

# Software Systems and Distributed Systems SS 2017

Shen Gao  
Bibek Paudel  
May 11, 2017

# InvertedIndex: Mapper

```
StringTokenizer tokenizer = new StringTokenizer(line);  
Map<String, Integer> wordCounts = new HashMap<String, Integer>();  
int movieId = Integer.parseInt(tokenizer.nextToken());
```

```
while (tokenizer.hasMoreTokens()) {  
    String word = tokenizer.nextToken().toLowerCase();  
    if(stopWords.contains(word))  
        continue;  
    if(wordCounts.containsKey(word)){  
        int count = wordCounts.get(word);  
        wordCounts.put(word, count+1);  
    }  
    else wordCounts.put(word, 1);  
}
```

```
for(Map.Entry<String, Integer> entry : wordCounts.entrySet()){  
    context.write(new Text(entry.getKey()), new  
Text(movieId+" "+entry.getValue()));  
}
```

# InvertedIndex: Reducer

```
StringBuilder str = new StringBuilder();  
  
for (Text val : values) {  
    str.append(val.toString()+" ");  
}  
  
str.setLength(str.length()-1);  
context.write(key, new Text(str.toString()));
```

# Join: Setup

```
MultipleInputs.addInputPath(job, new Path(args[0]  
    + "data/movie.metadata.tsv"), TextInputFormat.class,  
MovieMapper.class);
```

```
MultipleInputs.addInputPath(job, new Path(args[0]  
    + "data/character.metadata.tsv"), TextInputFormat.class,  
CharacterMapper.class);
```

# Join: Mapper

```
public static class CharacterMapper extends Mapper<Object, Text,
Text, Text> {
    private Text movieID = new Text();
    private Text actorName = new Text();
    private String actorIdentifier = new String("a");

    public void map(Object key, Text value, Context context)
        throws IOException, InterruptedException {
        String line = new String(value.toString());
        String[] fields = new String[13];
        fields = line.split("\t");
        movieID.set(fields[0]);
        actorName.set(actorIdentifier + fields[8]);
        context.write(movieID, actorName);
    }
}
```

# Join: Mapper

```
public static class MovieMapper extends Mapper<Object, Text, Text,
Text> {
    private Text movieID = new Text();
    private Text movieName = new Text();
    private String movieIdentifier = new String("m");

    public void map(Object key, Text value, Context context)
        throws IOException, InterruptedException {
        String line = new String(value.toString());
        String[] fields = new String[9];
        fields = line.split("\t");
        movieID.set(fields[0]);
        movieName.set(movieIdentifier + fields[2]);
        context.write(movieID, movieName);
    }
}
```

# Join: Reducer

```
actors = "";  
name = "";  
for (Text val : values) {  
    if (val.toString().charAt(0) == 'm') {  
        name = val.toString().substring(1);  
    } else if (val.toString().charAt(0) == 'a') {  
        if (actors.isEmpty())  
            actors += "," + val.toString().substring(1);  
        else  
            actors += "," + val.toString().substring(1);  
    }  
}  
nameAndActors.set(name + actors);  
context.write(key, nameAndActors);
```