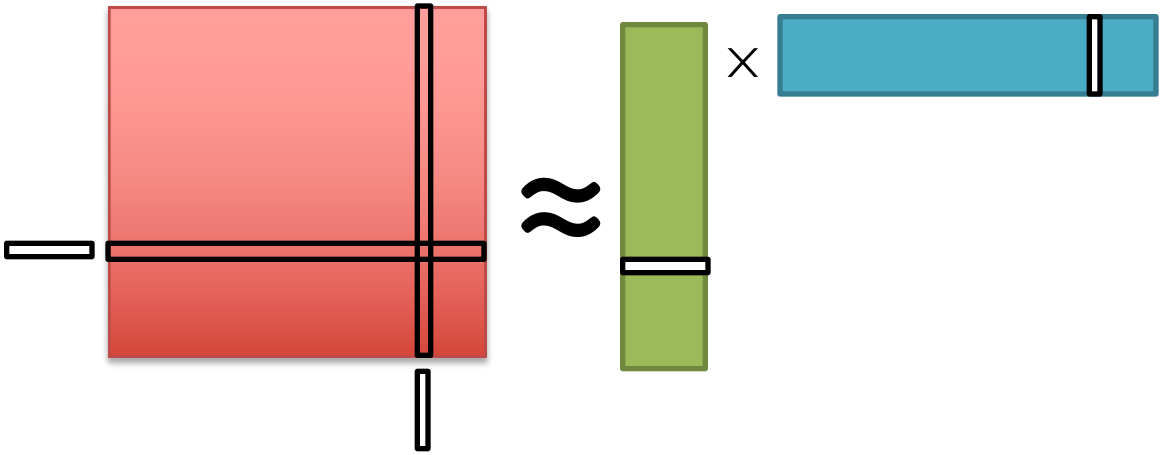


Graph Algorithms

PageRank

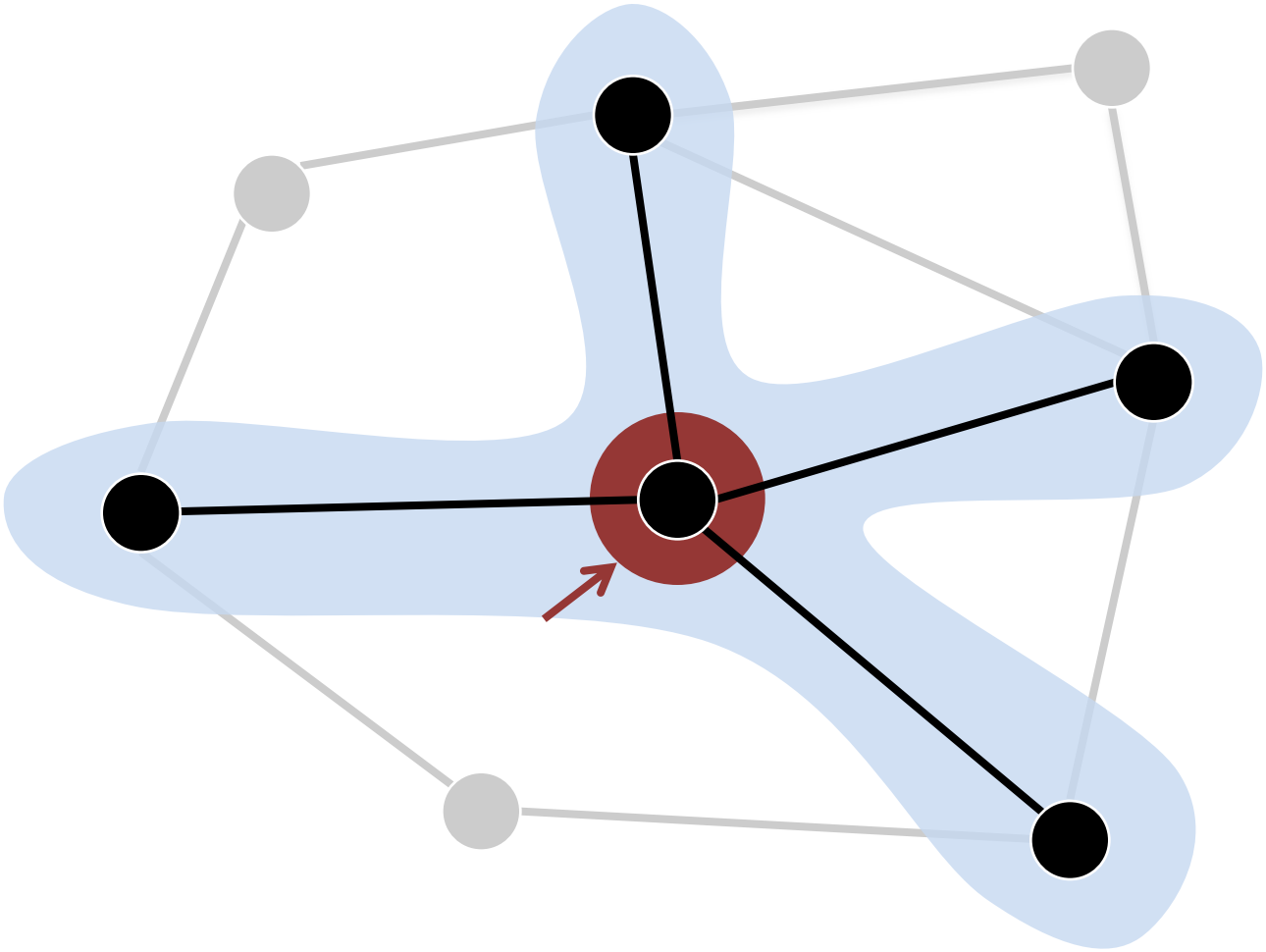


Alternating Least Squares



Challenges

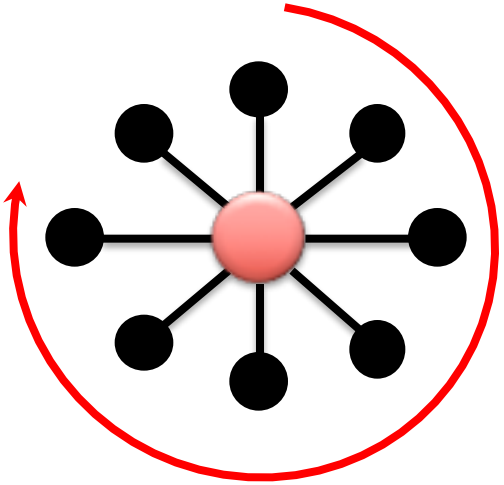
1. Diverse range of graph algorithms

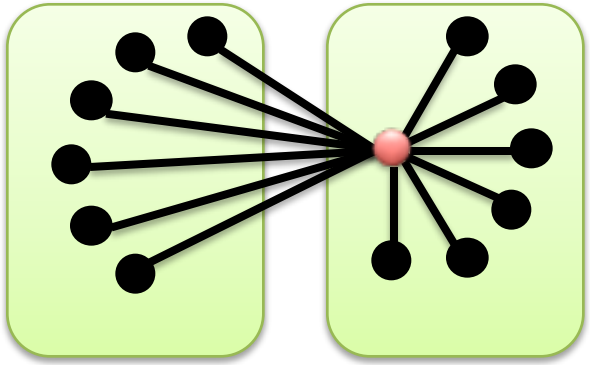


High-Degree Vertices

Taylor Swift
@taylorswift13

FOLLOWERS
51.4M

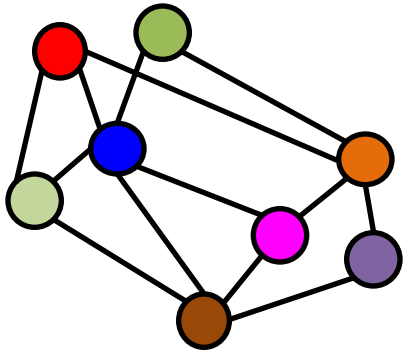




Challenges

1. Diverse range of graph algorithms
2. High-degree vertices

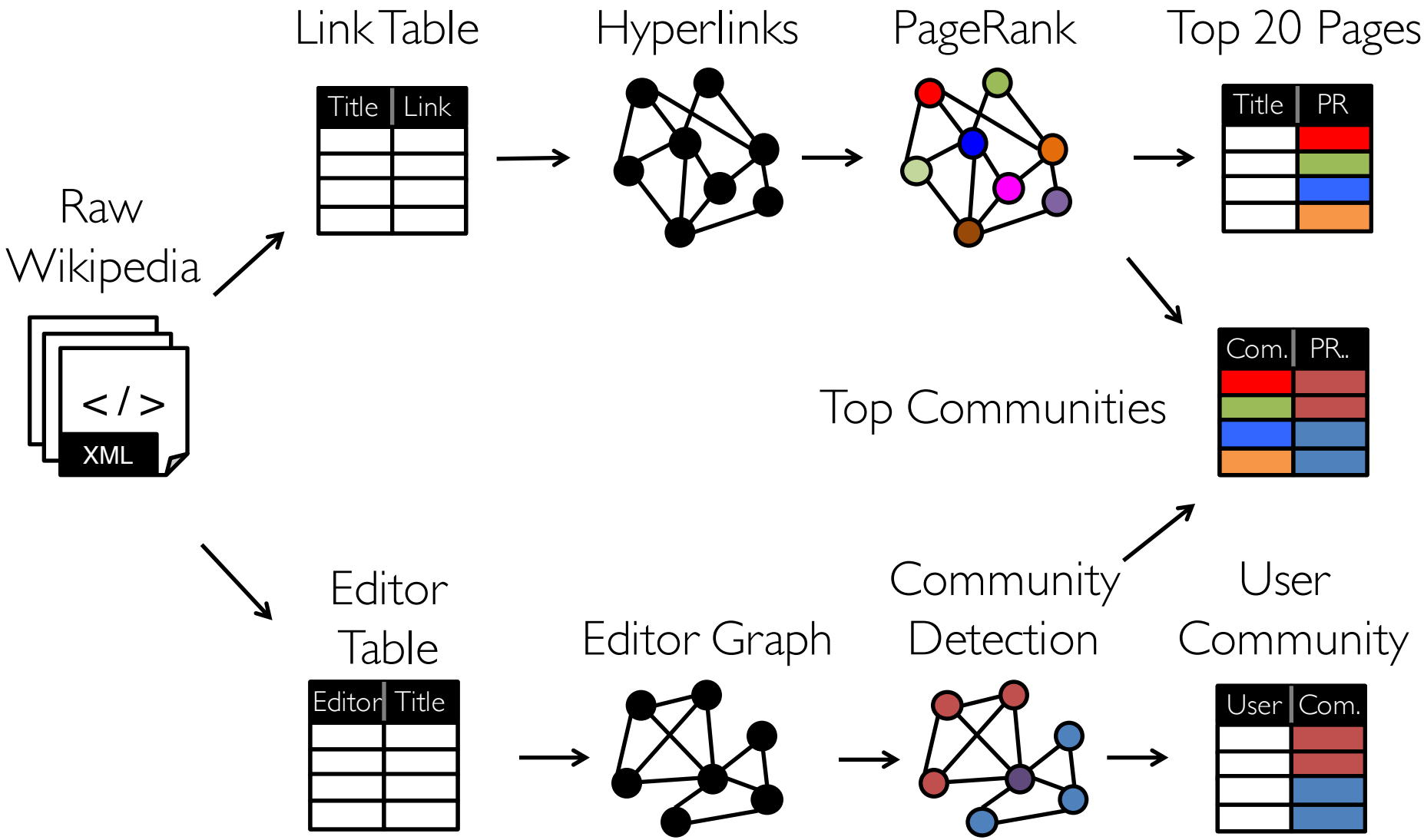
Graph Partitioning Heuristics



Edge-Parallel API

```
def aggregateMessages[A](
    sendMsg: EdgeContext[VD, ED, A]
    mergeMsg: (A, A) => A): RDD[(Ve
```

Complex Pipelines



Challenges

1. Diverse range of graph algorithms
2. High-degree vertices
3. Complex pipelines

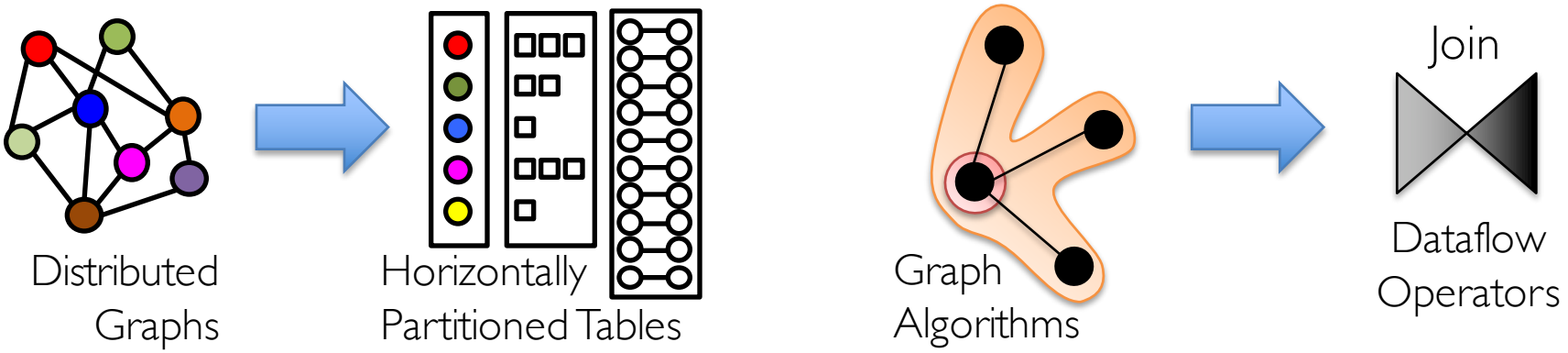


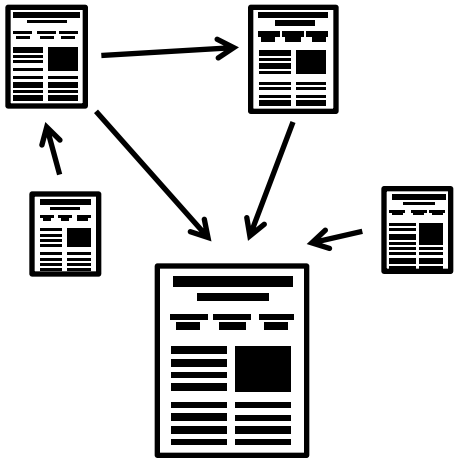
Table-Based Graph API

```
class Graph[VD, ED] {
    // Table Views -----------------------
    def vertices: RDD[(VertexId, VD)]
    def edges: RDD[Edge[ED]]
    def triplets: RDD[EdgeTriplet[VD, ED]]
    // Transformations ------------------------------------------
    def mapVertices[VD2](f: (VertexId, VD) => VD2): Graph[VD2, ED]
    def mapEdges[ED2](f: Edge[ED] => ED2): Graph[VD2, ED]
    def reverse: Graph[VD, ED]
    def subgraph(epred: EdgeTriplet[VD, ED] => Boolean,
                 vpred: (VertexId, VD) => Boolean): Graph[VD, ED]
    // Joins ----------------------------------------
    def outerJoinVertices[U, VD2]
        (tbl: RDD[(VertexId, U)])
        (f: (VertexId, VD, Option[U]) => VD2): Graph[VD2, ED]
    // Computation ----------------------------------
    def aggregateMessages[A](
        sendMsg: EdgeContext[VD, ED, A] => Unit,
        mergeMsg: (A, A) => A): RDD[(VertexId, A)]
```

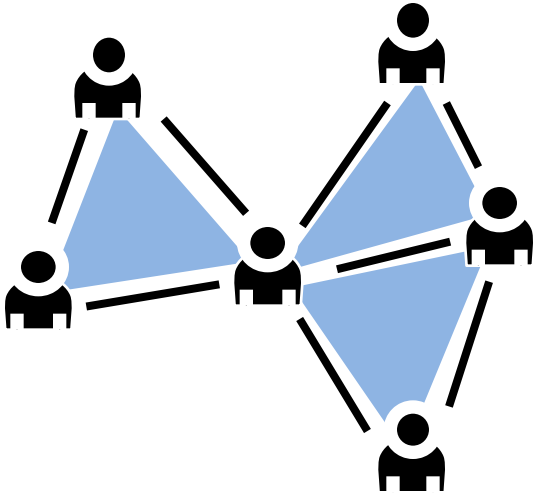
Built-in Algorithms (Scala)

```
    // Continued from previous slide
    def pageRank(tol: Double): Graph[Double, Double]
    def triangleCount(): Graph[Int, ED]
    def connectedComponents(): Graph[VertexId, ED]
    // ...and more: org.apache.spark.graphx.lib
}
```

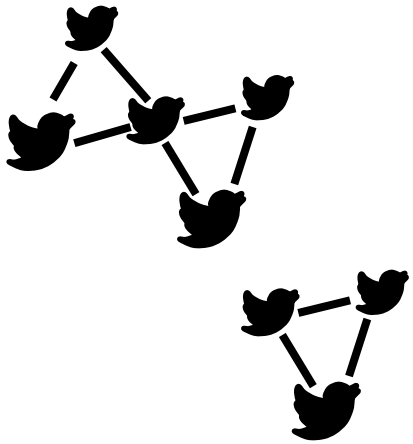
PageRank



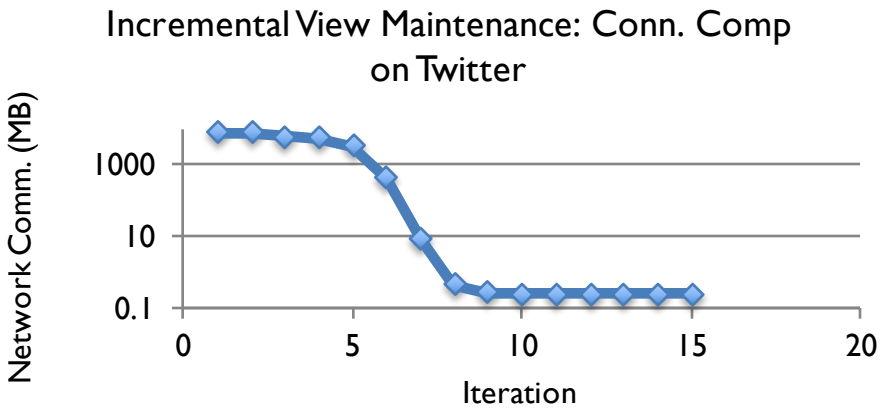
Triangle Count

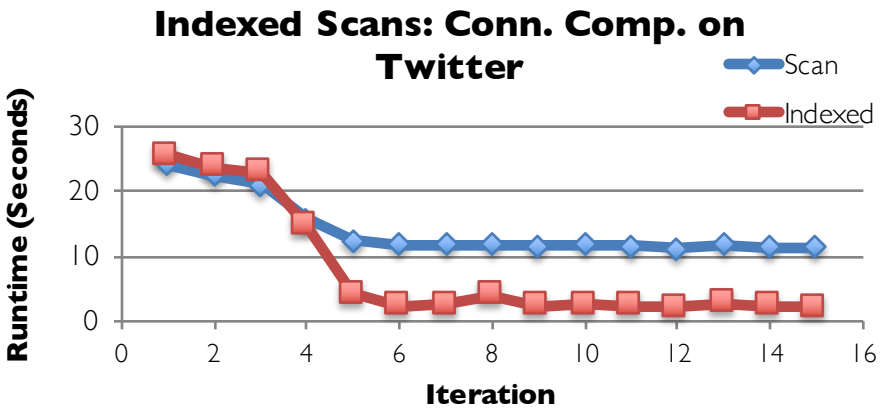


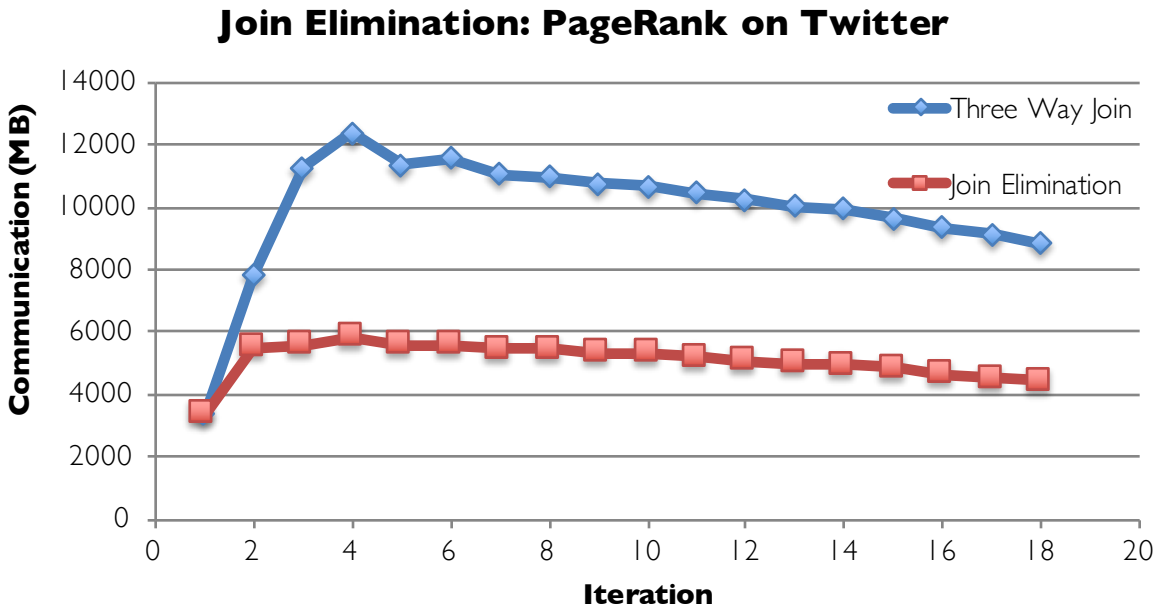
Connected Components



System and Query Optimizations

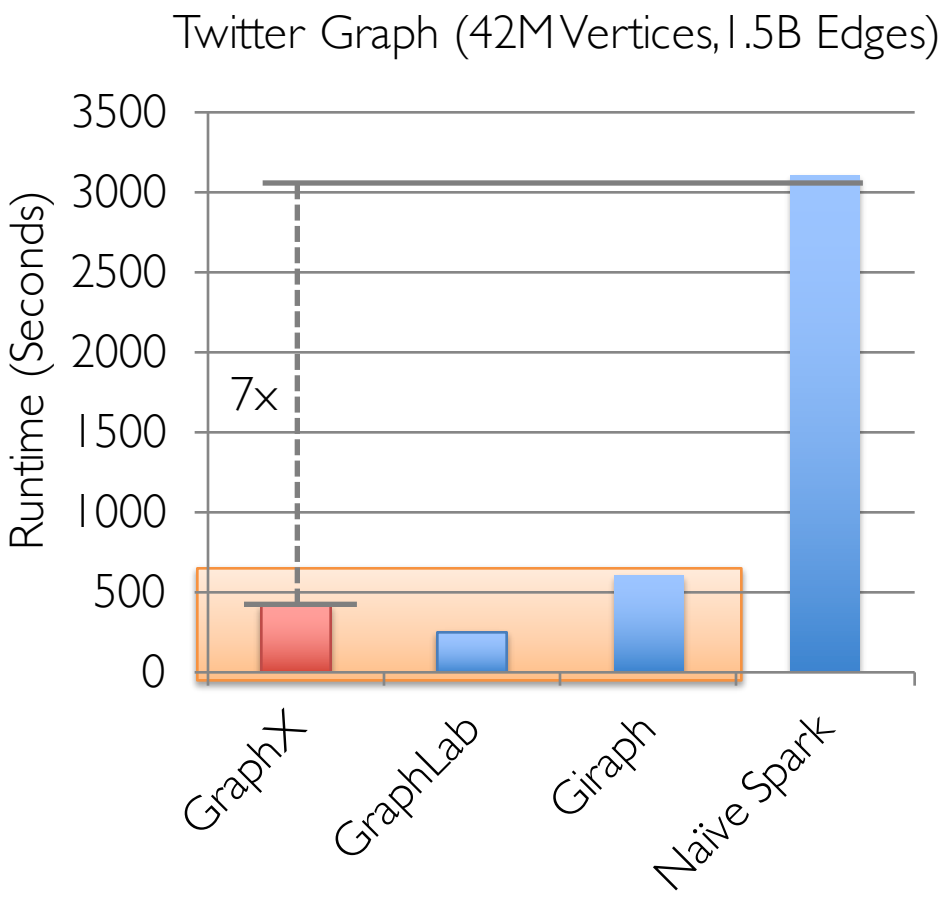


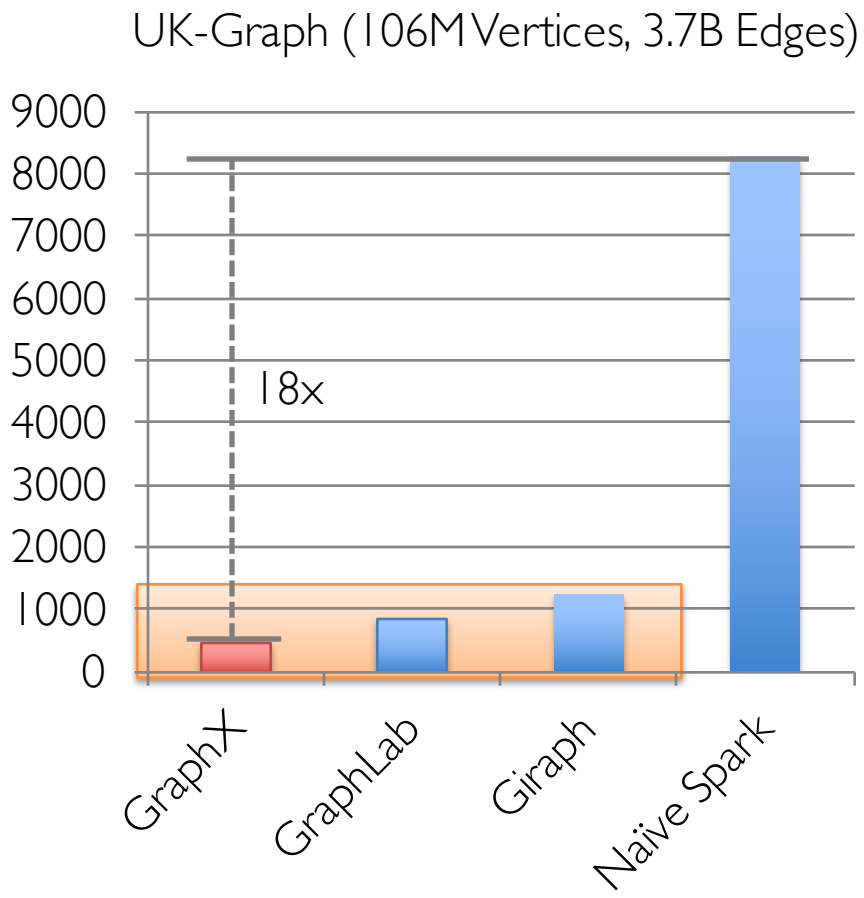




PageRank Benchmark

EC2 Cluster of 16 x m2.4xLarge (8 cores) + 1GigE

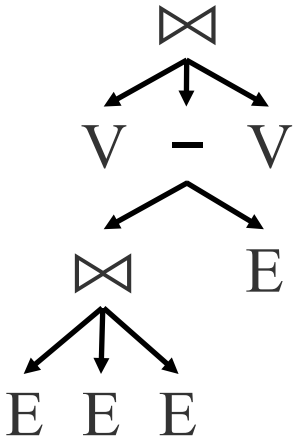




GraphFrames

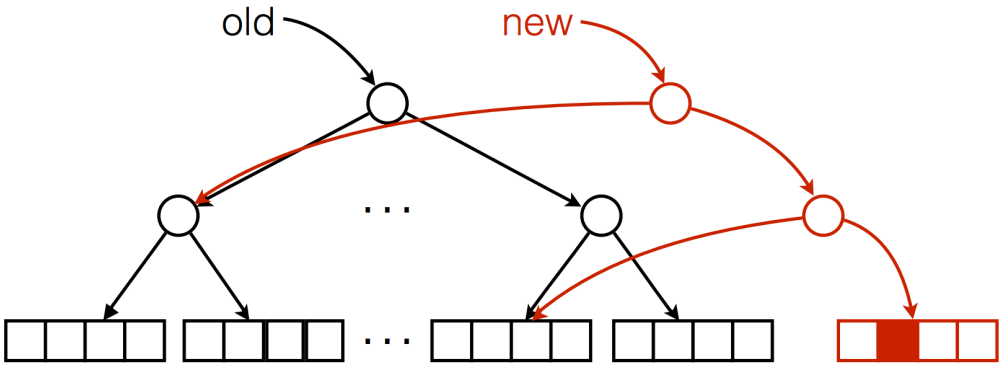
Support analytical graph queries using Spark SQL and graph-specific join optimizations

```
graph.find("(u)->(v)->(w); !(u)->(w)")
  .filter($"u_school" === $"w_school")
```

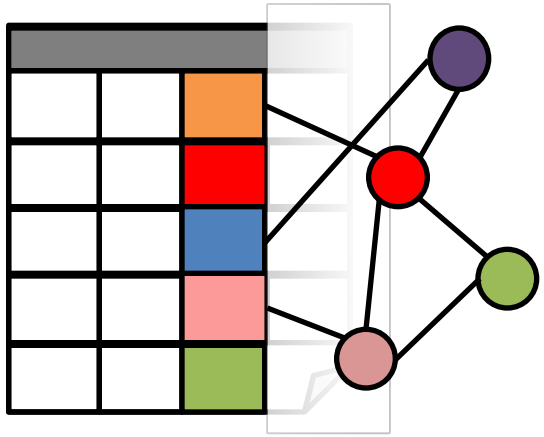


IndexedRDD

Support efficient updates to immutable RDDs using purely functional data structures



https://github.com/amplab/spark-indexedrdd



Thanks!

Joseph E. Gonzalez, Daniel Crankshaw, Ankur Dave, Reynold S. Xin, Michael J. Franklin, Ion Stoica. GraphX: Graph Processing in a Distributed Dataflow Framework. In OSDI 2014.

amplab
UC BERKELEY

Storing Graphs as Tables

Property Graph