Probeklausur

Teil 1: Multiple-Choice-Fragen

1. Was ist der Hauptzweck von MapReduce?
   - a) Um eine benutzerfreundliche grafische Oberfläche zu bieten.
   - b) Um große Datenmengen parallel zu verarbeiten.
   - c) Um kleine Datensätze auf einem einzelnen Computer zu verarbeiten.
   - d) Um Netzwerkprobleme in einem Cluster zu lösen.

2. Welcher der folgenden Schritte gehört nicht zu einem MapReduce-Job?
   - a) Map
   - b) Reduce
   - c) Shuffle and Sort
   - d) Deploy

3. Was passiert im Shuffle and Sort Schritt eines MapReduce-Jobs?
   - a) Die Daten werden von der Festplatte gelöscht.
   - b) Die Daten werden zwischen verschiedenen Rechnern ausgetauscht und sortiert.
   - c) Die Daten werden komprimiert.
   - d) Die Daten werden in eine Datenbank geschrieben.

4. Welches der folgenden Tools ist eine bekannte Implementierung von MapReduce?
   - a) Apache Spark
   - b) Apache Hadoop
   - c) MySQL
   - d) PostgreSQL

Teil 2: Lückentexte

1. Beim MapReduce-Programmierungsmodell wird der _____ Schritt verwendet, um Eingabedaten in Schlüssel/Wert-Paare zu transformieren, während der _____ Schritt verwendet wird, um die Zwischenpaare zu aggregieren und das Endergebnis zu erzeugen.

   Antwort: _____, _____

2. Das Hadoop Distributed File System (HDFS) ist ein verteiltes Dateisystem, das _____ Datenmengen speichert und so _____ und _____ gewährleistet.

   Antwort: _____, _____, _____

3. Die _____ API ermöglicht es Entwicklern, MapReduce-Jobs in verschiedenen Programmiersprachen zu schreiben, die Standard-Input und Standard-Output unterstützen.

   Antwort: _____

Teil 3: Code-Beispiele ergänzen

1. Vervollständigen Sie das Mapper-Skript (mapper.py):

```
python
  #!/usr/bin/env python3
  import sys

  # Lese die Eingabedaten zeilenweise
  for line in sys.stdin:
      # Entferne führende und nachgestellte Leerzeichen
      line = line.strip()
      # Zerlege die Zeile in _______
      words = line._______
      # Gib jedes Wort mit dem Wert 1 aus
      for word in words:
          print(f'{word}\t1')
```

Antwort:

```
  python
  #!/usr/bin/env python3
  import sys

  # Lese die Eingabedaten zeilenweise
  for line in sys.stdin:
      # Entferne führende und nachgestellte Leerzeichen
      line = line.strip()
      # Zerlege die Zeile in Wörter
      words = line.split()
      # Gib jedes Wort mit dem Wert 1 aus
      for word in words:
          print(f'{word}\t1')
```

2. Vervollständigen Sie das Reducer-Skript (reducer.py):

```
python
  #!/usr/bin/env python3
  import sys

  current_word = None
  current_count = 0
  word = None

  # Lese die Eingabedaten zeilenweise
  for line in sys.stdin:
      # Entferne führende und nachgestellte Leerzeichen
      line = line.strip()
      # Zerlege die Zeile in Wort und Zähler
      word, count = line.split('\t', 1)
      try:
          count = int(count)
      except ValueError:
```

```
            continue

        if current_word == word:
            current_count += count
        else:
            if current_word:
                print(f'{current_word}\t{_______}')
            current_count = count
            current_word = word

    if current_word == word:
        print(f'{current_word}\t{current_count}')
```

Antwort:

```
python
  #!/usr/bin/env python3
  import sys

  current_word = None
  current_count = 0
  word = None

  # Lese die Eingabedaten zeilenweise
  for line in sys.stdin:
      # Entferne führende und nachgestellte Leerzeichen
      line = line.strip()
      # Zerlege die Zeile in Wort und Zähler
      word, count = line.split('\t', 1)
      try:
          count = int(count)
      except ValueError:
          continue

      if current_word == word:
          current_count += count
      else:
          if current_word:
              print(f'{current_word}\t{current_count}')
          current_count = count
          current_word = word

  if current_word == word:
      print(f'{current_word}\t{current_count}')
```

3. Ergänzen Sie das Hadoop Job-Konfigurationsbeispiel:

```
java
  Configuration conf = new Configuration();
  Job job = Job.getInstance(conf, "word count");
  job.setJarByClass(WordCount.class);
  job.setMapperClass(TokenizerMapper.class);
  job.setCombinerClass(IntSumReducer.class);
```

```
  job.setReducerClass(IntSumReducer.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(IntWritable.class);
  FileInputFormat.addInputPath(job, new Path(args[0]));
  FileOutputFormat.setOutputPath(job, new Path(args[1]));
  System.exit(job._____________ ? 0 : 1);
```

Antwort:

```
java
  Configuration conf = new Configuration();
  Job job = Job.getInstance(conf, "word count");
  job.setJarByClass(WordCount.class);
  job.setMapperClass(TokenizerMapper.class);
  job.setCombinerClass(IntSumReducer.class);
  job.setReducerClass(IntSumReducer.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(IntWritable.class);
  FileInputFormat.addInputPath(job, new Path(args[0]));
  FileOutputFormat.setOutputPath(job, new Path(args[1]));
  System.exit(job.waitForCompletion(true) ? 0 : 1);
```