# Sorting mit Hadoop

# Sorting mit Hadoop

- Inhalt / Lern-Ziele
  - Sortierung mit Hadoop
    - Partial Sort
    - Total Sort
  - Sampling

# Partial Sort

- Hadoop sortiert in der Suffle&Sort Phase nach den Keys K2
  - => Output jedes Reducers ist sortiert nach K2 (nicht K3!)
  - Sortier-Ordnung über `RawComparator`
    - `mapred.output.key.comparator.class` setzen z.B. via `job.setSortComparatorClass()`
    - Keys Unterklasse von `WritableComparable`
- nur Sortierung innerhalb eines part-r-xxxxx Output-Files

# Total Sort

- naive: Single Partition (1 Reducer)
  - nicht parallel, d.h. nicht für große Datenmengen
- mittels Partitioner, der die Sortier-Ordnung respektiert
  - Merge der Outputs part-r-xxxxx ergibt eine sortierte Datei
- Gleiche Arbeitsaufteilung der Reducer mittels *Sampling* des Key-Space!

# Sampling

- Hadoop bietet etliche Sampler, z.B. `RandomSampler`, `IntervallSampler`, `SplitSampler`
- Erzeugen eines Partition File *_partitions* und Verteilen mittels *Distributed Cache*
- *Vorsicht: Partitionierung des Key-Space darf nicht feinteiliger als die Anzahl der Reducer sein*

# Beispiel

- siehe Verzeichnis
  - src/examples/org/apache/hadoop/examples/Sort.java

# Literatur


- Tom White, „Hadoop The Definite Guide“, third edition, 2012, O'Reilly