Gelly School

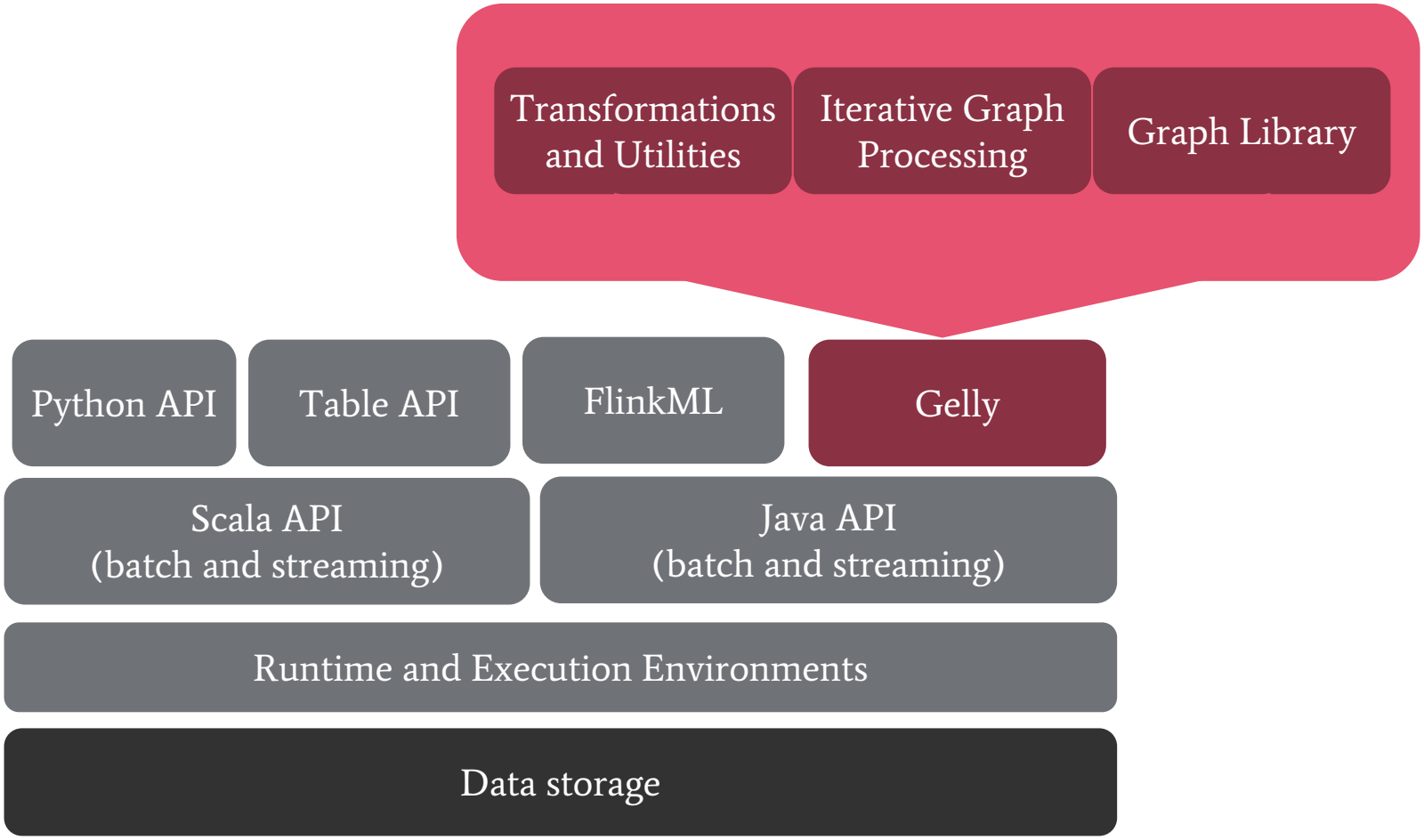
Vasia Kalavri
Apache Flink PMC, PhD student @KTH
kalavri@kth.se, @vkalavri

Meet Gelly

- Java & Scala Graph APIs on top of Flink
- Library of common Graph algorithms
- Iteration abstractions
- Can be seamlessly mixed with the DataSet Flink API

  → easily implement applications that use both *record-based* and *graph-based* analysis

Gelly in the Flink stack



Hello, Gelly!

Java

```
ExecutionEnvironment env = ExecutionEnvironment.getExecutionEnvironment();
DataSet<Edge<Long, NullValue>> edges = getEdgesDataSet(env);

Graph<Long, Long, NullValue> graph = Graph.fromDataSet(edges, env);
DataSet<Vertex<Long, Long>> verticesWithMinIds = graph.run(
                                    new ConnectedComponents(maxIterations));
```

Scala

```
val env = ExecutionEnvironment.getExecutionEnvironment
val edges: DataSet[Edge[Long, NullValue]] = getEdgesDataSet(env)

val graph = Graph.fromDataSet(edges, env)
val components = graph.run(new ConnectedComponents(maxIterations))
```

Tasks for Today

Go to http://gellyschool.com/flink-forward

- **Tutorial#0:** Why Graphs and Gelly Basics
- **Tutorial#1:** Calculate Degree Distributions
- **Tutorial#2:** PageRank
- **Tutorial#3:** People you Might Know

Skeleton: http://github.com/vasia/gelly-school/tree/**ff-skeleton**

Solutions: http://github.com/vasia/gelly-school/tree/**ff-solutions**

...or in the home folder of your VM :-)

Today you will learn how to...

Create a Graph from a file of edges

Compute simple Graph properties

Use Gelly's neighborhood methods

Run Gelly library algorithms

Use DataSet and Gelly APIs together

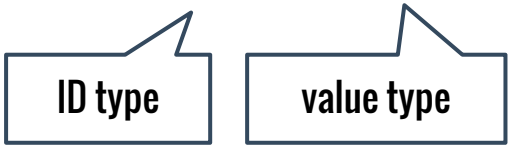
Today you will *not* learn how to...

Use the Scala Gelly API

Write your own vertex-centric or gather-sum-apply iterative programs

Tutorial#0: Let’s get Started!

```
// create a vertex with ID=42 and value=0.8
Vertex<Integer, Double> v = new Vertex<Integer, Double>(42, 0.8);
```



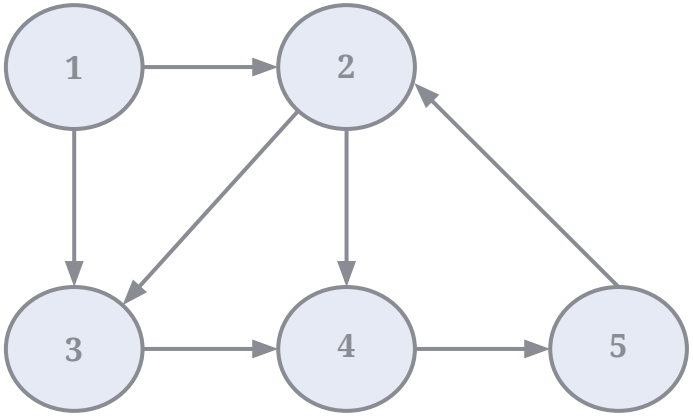

```
// create an edge from 5 to 6 with value="foo"
Edge<Integer, Integer, String> e =
    new Edge<Integer, Integer, String>(5, 6, "foo");
```


```
// create an edge from 5 to 6 with no value
Edge<Integer, Integer, NullValue> e =
    new Edge<Integer, Integer, NullValue>(5, 6, NullValue.getInstance());
```

```
ExecutionEnvironment env = ExecutionEnvironment.getExecutionEnvironment();

// create a Graph from Vertex and Edge DataSets
DataSet<Vertex<String, Long>> vertices = ...
DataSet<Edge<String, Double>> edges = ...

Graph<String, Long, Double> g1 = Graph.fromDataSet(vertices, edges, env);


...

// create a Graph from a Tuple3 DataSet
DataSet<Tuple3<String, String, Double>> edges = ...

Graph<String, NullValue, Double> g2 = Graph.fromTupleDataSet(edges, env);
```

```
// create a Graph from a Tuple2 DataSet
DataSet<Tuple2<String, String>> input = ...

DataSet<Edge<String, NullValue>> edges = input.map(
    new MapFunction<Tuple2<String, String>, Edge<String, NullValue>>() {
      public Edge<String, NullValue> map(Tuple2<String, String> in) {
        return new Edge(in.f0, in.f1, NullValue.getInstance());
        }
    })

Graph<String, NullValue, NullValue> g3 = Graph.fromDataSet(edges, env);
```

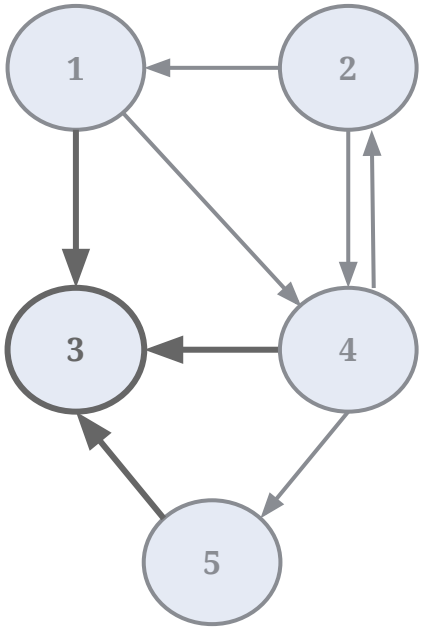
Tutorial#1: Degree Distributions



vertexID	in-degree	out-degree	degree
1	0	2	2
2	2	2	4
3	2	1	3
4	2	1	3
5	1	1	2

degree	#vertices	distribution
2	2	2/5
3	2	2/5
4	1	1/5

Tutorial#2: PageRank



vertexID	out-degree	transition probability
1	2	1/2
2	2	1/2
3	0	-
4	3	1/3
5	1	1

simplified PageRank

PR(3) = 0.5*PR(1) + 0.33*PR(4) + PR(5)

```
Graph<Long, Double, Double> network = ...

DataSet<Tuple2<Long, Long>> vertexOutDegrees = network.outDegrees();

// assign the transition probabilities as the edge weights
Graph<Long, Double, Double> networkWithWeights =
    network.joinWithEdgesOnSource(vertexOutDegrees,
                new MapFunction<Tuple2<Double, Long>, Double>() {
                    public Double map(Tuple2<Double, Long> value) {
                        return value.f0 / value.f0;
                    }
                });
```

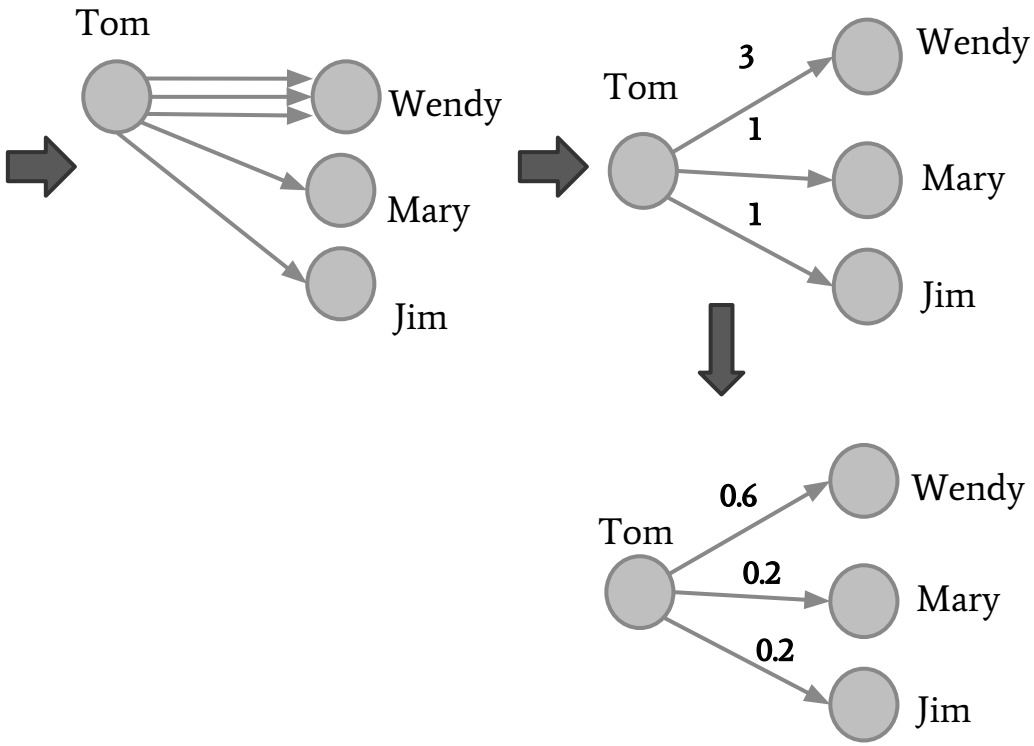
current Edge value

value from degrees

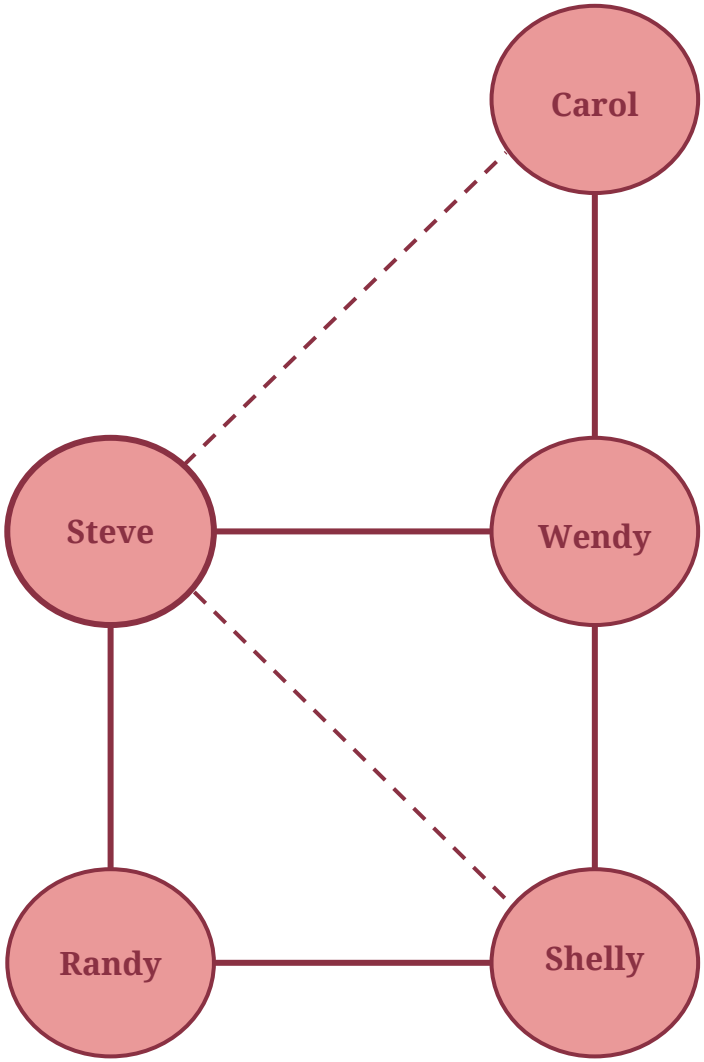
new Edge value

Interactions as weights

Tom RT Wendy
Tom RT Mary
Tom RT Wendy
Sarah RT James
Tom RT Jim
Tom RT Wendy
Jim RT Obama
...



Tutorial#3: People you Might Know



Steve knows Wendy
Wendy knows Carol
→ Steve knows Carol?

Steve knows Randy
Randy and Wendy know Shelly
→ Steve knows Shelly?

Feeling Gelly?

Gelly programming guide: https://ci.apache.org/projects/flink/flink-docs-master/libs/gelly_guide.html

Blog post: https://flink.apache.org/news/2015/08/24/introducing-flink-gelly.html

More Gelly School: http://gellyschool.com