第十六讲　Giraph编程

徐 辰
cxu@dase.ecnu.edu.cn

华东师范大学

DaSE
Data Science
&Engineering

大纲

2

- 编程方式
- 编程接口
- 编程实例

应用程序编写流程

3

- 在IDE中编写Java程序
- 利用IDE打包jar
- 命令行提交jar包

大纲

4

- 编程方式
- 编程接口
- 编程实例

Programming Sketch

5

```
public class xxx extends BasicComputation
    < ID类型, value类型, weight类型, message类型> {

    public void compute(
        Vertex<ID类型, value类型, weight类型> vertex,
        Iterable<message类型> messages) {
            ……
    }
}
```

大纲

6

- 编程方式
- 编程接口
- 编程实例
  - 最短路径
  - 最大值
  - 统计边的个数
  - K-Means

最短路径

7

```java
public void compute(Vertex<LongWritable, DoubleWritable, FloatWritable>
vertex, Iterable<DoubleWritable> messages) {
    double minDist = isSource(vertex) ? 0d : Double.MAX_VALUE;
    for (DoubleWritable message : messages) {
        minDist = Math.min(minDist, message.get());
    }

    if (minDist < vertex.getValue().get()) {
        vertex.setValue(new DoubleWritable(minDist));
        for (Edge<LongWritable, FloatWritable> edge : vertex.getEdges()) {
            double distance = minDist + edge.getValue().get();
            sendMessage(edge.getTargetVertexId(), new DoubleWritable(distance));
        }
    }
    vertex.voteToHalt();
}
```
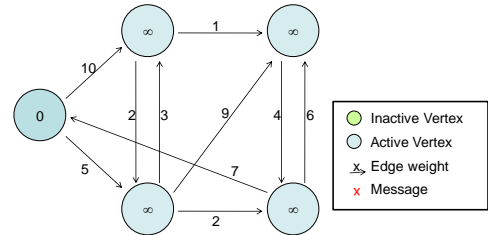
https://github.com/apache/giraph/blob/trunk/giraph-examples/src/main/java/org/apache/giraph/examples/SimpleShortestPathsComputation.java
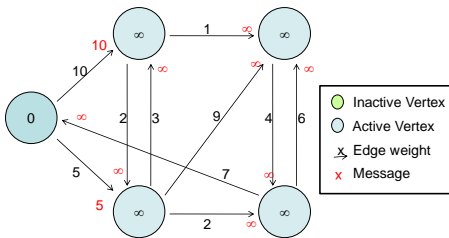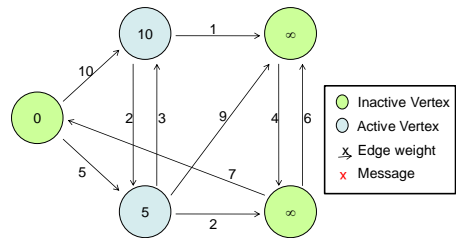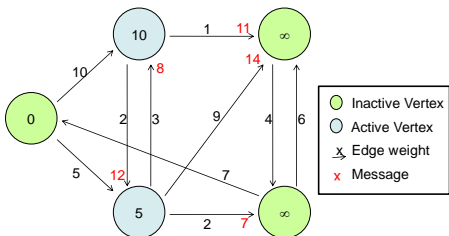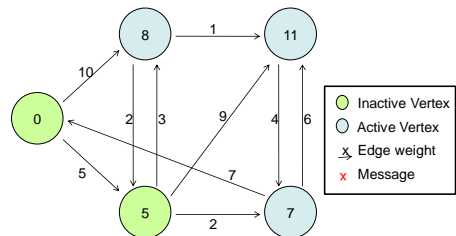
SSSP

8

SSSP

9

SSSP

10

SSSP

11

SSSP

12

SSSP

SSSP

SSSP

SSSP

大纲
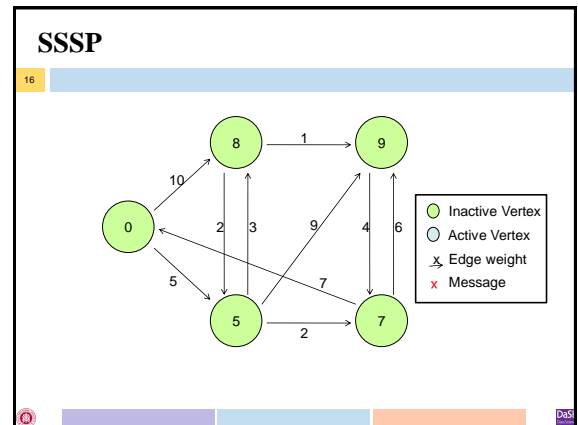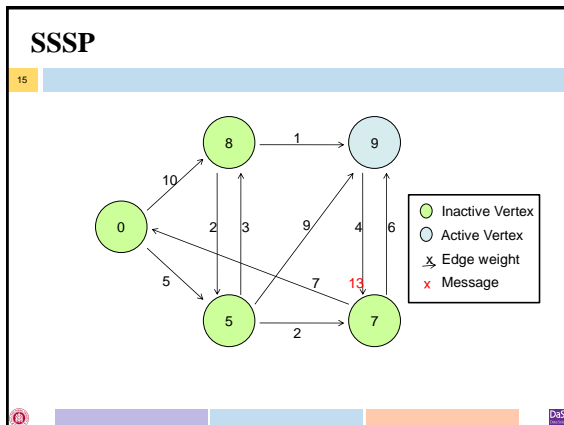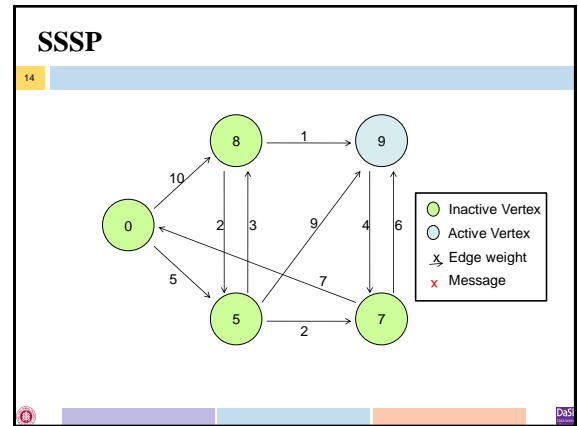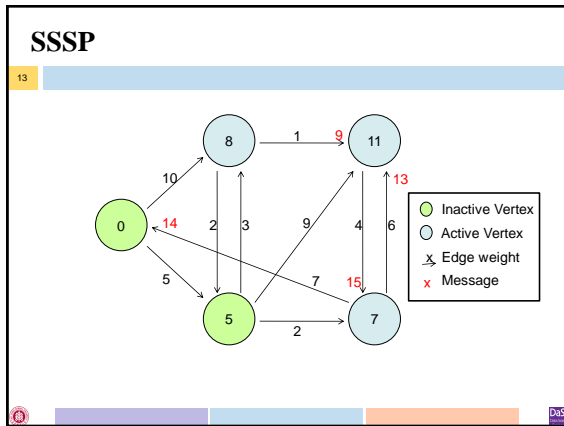
- ❑ 编程方式
- ❑ 编程接口
- ❑ 编程实例
  - ↓ 最短路径
  - ↓ 最大值
  - ↓ 统计边的个数
  - ↓ K-Means
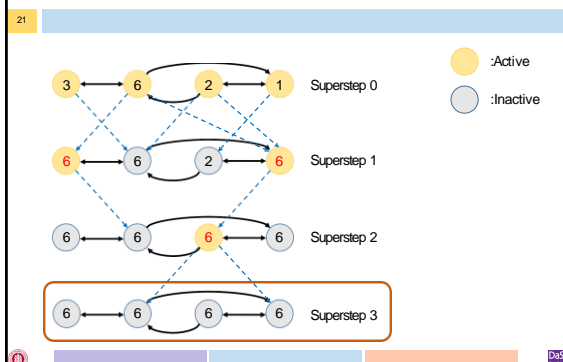
Maximum value (1/4)
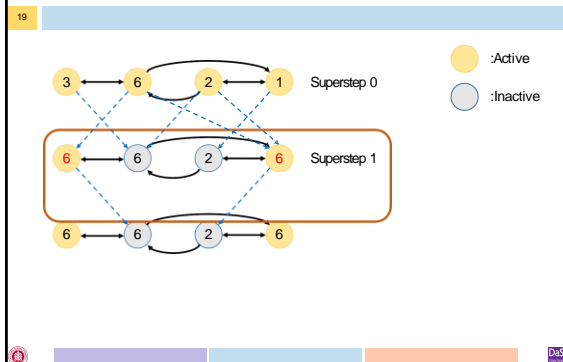
Maximum value (2/4)

19

Supersteps diagram with Active (yellow) and Inactive (gray) legend.

Maximum value (3/4)

20

Supersteps diagram with Active (yellow) and Inactive (gray) legend.

Maximum value (4/4)

21

Supersteps diagram with Active (yellow) and Inactive (gray) legend.

最大值

22
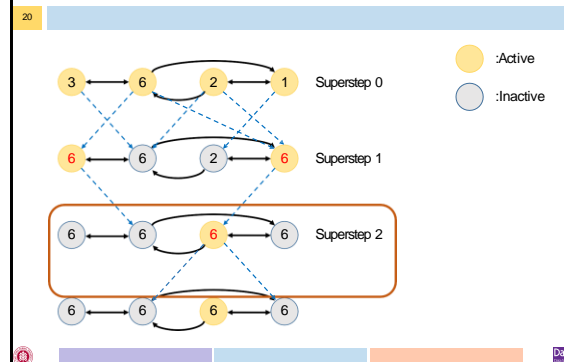
```java
public void compute(Vertex<IntWritable, IntWritable, NullWritable> vertex,
    Iterable<IntWritable> messages) {
  boolean changed = false;
  for (IntWritable message : messages) {
    if (vertex.getValue().get() < message.get()) {
      vertex.setValue(message);
      changed = true;
    }
  }

  if (getSuperstep() == 0 || changed) {
    sendMessageToAllEdges(vertex, vertex.getValue());
  }
  vertex.voteToHalt();
}
```

https://github.com/apache/giraph/blob/trunk/giraph-examples/src/main/java/org/apache/giraph/examples/MaxComputation.java

大纲

23

- ❑ 编程方式
- ❑ 编程接口
- ❑ 编程实例
  - ✦ 最短路径
  - ✦ 最大值
  - ✦ 统计边的个数
  - ✦ K-Means

Aggregator

24

- ❑ Master
  ```java
  public static class yyy extends DefaultMasterCompute {
      registerAggregator(......, IntSumAggregator.class)
      //注册aggregator，并处理
  }
  ```
- ❑ Worker
  ```java
  public class xxx extends BasicComputation
      < ID类型, value类型, weight类型, message类型> {

      public void compute(
          Vertex<ID类型, value类型, weight类型> vertex,
          Iterable<message类型> messages) {
          ......
          aggregate(......)  //向master汇报
      }
  }
  ```
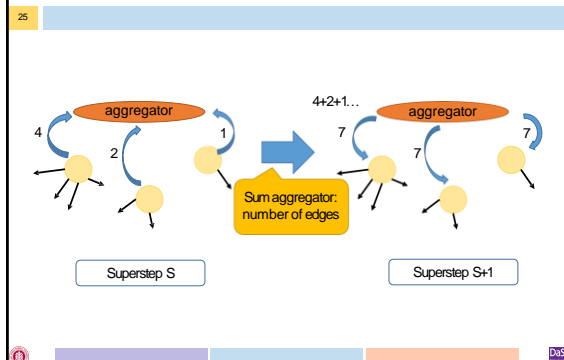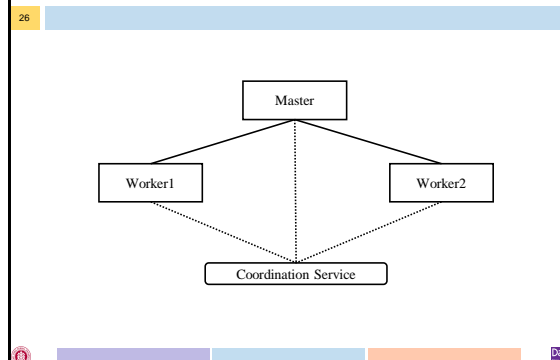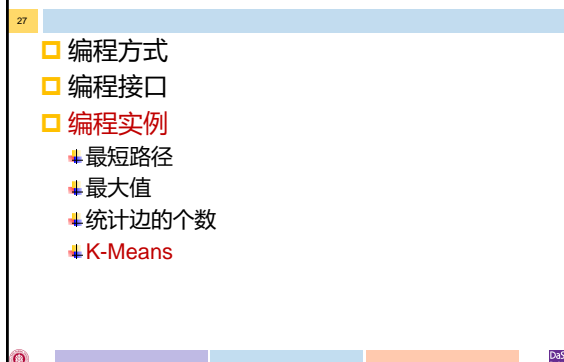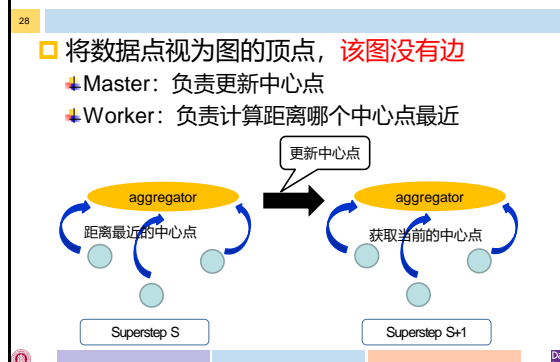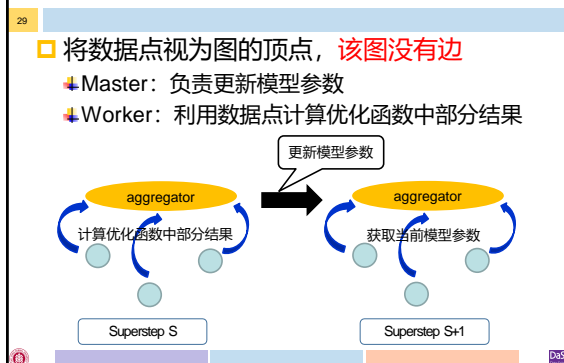
图中的边数

25

Pregel抽象架构图

26

大纲

27

- 编程方式
- 编程接口
- 编程实例
  - 最短路径
  - 最大值
  - 统计边的个数
  - K-Means

K均值

28

- 将数据点视为图的顶点，该图没有边
  - Master：负责更新中心点
  - Worker：负责计算距离哪个中心点最近

推广：机器学习算法

29

- 将数据点视为图的顶点，该图没有边
  - Master：负责更新模型参数
  - Worker：利用数据点计算优化函数中部分结果

推广：线性回归算法

30

Gradient descent algorithm

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)}$$

update θ_0 and θ_1 simultaneously

}
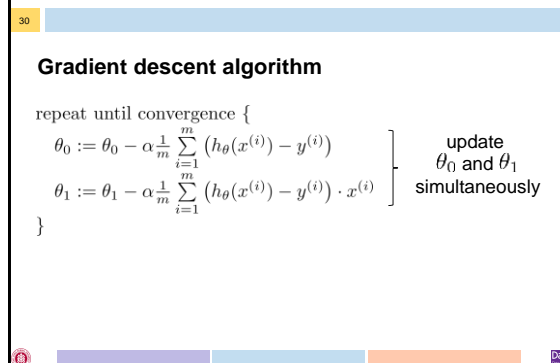
本讲小结

31

- □ 编程方式
- □ 编程接口
- □ 编程实例

谢谢！ Q&A