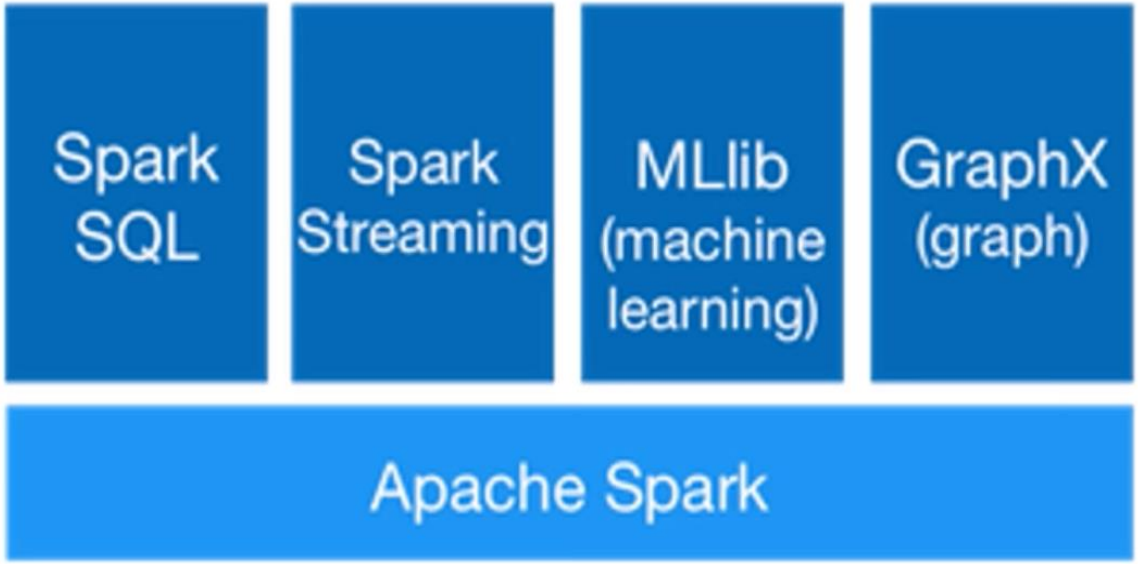
Spark Built-in Libraries

Dr. Rajiv Misra
Dept. of Computer Science & Engg.
Indian Institute of Technology Patna
rajivm@iitp.ac.in

Introduction

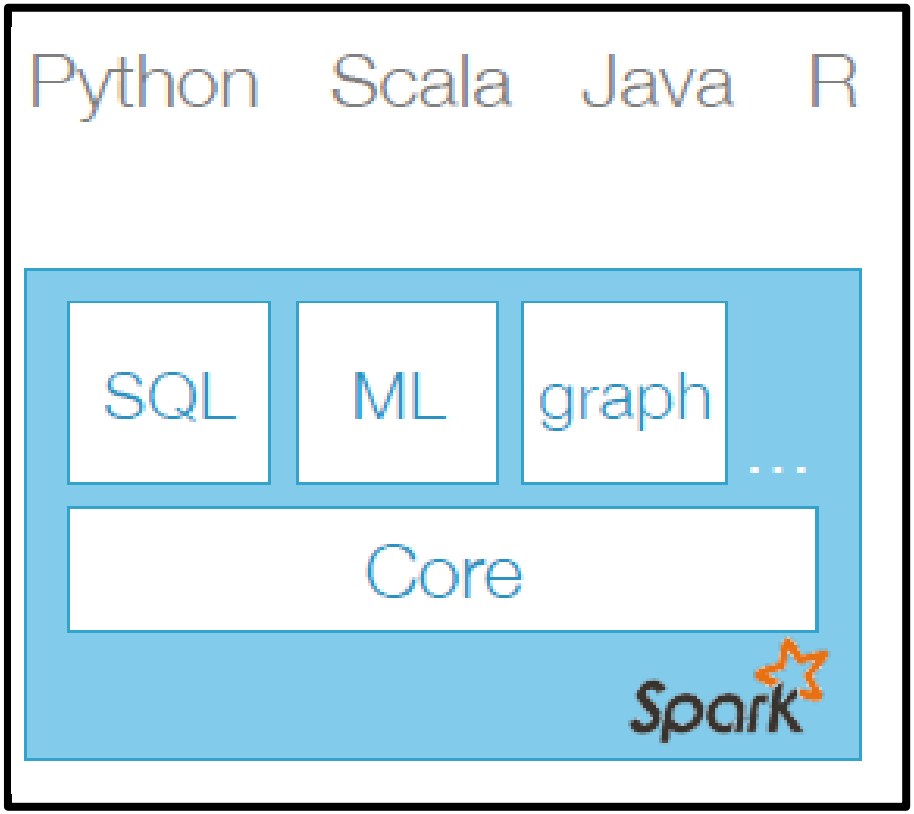
- Apache Spark is a fast and general-purpose cluster computing system for large scale data processing

- High-level APIs in Java, Scala, Python and R

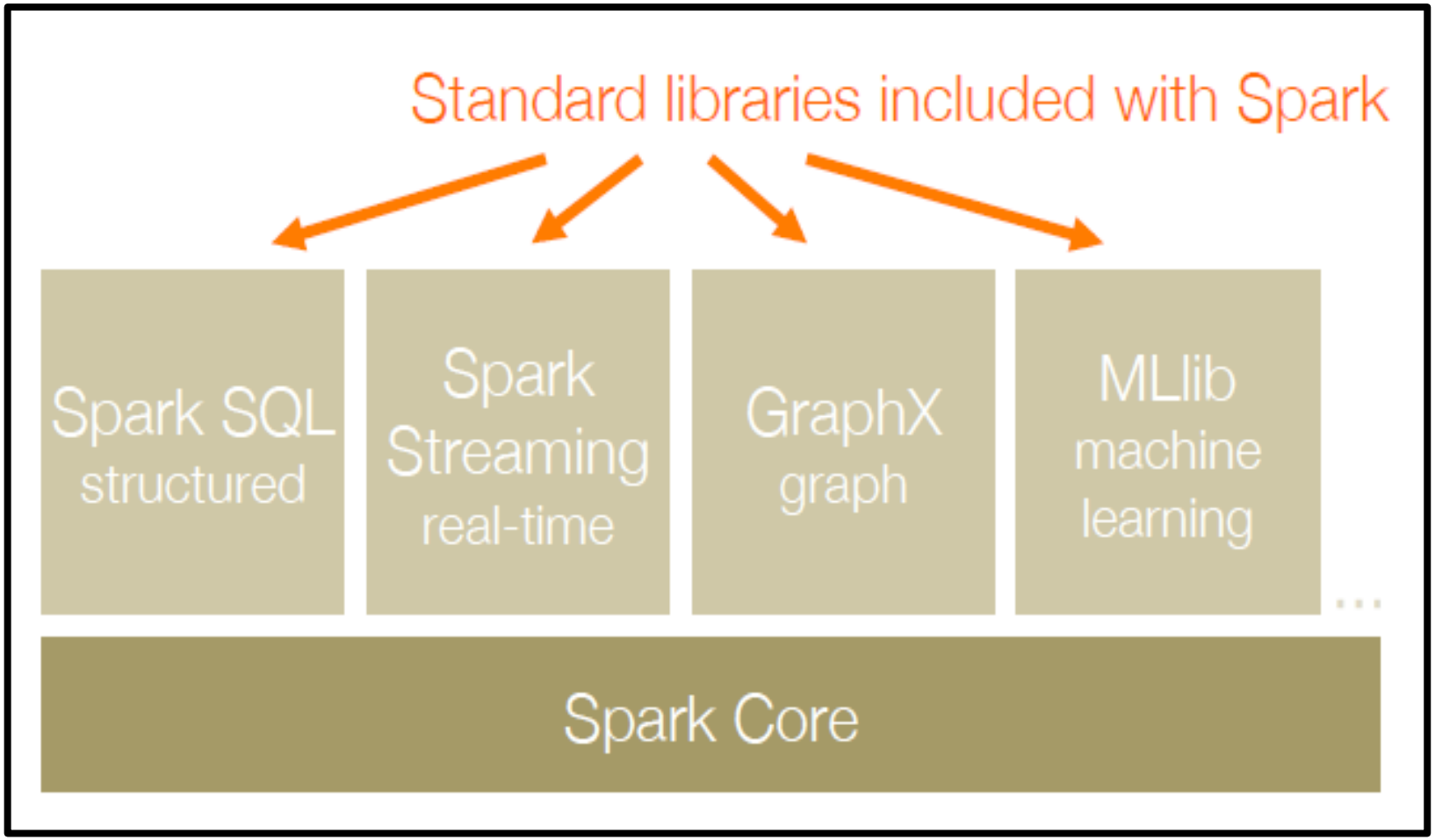


Standard Library for Big Data

- Big data apps lack libraries“ of common algorithms
- Spark’s generality + support“ for multiple languages make it“ suitable to offer this
- Much of future activity will be in these libraries



A General Platform



Machine Learning Library (MLlib)

MLlib algorithms:

(i) Classification: logistic regression, linear SVM," naïve Bayes, classification tree

(ii) Regression: generalized linear models (GLMs), regression tree

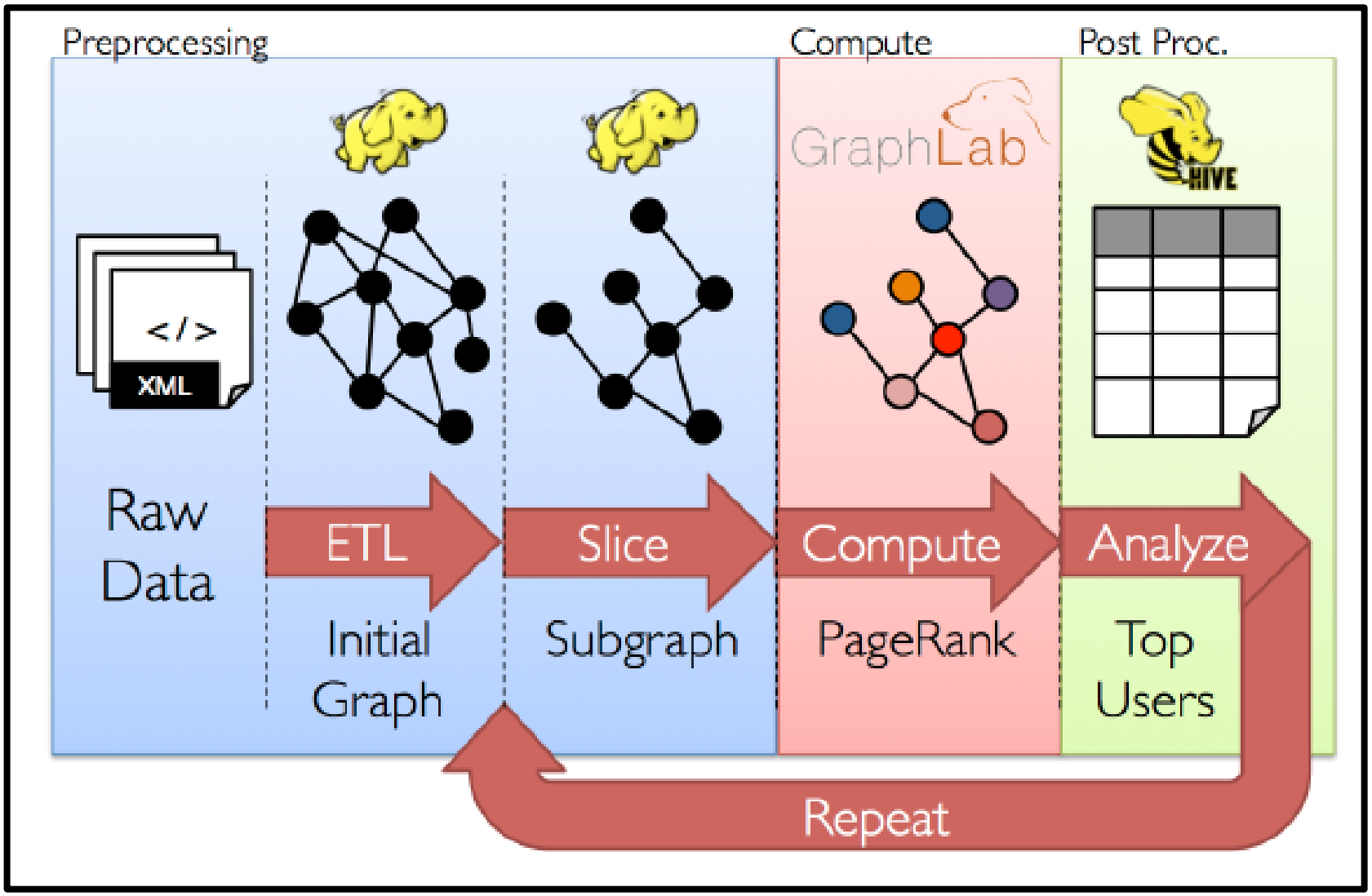
(iii) Collaborative filtering: alternating least squares (ALS), non-negative matrix factorization (NMF)

(iv) Clustering: k-means

(v) Decomposition: SVD, PCA

(vi) Optimization: stochastic gradient descent, L-BFGS

GraphX


Preprocessing
Compute
Post Proc.
GraphLab
HIVE
XML
Raw Data
ETL
Slice
Compute
Analyze
Initial Graph
Subgraph
PageRank
Top Users
Repeat

GraphX

- General graph processing library
- Build graph using RDDs of nodes and edges
- Large library of graph algorithms with composable steps

GraphX Algorithms

(i) Collaborative Filtering
Alternating Least Squares
Stochastic Gradient Descent
Tensor Factorization

(ii) Structured Prediction
Loopy Belief Propagation
Max-Product Linear Programs
Gibbs Sampling

(iii) Semi-supervised ML
Graph SSL
CoEM

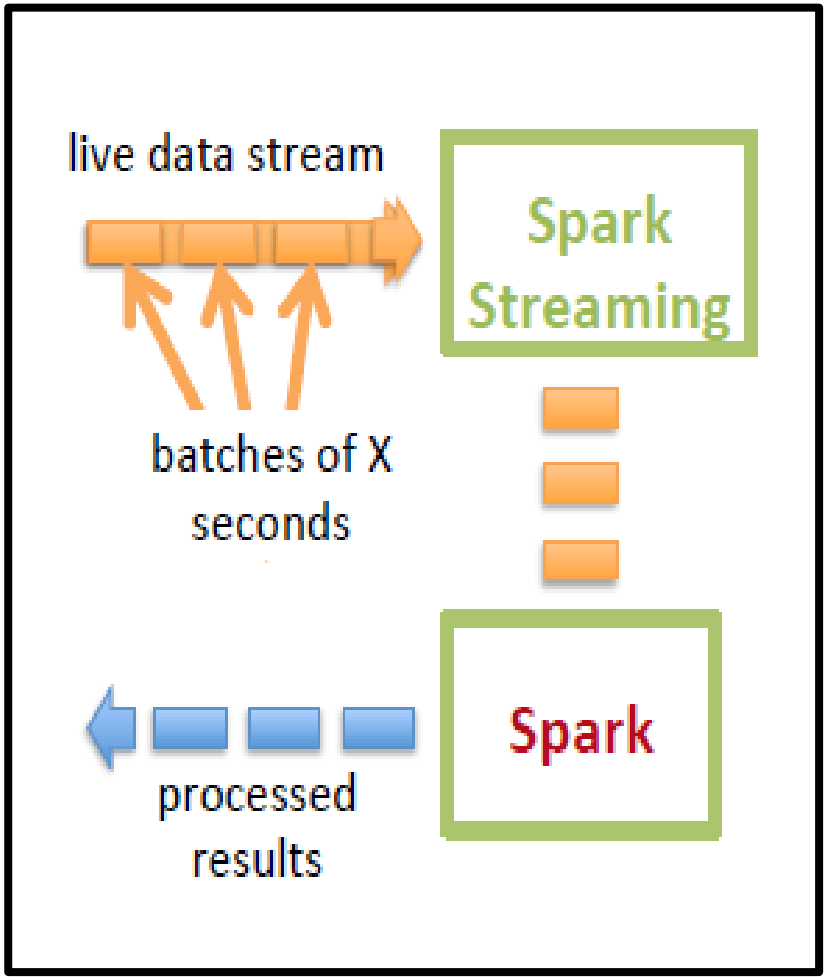
(iv) Community Detection
Triangle-Counting
K-core Decomposition
K-Truss

(v) Graph Analytics
PageRank
Personalized PageRank
Shortest Path
Graph Coloring

(vi) Classification
Neural Networks

Spark Streaming

- Large scale streaming computation
- Ensure exactly one semantics
- Integrated with Spark → unifies batch, interactive, and streaming computations!



Spark SQL

Enables loading & querying structured data in Spark

From Hive:

```
c = HiveContext(sc)
rows = c.sql("select text, year from hivetable")
rows.filter(lambda r: r.year > 2013).collect()
```

From JSON:

```
c.jsonFile("tweets.json").registerAsTable("tweets")
c.sql("select text, user.name from tweets")
```

Spark Community

- Most active open source community in big data
- 200+ developers, 50+ companies contributing