

# MapReduce und Datenbanken

## Thema 15: Strom bzw. Onlineverarbeitung mit MapReduce

Jan Kristof Nidzwetzki

# Übersicht

- 1** Begriffe
- 2** Verschiedene Arbeiten
- 3** MapReduce Online
  - Ziele
  - Hadoop Online Prototype
- 4** DEDUCE: at the intersection of MapReduce and stream processing
  - Beispiel
- 5** Beyond online aggregation: parallel and incremental data mining with online Map-Reduce
- 6** Zusammenfassung

# Begriffe

## Definition (Stromverarbeitung)

Unter dem Begriff der Stromverarbeitung versteht man das fortlaufende Verarbeiten von Datenströmen.

- Bsp. Börsenticker
- Continuous-Queries

## Definition (Onlineverarbeitung)

Unter dem Begriff der Onlineverarbeitung versteht man, dass die Verarbeitung der Daten im Dialog mit dem Benutzer erfolgt.

- Keine Stapelverarbeitung
- Bsp. Vorzeitige Ergebnisse bei lang laufenden Berechnungen

# Verschiedene Arbeiten

- MapReduce Online (**HOP - Hadoop Online Prototype**)
- DEDUCE: at the intersection of MapReduce and stream processing (**Erweiterung IBM System S**)
- Beyond online aggregation: parallel and incremental data mining with online Map-Reduce (**Eigenes Framework**)

# Map Reduce Online

## Hadoop Online Prototype - HOP

### Hadoop Online Prototype

- Erweiterung von Hadoop
- MapReduce in der Strom- bzw. Onlineverarbeitung
  - Online-Aggregation
  - Continuous-Queries
- Erhöhung der Ausführungsgeschwindigkeit normaler MapReduce-Jobs

# Pipelining

## Idee

- Sofortiges weiterleiten der Ausgaben des Map-Task an den Reduce-Task
- **Pipelining** ermöglicht Überlappung von Map- und Reduce-Task

## Schwierigkeiten

- Kopieren der Daten zwischen Map- und Reduce-Task benötigt viel Zeit
- Nicht genug TCP-Sockets verfügbar
- **Combiner-Funktion** schwierig anzuwenden

# Pipelining

## Fehlertoleranz

### Fehlschlag eines Map-Task

- Verwerfen der Ergebnisse und Neustart des Map-Task
- Einsatz von Checkpoints

### Fehlschlag eines Reduce-Task

- Neustart des betreffenden Reduce-Task

# Online Aggregation

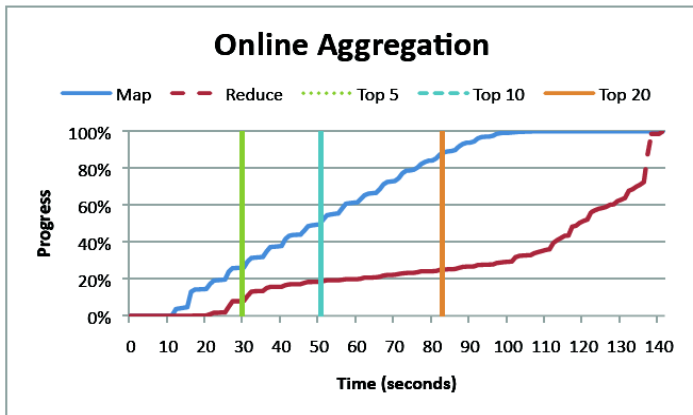
Online Aggregation ermöglicht es, vorab Ergebnisse eines Jobs „**Snapshots**“ zu erhalten.

## Snapshots ...

- ... enthalten alle Daten des Map-Tasks zu einem bestimmten Zeitpunkt
- ... sind auch zwischen mehreren Jobs möglich
- Genauigkeit ist nur schwer zu bestimmen
- Reduce-Phase muss mit jedem Snapshot erneut durchgeführt werden



# Online Aggregation - Beispiel



Häufigste K Wörter in 5.5 GB Text der Wikipedia.  
(Quelle: MapReduce Online [CCA+09] S. 7)

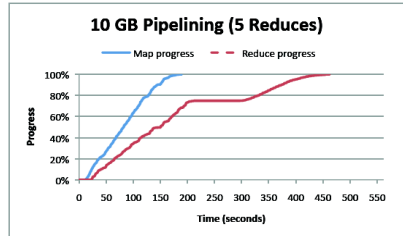
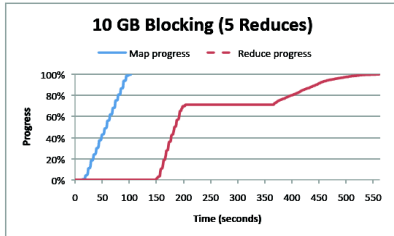
# Continuous-Queries

- **Continuous-Queries** ermöglichen es, sich verändernde Datenbestände fortlaufend auszuwerten.
- **Beispiel: Auswertung von Logdateien**
- Vorteil: Daten müssen nicht vollständig neu ausgewertet werden, wenn Änderungen vorgenommen werden.
- **Fehlertoleranz** teilweise schwierig zu Implementieren

# Beispielanwendung: Monitoring

- Cluster mit 7 Systemen
- Systeme mit höherer Last sollen ermittelt werden
- Einsatz von **Continuous-Queries**
- **Agents** (Map-Task) liefern Ihre Ergebnisse an einen **Aggregator** (Reduce-Task)
- Berechnung der Systemlast über 120 Sekunden als gleitender Durchschnitt
- Systeme die mehr als zwei Standardabweichungen abweichen, werden gemeldet

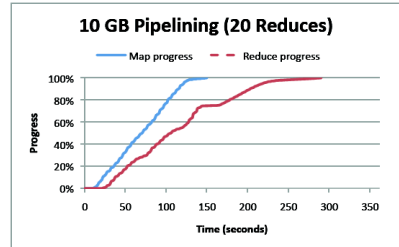
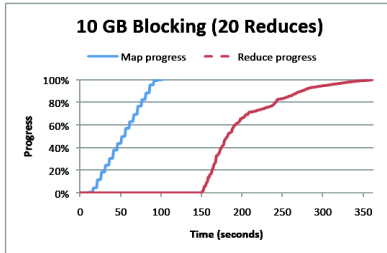
# Performance



Wordcount über 10 GB mit 20 Map-Tasks und 5 Reduce-Tasks.  
Links: Hadoop ohne Pipelining (561 Sekunden),  
Rechts: HOP mit Pipelining (462 Sekunden)

(Quelle: MapReduce Online [CCA+09] S. 11)

# Performance



Wordcount über 10 GB mit 20 Map-Tasks und 20 Reduce-Tasks.  
Links: Hadoop ohne Pipelining (361 Sekunden),  
Rechts: HOP mit Pipelining (290 Sekunden)

(Quelle: MapReduce Online [CCA+09] S. 11)

# DEDUCE: at the intersection of MapReduce and stream processing

## IBM System S / IBM InfoSphere Streams

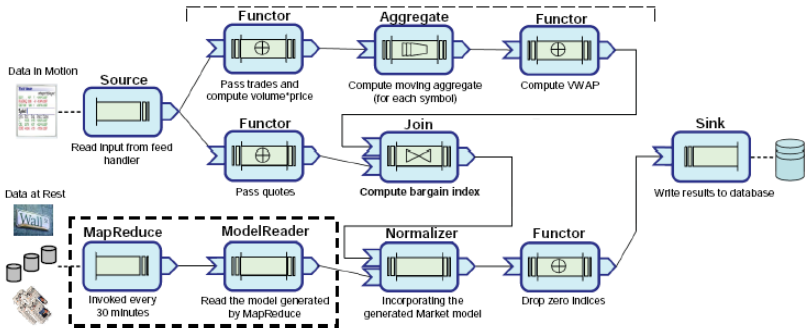
- Software zur Stromverarbeitung
- Auswertung der Datenströme wird mittels **Data-Flow Graphen** beschrieben
- Auswertung kann auf mehreren Computern parallel erfolgen
- **SPADE - Stream Processing Application Declarative Engine**  
Sprache zum beschreiben der **Data-Flow Graphen**
- Erweiterung dieser Sprache mittels **UBOPs - user-defined build-in operators**

# IBM System S / IBM InfoSphere Streams

## DEDUCE als Erweiterung des IBM System S

- MapReduce-Operator als Datenlieferant
- Beliefert die **stream processing operators** periodisch mit Daten
- Ziel ist es die Stromverarbeitung um statische Daten anzureichern
- **Beispiel: Auswertung von Aktienmärkten**
  - Statische Daten: Historische Aktienkurse
  - Datenstrom: Aktuelle Aktienkurse

# Beispielanwendung - Auswertung von Finanzmärkten



(Quelle: DEDUCE: At the intersection of MapReduce and stream processing  
 [KAGW10] S. 5)



# Beyond online aggregation: parallel and incremental data mining with online Map-Reduce

- Software zur Evaluierung von Algorithmen zur Berechnung des Job-Fortschritts und der Genauigkeit von Snapshots
- Implementation als **Shared-Memory-Architektur**
- Primär als Software zum Testen von Algorithmen gedacht
- Schlechte Fehlertoleranz und keine Skalierung der Jobs über mehrere Rechner

# Zusammenfassung

## MapReduce Online

Erweiterung von Hadoop um **Snapshots** und **Continuous-Queries**

## Beyond online aggregation: parallel and incremental data mining with online Map-Reduce

Software zur Evaluierung von Algorithmen zur Berechnung des Job-Fortschritts und der Genauigkeit von Snapshots

## DEDUCE als Erweiterung des IBM System S

Erweiterung des IBM System S um die Möglichkeit Daten periodisch mittels MapReduce auszuwerten