Übung 17: Cluster

1. Verbinden Sie sich mittels `ssh cluster` mit dem Cluster
2. Schauen Sie sich an, welche Partitionen und Nodes es gibt. Sind Nodes momentan nicht verfügbar?
3. Zählen Sie, wie viele Jobs momentan auf der Partition `gpulong` laufen.
4. Allozieren Sie mit `salloc` Ressourcen für einen Job auf der main-Partition. Geben Sie die folgenden Optionen an:
   - Der Task soll maximal zehn Minuten laufen
   - Wir wollen nur eine CPU pro Task verwenden
   - Wir wollen mindestens auf zwei Nodes laufen
   - Insgesamt wollen wir vier Prozessoren verwenden
5. Rufen Sie `sacct` auf und schauen Sie sich ihren Job an. Welche ID hat er?
6. Führen Sie mit `srun` das Kommando `hostname` auf den allozierten Nodes aus. Welche Ausgabe erhalten Sie? Warum?
7. Führen Sie das Kommando ohne `srun` aus. Was ist anders?
8. Beenden Sie ihren Job durch `scancel`
9. Verwandeln Sie die oben ausgeführten Aufrufe in ein Batch-Script und führen Sie es aus. Übertragen Sie alle Optionen, die Sie zuvor bei `salloc` verwendet haben. Fügen Sie außerdem Anweisungen hinzu, damit Sie über den Fortschritt des Jobs per Mail informiert werden. Welche Nachrichten bekommen Sie?
10. Wir wollen nun einige Dateien auf dem Cluster verarbeiten. In der Datei `/home/public/vorkurs_ws18/nltk-tagger` finden sie einen simplen Tagger, den wir auf dem Cluster ausführen wollen. Kopieren Sie ihn auf das Cluster.
11. Neben dem Tagger brauchen wir auch Daten. Diese haben die Form `kant.txt-split0x` und finden sich im selben Verzeichnis. Die Daten sind Teilstücke eines größere Korpus und wurden aufgeteilt, damit wir sie parallel verarbeiten können. Kopieren Sie sie auch auf das Cluster.
12. Erstellen Sie eine Python 2 virtualenv auf dem Cluster und aktivieren Sie sie.
13. Installieren Sie die benötigten NLTK-Werkzeuge, indem Sie in Python folgende Befehle ausführen:

```
import nltk
nltk.download('averaged_perceptron_tagger')
nltk.download('punkt')
```

14. Verwenden Sie ein Job Array, um Jobs zu erstellen, die den Tagger jeweils einmal auf jede Datei anwenden. Der Aufruf für das Programm lautet dabei:

```
python nltk-tagger.py <input-file>
```

15. Der Tagger hat für jede Eingabedatei eine Ausgabedatei erstellt. Kopieren Sie sie alle zurück in ihr (Pool-)Home-Verzeichnis und fügen Sie sie zusammen.