

## Aufgabenblatt 7 Web-Scale Data Management

- Abgabetermin: -
- Zur Prüfungszulassung muss ein Aufgabenblatt mit mind. 25% der Punkte bewertet werden und alle weiteren Aufgabenblätter mit mindestens 50% der Punkte.
- Die Aufgaben sollen in Zweiergruppen bearbeitet werden.
- Abgabesystem unter  
<http://www.dcl.hpi.uni-potsdam.de/submit>
  - ausschließlich pdf-Dateien
  - eine Datei pro Aufgabe namens Aufgabe-<aufgabenNr>.pdf
  - jedes Blatt beschriftet mit Namen

### Aufgabe 1: MapReduce

Ein optisches Meßverfahren erfasst die Oberfläche von Objekten als Menge von 3D-Ortsvektoren  $(x, y, z)^T$  und speichert diese in der Datenbanktabelle  $S(\text{vectorId}, \text{dimension}, \text{value})$  ab. Durch Störeinflüsse werden beim Messen nicht alle Ortsvektoren vollständig erfasst.

Für eine grafische Simulation werden nun die Vektorlängen  $length$  aller **vollständig erfasster** Ortsvektoren benötigt,  $length = \sqrt{x^2 + y^2 + z^2}$ . Aufgrund der großen Datenmenge soll die Berechnung auf einem MapReduce-Cluster erfolgen.

- a) Erzeuge als Ausgabe eine Tabelle  $T(\text{vectorId}, \text{length})$ , in der für jeden vollständig erfassten Ortsvektor die Vektorlänge steht. Die Ausgabe wird partitioniert auf den verschiedenen Nodes ins verteilten Dateisystem geschrieben. Löse die Aufgabe mit nur einem MapReduce-Job. Verwende auch die in der Übung vorgestellte Funktion *combine*, um Netzwerklast zu reduzieren. Beschreibe dein Vorgehen kurz in wenigen Sätzen. Erstelle den Pseudocode für *map*, *combine* und *reduce*. **8 P**
- b) Die Abbildung zeigt beispielhaft eine Verteilung von  $S$  auf drei MapReduce-Nodes. Zeige für dieses Beispiel die Ausgabe der einzelnen Phasen (*map*, *combine* und *reduce*) deines MapReduce-Jobs auf jeder der drei Nodes. **3 P**

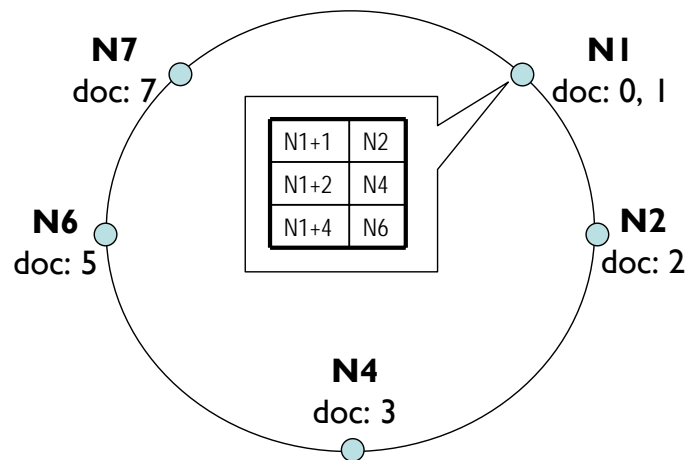
<i>id</i>	<i>dim</i>	<i>val</i>
1	x	2
2	z	4
1	y	3
3	y	2
4	y	4

<i>id</i>	<i>dim</i>	<i>val</i>
3	z	4
4	z	2
2	x	2
2	y	4

<i>id</i>	<i>dim</i>	<i>val</i>
5	y	3
3	x	4
4	x	4

### Aufgabe 2: Distributed Hash Tables (Bonusaufgabe)

Gegeben sei folgendes Peer-to-Peer-System zur verteilten Speicherung von Dateien. So speichert beispielsweise Peer  $N_4$  das Dokument mit der eindeutigen Dokumenten-ID 3. Für die Suche und das Routing wird eine Chord DHT mit 3-Bit Schlüsseln verwendet. **6 Bonuspunkte**

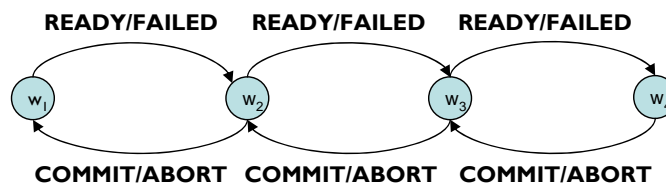


- Erstelle die *FingerTables* für die Peers N2, N4, N6 und N7.
- An Peer N2 wird eine Lookup-Anfrage nach *Dokument 0* gestellt. An welche Peers wird die Anfrage weitergeleitet?
- Angenommen, Peer N4 fällt aus. An Peer N7 wird danach eine Lookup-Anfrage nach *Dokument 5* gestellt. An welche Peers wird die Anfrage weitergeleitet, unter der Annahme, dass die *FingerTables* der Peers bereits aktualisiert wurden?

### Aufgabe 3: Two Phase Commit (Bonusaufgabe)

In der Vorlesung wurde eine hierarchische Organisationsstruktur (ein Koordinator und mehrere untergeordnete Worker) beim 2PC-Protokoll beschrieben. Es ist auch möglich, die in der Abbildung gezeigte lineare Organisationsstruktur vorzunehmen.

Hierbei ist kein ausgezeichnete Koordinator erforderlich. In der ersten Phase reichen die Worker ihren eigenen Status und den der linken Nachbarn von *links nach rechts* weiter, nachdem sie einen entsprechenden Statusbericht von links bekommen haben. Der letzte Worker in der Reihe – hier Worker  $w_4$  – trifft die Entscheidung und reicht sie nach links weiter.



Entwickle das 2PC-Protokoll – analog zur Vorlesung – für diese lineare Anordnung der Worker als Pseudocode und beschreibe kurz das Vorgehen. **4 Bonuspunkte**