

## 实验 4：MapReduce 高级编程技术

陈德丹 221220159

### 任务 1：社交网络中的互相关注好友

○ **基本思路**：map 阶段对于输入的每个用户的关注列表，各生成一个用户对，小号在前，大号在后，例如 3-5，这样可以避免重复，同时打上标记，小号关注大号则标记 1，大号关注小号则标记 2，reduce 阶段对于每个用户对，如果两种标记都存在，则说明互相关注并输出

○ **mapper** 输出的 key 的类型为 Text，value 为 IntWritable

```
private Text userPair = new Text();
private final static IntWritable DIRECTION = new IntWritable();
public void map(Object key, Text value, Context context)
    throws IOException, InterruptedException {
    // 解析输入行，格式如 "user:follow1 follow2 ..."
    String[] parts = value.toString().split(":");
    if (parts.length != 2) return;
    String userA = parts[0].trim();
    int a = Integer.parseInt(userA); //将输入的用户A转换为整数

    String[] following = parts[1].trim().split(regex:"\\s+");
    for (String userB : following) { // 遍历用户A的关注列表
        if (userB.isEmpty()) continue;
        int b = Integer.parseInt(userB);

        // 确保小号在前，大号在后
        int min = Math.min(a, b);
        int max = Math.max(a, b);
        String pair = min + "-" + max;
        userPair.set(pair);

        // 判断关注方向（标记1或2）
        if (a == min) {
            DIRECTION.set(1); // 小号关注大号
        } else {
            DIRECTION.set(2); // 大号关注小号
        }
    }
    context.write(userPair, DIRECTION);
}
```

○ reducer 输出的 key 的类型为 Text，value 为 NullWritable

```
// Reducer: 检查是否存在双向标记
public static class MutualFollowReducer
    extends Reducer<Text, IntWritable, Text, NullWritable> {

    public void reduce(Text key, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException {
        boolean hasForward = false;
        boolean hasReverse = false;

        for (IntWritable val : values) {
            if (val.get() == 1) hasForward = true;
            else if (val.get() == 2) hasReverse = true;

            // 提前终止循环优化性能
            if (hasForward && hasReverse) break;
        }

        // 输出互相关注的用户对
        if (hasForward && hasReverse) {
            context.write(key, NullWritable.get());
        }
    }
}
```

○ 平台输出结果（路径为/user/221220159stu/output-Exp4/1）



○ 平台运行过程

application_1745632591680_0559	221220159stu	MutualFollow	MAPREDUCE	2025class03	0	Wed May 7 06:46:25 +0800 2025	Wed May 7 06:46:26 +0800 2025	Wed May 7 06:46:43 +0800 2025	FINISHED	SUCCEEDED	N/A	N/A
<div><div>首页 作业运行</div><div>查看日志</div><div><p>WARNING: YARN_CONF_DIR has been replaced by HADOOP_CONF_DIR. Using value of YARN_CONF_DIR.</p><p>2025-05-06 18:46:24.342 INFO client.RMProxy: Connecting to ResourceManager at master02/192.168.100.200:8032</p><p>2025-05-06 18:46:24.953 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.</p><p>2025-05-06 18:46:24.968 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/221220159stu/staging/job_1745632591680_0559</p><p>2025-05-06 18:46:25.070 INFO mapreduce.JobResourceUploader: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false</p><p>2025-05-06 18:46:25.253 INFO input.FileInputFormat: Total input files to process : 110</p><p>2025-05-06 18:46:25.301 INFO mapreduce.JobResourceUploader: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false</p><p>2025-05-06 18:46:25.342 INFO mapreduce.JobResourceUploader: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false</p><p>2025-05-06 18:46:25.348 INFO mapreduce.JobSubmitter: number of splits:110</p><p>2025-05-06 18:46:25.453 INFO mapreduce.JobSubmitter: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false</p><p>2025-05-06 18:46:25.493 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1745632591680_0559</p><p>2025-05-06 18:46:25.493 INFO mapreduce.JobSubmitter: Executing with tokens: []</p><p>2025-05-06 18:46:25.828 INFO mapreduce.JobSubmitter: found resource resource-types.xml at file:/home/workspace/hadoop-3.2.1/etc/hadoop/resource-types.xml</p><p>2025-05-06 18:46:25.852 INFO mapreduce.JobSubmitter: Adding resource type - name = yarn.io/gpu, units = , type = COUNTABLE</p><p>2025-05-06 18:46:25.974 INFO impl.YarnClientImpl: Submitted application application_1745632591680_0559</p><p>2025-05-06 18:46:26.034 INFO mapreduce.Job: The url to track the job: http://114.212.130.202:8088/track/application_1745632591680_0559/</p><p>2025-05-06 18:46:26.035 INFO mapreduce.Job: Running job: job_1745632591680_0559</p><p>2025-05-06 18:46:32.128 INFO mapreduce.Job: Job job_1745632591680_0559 running in uber mode : false</p><p>2025-05-06 18:46:32.131 INFO mapreduce.Job: map 0% reduce 0%</p><p>2025-05-06 18:46:32.166 INFO mapreduce.Job: map 5% reduce 0%</p><p>2025-05-06 18:46:39.225 INFO mapreduce.Job: map 15% reduce 0%</p><p>2025-05-06 18:46:40.234 INFO mapreduce.Job: map 42% reduce 0%</p><p>2025-05-06 18:46:41.347 INFO mapreduce.Job: map 65% reduce 0%</p></div><div>返回</div></div> <div>共 3 条 10 条/页</div>												

## 任务 2：社交网络寻找共同关注

○ **基本思路**：根据任务 1 的输出结果，首先建立 每个用户 与 参与的所有互相关注对的映射，map 阶段对于输入的用户，把它参与的互相关注对分别作为键，它的关注列表作为值输出，这样在 reduce 阶段，对于一个互相关注用户对（如 1-2），能接收到 1 的关注列表和 2 的关注列表，从而计算它们的共同关注并输出

○ **输入**依次为：①.原始关注数据（user:follow1 follow2...）②任务 1 生成的互相关注用户对（如 1-2）③x

**输出**：

共同关注数  $\leq x$  的结果到 lessX 文件（如 less15）

共同关注数  $> x$  的结果到 greaterX 文件（如 greater15）

○ **Mapper** 需要加载互相关注用户对数据，并为每个用户对关联其关注列表

首先在 setup 阶段，将每个用户相关联的所有互相关注用户对存到一个 Map 中，例如用户 1 对应 1-2, 1-3

```
// 存储格式：<用户ID, 该用户参与的所有互相关注用户对>
// 例如用户1参与的用户对：["1-2", "1-3"]
private Map<String, List<String>> mutualPairs = new HashMap<>();
private BufferedReader fis;

@Override
protected void setup(Context context) throws IOException {
    // 从分布式缓存读取任务1生成的互相关注用户对
    Configuration conf = context.getConfiguration();
    URI[] patternsURIs = Job.getInstance(conf).getCacheFiles();
    for (URI patternsURI : patternsURIs) {
        Path patternsPath = new Path(patternsURI.getPath());
        String fileName = patternsPath.getName().toString();
        fis = new BufferedReader(new FileReader(fileName));
        String line = null;
        while ((line = fis.readLine()) != null) {
            String pair = line.trim(); // 用户对格式如"1-2"
            String[] users = pair.split(regex:"-");
            if (users.length != 2) continue; // 跳过格式错误的数据

            // 将用户对关联到两个用户上，便于后续快速查找
            // 示例：用户对"1-2"会添加到用户1和用户2的关联列表中
            mutualPairs.computeIfAbsent(users[0], k -> new ArrayList<>()).add(pair);
            mutualPairs.computeIfAbsent(users[1], k -> new ArrayList<>()).add(pair);
        }
    }
}
```

map 阶段，读入每个用户的关注列表，遍历它参与的所有互相关注用户对，将关注列表作为值传给 reducer（map 输出的键和值类型均为 Text）

```
public void map(Object key, Text value, Context context)
    throws IOException, InterruptedException {
    // 输入格式示例: "1:2 3 4"
    String[] parts = value.toString().split(":");
    if (parts.length != 2) return; // 跳过格式错误的数据

    String currentUser = parts[0].trim(); // 当前用户ID
    String followsList = parts[1].trim(); // 该用户的关注列表

    // 查找当前用户参与的所有互相关注用户对
    List<String> pairs = mutualPairs.get(currentUser);
    if (pairs != null) {
        for (String pair : pairs) {
            /* 输出格式:
             * Key   : 用户对 (如"1-2")
             * Value : "当前用户:关注列表" (如"1:2 3 4")
             * 目的: 将同一用户对的两个用户的关注列表发送到同一个Reducer
             */
            context.write(new Text(pair), value);
        }
    }
}
```

○ **Reducer** 的 setup 阶段，首先根据传入的 x，动态生成输出文件名

```
public static class CommonFollowersReducer
    extends Reducer<Text, Text, Text, Text> {

    private MultipleOutputs<Text, Text> mos; // 多路输出控制器
    private int x; // 阈值参数
    private String lessOutputName, greaterOutputName; // 动态输出文件名

    @Override
    protected void setup(Context context) {
        mos = new MultipleOutputs<>(context);
        // 从作业配置中读取参数x (默认值10)
        x = context.getConfiguration().getInt("x", 10);
        // 动态生成输出文件名 (例如x=15 → "less15"和"greater15")
        lessOutputName = "less" + x;
        greaterOutputName = "greater" + x;
    }
}
```

reduce 函数中，分别存储用户对中两个用户的关注列表，通过集合的 retainAll 方法筛选出共同关注，并根据共同关注数量选择输出路径（输出的键和值类型均为 Text）

```
public void reduce(Text key, Iterable<Text> values, Context context)
    throws IOException, InterruptedException {
    // 用户对格式如"1-2"
    String[] users = key.toString().split("-");
    String userA = users[0]; // 用户A (小号)
    String userB = users[1]; // 用户B (大号)

    // 存储两个用户的关注列表
    Set<String> followsA = new HashSet<>();
    Set<String> followsB = new HashSet<>();

    // 解析来自Mapper的值（格式："用户ID:关注列表"）
    for (Text val : values) {
        String[] parts = val.toString().split(":");
        if (parts.length != 2) continue; // 跳过格式错误的数据

        String userId = parts[0]; // 用户ID
        String[] follows = parts[1].trim().split(regex:"\\s+"); // 分割关注列表

        // 将关注列表按用户归属分类
        if (userId.equals(userA)) {
            followsA.addAll(Arrays.asList(follows));
        } else if (userId.equals(userB)) {
            followsB.addAll(Arrays.asList(follows));
        }
    }
}
```

```
// 计算交集：共同关注
Set<String> common = new HashSet<>(followsA);
common.retainAll(followsB);
if (common.isEmpty()) return; // 无共同关注则跳过

System.out.println(common.size());
System.out.println(common);

// 将共同关注的交集转化为String
String outputValue = String.join(delimiter:" ", common);

// 构造输出键（原用户对追加冒号，如"1-2:"）
Text outputKey = new Text(key.toString() + ":");

// 根据共同关注数量选择输出路径
if (common.size() <= x) {
    mos.write(lessOutputName, outputKey, new Text(outputValue));
} else {
    mos.write(greaterOutputName, outputKey, new Text(outputValue));
}
}
```



○ 平台输出结果（路径为/user/221220159stu/output-Exp4/2）（x=15）

文件 - /user/221220159stu/output-Exp4/...

Page 1 of 6186

1-189: 45 8 21 140 85

1-238: 88 45 48 15 161 21 140 105 84

10-12: 167 25 205

10-162: 167 25 236

10-167: 12 25 205 162

10-192: 236 205

10-205: 12 167 25 236 192

10-236: 25 205 192 162

10-25: 12 167 236 205 162

1000-1008: 1092 996 857

1000-1092: 1008 996 857

1000-1093: 984 952 996

10000-10030: 10218

10000-10056: 10318 10235

10000-10090: 10282 9999 10235

取消

下载

文件 - /user/221220159stu/output-Exp4/...

Page 1 of 70914

1-100: 88 47 48 231 155 156 158 90 91 51 54 12 58 15 160 161 165 2 123 167 201 204 6 128 207 60 62 20 21 65

170 134 179 215 139 218 75 79 39 182 184 140 221 146 147 148 105 227 229 108 109 84 85

1-105: 88 45 47 48 231 155 156 238 90 91 54 12 15 161 165 166 2 123 167 201 204 6 127 128 8 207 60 62 20 21

65 170 134 137 215 139 218 35 79 39 184 140 221 100 146 147 148 227 229 108 109 84 41 85

1-108: 88 45 48 231 155 156 158 90 91 51 54 58 15 160 161 165 2 167 201 204 6 127 128 207 60 62 20 21 65 170

179 137 215 139 218 79 39 182 184 140 221 100 146 105 229 109 84 85

1-109: 88 45 47 48 231 155 156 158 90 91 51 54 58 15 160 161 166 2 167 201 6 127 128 8 207 60 21 170 134 179

137 215 139 218 79 39 182 184 140 100 146 148 105 227 229 108 84 85

1-12: 88 45 47 48 231 155 156 158 90 91 51 58 15 160 161 165 166 2 123 167 201 204 6 127 128 8 60 62 20 65

28 170 134 137 215 218 75 35 79 39 182 184 140 221 100 146 147 148 105 227 84 85

1-123: 88 45 47 48 231 155 158 90 91 51 12 58 15 160 161 165 166 167 201 204 127 128 8 60

取消

下载

○ 平台运行过程

<a href="#">application_1745632591680_0663</a>	221220159stu	CommonFollowers	MAPREDUCE	2025class03	0	Thu May 8 04:24:13 +0800 2025	Thu May 8 04:24:13 +0800 2025	Thu May 8 04:24:59 +0800 2025	FINISHED	SUCCEEDED	N/A	N/A	N
--	--------------	-----------------	-----------	-------------	---	-------------------------------	-------------------------------	-------------------------------	----------	-----------	-----	-----	---

查看日志

WARNING: YARN\_CONF\_DIR has been replaced by HADOOP\_CONF\_DIR. Using value of YARN\_CONF\_DIR.

2025-05-07 16:24:11.431 INFO client.RMProxy: Connecting to ResourceManager at master02/192.168.100.200:8032

2025-05-07 16:24:12.048 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with h ToolRunner to remedy this.

2025-05-07 16:24:12.063 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/221220159stu/staging/job\_1745632591680\_0663

2025-05-07 16:24:12.162 INFO util.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false

2025-05-07 16:24:12.351 INFO InputFileInputFormat: Total input files to process = 110

2025-05-07 16:24:12.427 INFO util.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false

2025-05-07 16:24:12.475 INFO util.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false

2025-05-07 16:24:12.484 INFO mapreduce.JobSubmitter: number of splits=110

2025-05-07 16:24:12.694 INFO util.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false

2025-05-07 16:24:12.728 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1745632591680\_0663

2025-05-07 16:24:12.728 INFO mapreduce.JobSubmitter: Setting up token: {}

2025-05-07 16:24:13.061 INFO conf.Configuration: found resource resource-types.xml at file:/home/workspace/hadoop-3.2.1/etc/hadoop/resource-types.xml

2025-05-07 16:24:13.180 INFO impl.YarnClientImpl: Submitted application application\_1745632591680\_0663

2025-05-07 16:24:13.252 INFO mapreduce.Job: The url to track the job: http://114.212.130.202:8088/proxy/application\_1745632591680\_0663/

2025-05-07 16:24:13.313 INFO mapreduce.Job: Running job: job\_1745632591680\_0663

2025-05-07 16:24:13.344 INFO mapreduce.Job: job: job\_1745632591680\_0663 running in uber mode : false

2025-05-07 16:24:19.346 INFO mapreduce.Job: map 0% reduce 0%

2025-05-07 16:24:26.437 INFO mapreduce.Job: map 2% reduce 0%

2025-05-07 16:24:27.646 INFO mapreduce.Job: map 32% reduce 0%

2025-05-07 16:24:28.455 INFO mapreduce.Job: map 59% reduce 0%

10790 102 107 162.54.70 424 10479 1000000000 1000 1000 1000 1000

返回

共4条 10条/页

1

继续 1 页

### 任务 3：好友推荐

○ **基本思路**：观察示例可知，如果用户 B 和用户 A 互关，用户 A 和用户 C 互关，但 B 没有关注 C，则向 B 推荐 C。因此可以根据任务 1 的输出结果建立每个用户互相关注的用户集合，对于每个用户，遍历它互相关注的互相关注，如果它没有关注则推荐，再将计数前 5 的结果输出即可。

○ **Mapper** 的 setup 阶段，从互相关注用户对中建立每个用户互相关注的用户集合

```
// 建立互相关注列表 存储格式：<用户ID, 与该用户互相关注的所有用户>
// 例如用户1对应 ["2", "3"]
private Map<String, Set<String>> mutualFollowers = new HashMap<>();
private BufferedReader fis;

@Override
protected void setup(Context context) throws IOException {
    // 从分布式缓存读取任务1生成的互相关注用户对
    Configuration conf = context.getConfiguration();
    URI[] patternsURIs = Job.getInstance(conf).getCacheFiles();
    for (URI patternsURI : patternsURIs) {
        Path patternsPath = new Path(patternsURI.getPath());
        String fileName = patternsPath.getName().toString();
        fis = new BufferedReader(new FileReader(fileName));
        String line = null;
        while ((line = fis.readLine()) != null) {
            String pair = line.trim(); // 用户对格式如"1-2"
            String[] users = pair.split(regex:"-");
            if (users.length != 2) continue; // 跳过格式错误的数据

            // 将用户1添加到用户0的互相关注列表中，将用户0添加到用户1的互相关注列表中
            mutualFollowers.computeIfAbsent(users[0], k -> new HashSet<>()).add(users[1]);
            mutualFollowers.computeIfAbsent(users[1], k -> new HashSet<>()).add(users[0]);
        }
    }
}
```

map 函数，首先存储输入用户的关注列表，再遍历它互相关注的互相关注，如果没有关注则输出（map 的 key 和 value 的类型均为 Text）

```
public void map(Object key, Text value, Context context)
    throws IOException, InterruptedException {
    // 输入格式: user:follow1 follow2...
    String[] parts = value.toString().split(":");
    String userB = parts[0].trim();

    // 将用户B的关注列表转换为集合
    Set<String> userB_following = new HashSet<>({
        Arrays.asList(parts[1].trim().split(regex: " "))
    });
    // 获取用户B的互相关注列表
    Set<String> userB_mutual_follow = mutualFollowers.get(userB);
    if (userB_mutual_follow == null) return; // 如果没有互相关注的用户，跳过
    // 遍历用户B的所有互相关注用户
    for (String userA : userB_mutual_follow) {
        // 对于与用户B互相关注的用户A，获取其互相关注列表
        Set<String> userA_mutual_follow = mutualFollowers.get(userA);
        for (String userC : userA_mutual_follow) {
            if (userB_following.contains(userC)) continue;
            if (userB.equals(userC)) continue; // 跳过自己

            // 此时A和B、A和C都互相关注，但B不关注C
            // 生成推荐关系：给B推荐C
            // Key: 被推荐人(userB), Value: 候选用户
            context.write(new Text(userB), new Text(userC));
        }
    }
}
```

○ **Reducer** 对于每个用户的候选被推荐用户，用一个 Map 记录推荐次数，排序后输出推荐次数前 5 的用户（键和值类型均为 Text）

```
public static class RecommendationReducer
    extends Reducer<Text, Text, Text, Text> {

    public void reduce(Text key, Iterable<Text> values, Context context)
        throws IOException, InterruptedException {
        // 存储候选用户的推荐次数
        Map<String, Integer> recommendations = new HashMap<>();

        // 统计推荐次数
        for (Text val : values) {
            String candidate = val.toString();
            recommendations.put(candidate, recommendations.getOrDefault(candidate, defaultValue:0) + 1);
        }

        // 构造前5推荐结果
        // 排序并取前 5 个 key，拼接成字符串
        String top5 = recommendations.entrySet().stream()
            .sorted((e1, e2) -> e2.getValue().compareTo(e1.getValue())) // 按 value 倒序
            .limit(maxSize:5)
            .map(Map.Entry::getKey) // 只保留 key
            .collect(Collectors.joining(delimiter: " ")); // 用空格拼接

        context.write(key, new Text(top5));
    }
}
```



○ 平台输出结果（路径为/user/221220159stu/output-Exp4/3）

文件 - /user/221220159stu/output-Exp4/...

Page 1 of 834

120576959731

101404515320215

100137451277695

10008751083951976877

100009985997399821036910106

100011005110094102161001810176

1000299681020410106998610131

100031022510369102481017210295

10004100731018510392102729983

10005103159973103241014610204

10006103221008599921018110133

100071021899739979998510143

100081012810030994899739968

10009102671037399861018210369

1001101496494710469970

取消

下载

○ 平台运行过程

application_1745632591680_0647	221220159stu	FriendRecommendation	MAPREDUCE	2025class03	0	Thu May 8 03:10:11 +0800 2025	Thu May 8 03:10:11 +0800 2025	Thu May 8 03:10:52 +0800 2025	FINISHED	SUCCEEDED	N/A	N/A	N/A
--------------------------------	--------------	----------------------	-----------	-------------	---	-------------------------------	-------------------------------	-------------------------------	----------	-----------	-----	-----	-----

查看日志

WARNING: YARN\_CONF\_DIR has been replaced by HADOOP\_CONF\_DIR. Using value of YARN\_CONF\_DIR.

2025-05-07 15:10:09,694 INFO clientRMProxy: Connecting to ResourceManager at master02/192.168.100.200:8032

2025-05-07 15:10:10,444 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with 'hadoop' to remedy this.

2025-05-07 15:10:10,460 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/221220159stu/staging/job\_1745632591680\_0647

2025-05-07 15:10:10,563 INFO safeSaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

2025-05-07 15:10:10,760 INFO inputFileInputFormat: Total input files to process: 110

2025-05-07 15:10:10,810 INFO safeSaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

2025-05-07 15:10:10,851 INFO safeSaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

2025-05-07 15:10:10,860 INFO mapreduce.JobSubmitter: number of splits:110

2025-05-07 15:10:11,069 INFO safeSaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

2025-05-07 15:10:11,104 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1745632591680\_0647

2025-05-07 15:10:11,104 INFO mapreduce.JobSubmitter: Executing with tokens: []

2025-05-07 15:10:11,461 INFO conf.Configuration: found resource resource-types.xml at file:/home/workspace/hadoop-3.2.1/etc/hadoop/resource-types.xml

2025-05-07 15:10:11,484 INFO resource.ResourceUtils: Adding resource type - name = yarn.io/gpu, units = , type = COUNTABLE

2025-05-07 15:10:11,601 INFO impl.YarnClientImpl: Submitted application application\_1745632591680\_0647

2025-05-07 15:10:11,870 INFO mapreduce.Job: The url to track the job: http://114.212.130.202:8086/Proxy/application\_1745632591680\_0647/

2025-05-07 15:10:11,871 INFO mapreduce.Job: Running job: job\_1745632591680\_0647

2025-05-07 15:10:17,763 INFO mapreduce.Job: Job job\_1745632591680\_0647 running in uber mode : false

2025-05-07 15:10:17,765 INFO mapreduce.Job: map 0% reduce 0%

2025-05-07 15:10:23,952 INFO mapreduce.Job: map 4% reduce 0%

2025-05-07 15:10:24,978 INFO mapreduce.Job: map 7% reduce 0%

2025-05-07 15:10:26,026 INFO mapreduce.Job: map 21% reduce 0%

10106.98.07.15:10:27.033 INFO mapreduce.Job: map: 62% reduce: 0%

返回