MAP REDUCE

Entwurfsmuster

Lars Briem

(briem.lars@googlemail.com)

Duale Hochschule Baden Württemberg - Standort Karlsruhe

Gliederung

Gliederung

Warum Entwurfsmuster?

- Vergleichbar mit Entwurfsmuster von „Gang of Four“
- Allgemeine Lösung für wiederkehrende Probleme
- Vokabular für Entwickler
- Geeignete Abstraktion zur Kommunikation

Bestandteile eines Entwurfsmusters

- Beschreibung
- Ziel / Problem
- Anwendbarkeit / Voraussetzungen
- Ergebnis
- Verwendungen

$\Rightarrow$ Nicht jedes Entwurfsmuster ist für jedes Problem geeignet

Gliederung

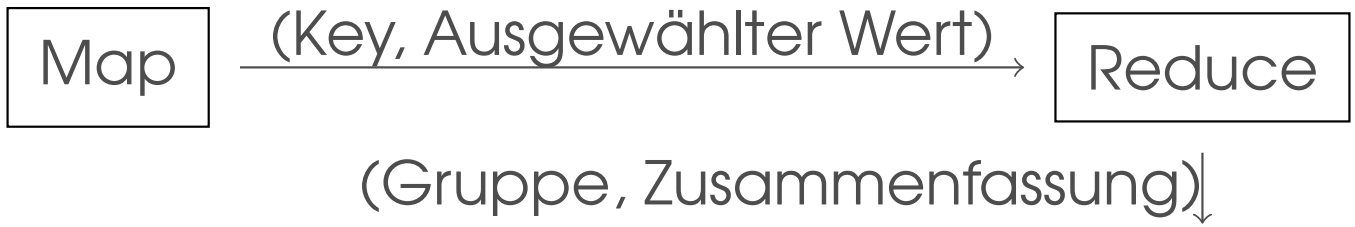
Gliederung

Summationsmuster

- Zusammenfassung der Daten
- Gruppierung der Daten
- Aussagen über einzelne Gruppen
- Beispiele
  - Numerische Summation
  - Invertierter Index

Numerische Summation

- Berechnung statistischer Merkmale einer Menge
  - Anzahl
  - Minimum / Maximum
  - Durchschnitt / Median / Standardabweichung
- Voraussetzung
  - Numerischen Daten oder Anzahl von Elementen
  - Gruppierung von Daten nach Key
- Verwendung
  - Statistik



Invertierter Index

- Zuordnung eines Begriffs zu einer Liste von Bezeichnern
  - Suchbegriff zu URL
- Verwendung
  - Suche / Suchmaschine

Invertierter Index - Beispiel

Einfügen von Stackoverflow Links in Wikipedia

Map

- Durchsuchen aller Posts nach Wikipedia Links
- Ausgabe: (Wikipedia Link, Post ID)

Reduce

- Einfügen von Stackoverflow Links auf Wikipedia Seiten

Live Demo

Gliederung

Filterungsmuster

- Keine Änderung von Daten
- Nur „Filterung“
  - Nur Teil der Daten relevant
  - Kleinerer Datensatz für Entwicklung
- Beispiele
  - Filterung
  - Top Ten
  - Eindeutig / Einzigartig (Distinct)

Filterung

- Verarbeitung einzelner Key-Value Paare
- Entscheidung ob Key-Value Paar verwendet wird oder nicht
- Map Phase ausreichend
- Verwendung
  - Überall

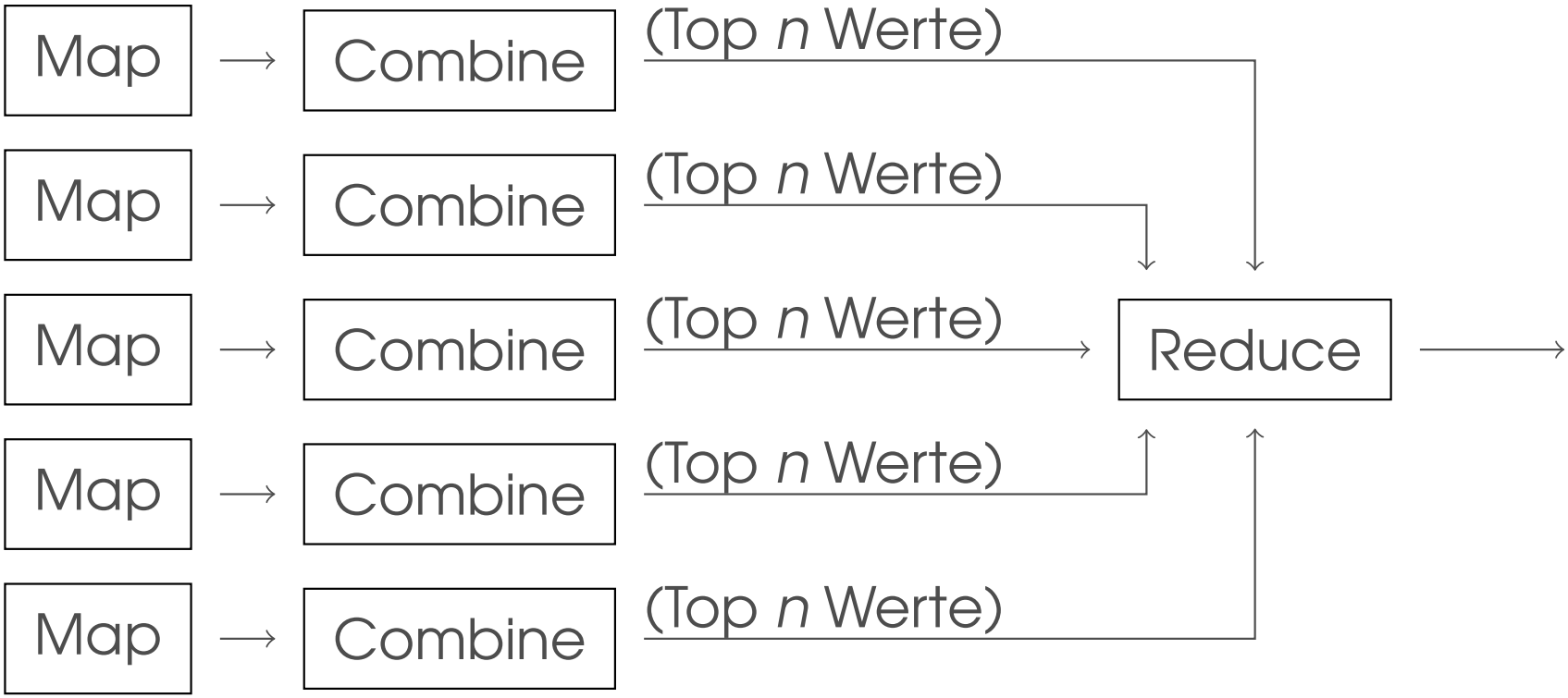
Erweiterung - Bloom Filter

- Filterkriterium zu groß
  - Liste von Personen
- Verringert den Aufwand
- Voraussetzungen
  - Liste von Elementen als Kriterium
  - Falsch Positive erlaubt

Top Ten

- Finden der besten *n* Elemente
- 1 Reducer
- Voraussetzungen
  - Vergleichbare Daten
- Verwendung
  - Suche von Ausreißern

Top Ten - Combiner



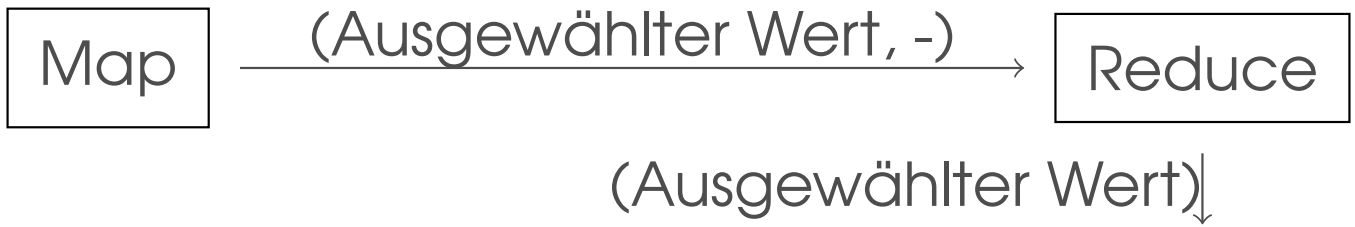
$\Rightarrow$ Combiner reduziert Netzwerklast

$\Rightarrow$ Gleicher Code in Combine und Reduce

Live Demo

Eindeutig / Einzigartig (Distinct)

- Reduzierung einer Datenmenge auf einzigartige Werte
- Entfernen von Duplikaten
- Ohne Duplikate nutzlos



Gliederung

Vereinigungsmuster (Join)

- Verknüpfung mehrerer Datenquellen
- Aufwendige Berechnung
- Viel Last im Netzwerk

Beispiele

- Reduce Side Join
- Replicated Join
- Comosite Join
- Kartesisches Produkt / Kreuzprodukt

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Inner Join

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Inner Join

Benutzer ID	Name	Ort
0	Lars	Karlsruhe
2	Linus	Freiburg

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Outer Join

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Outer Join

Benutzer ID	Name	Ort
0	Lars	Karlsruhe
1	Lena	null
2	Linus	Freiburg
3	null	Karlsruhe

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Left Outer Join

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Left Outer Join

Benutzer ID	Name	Ort
0	Lars	Karlsruhe
1	Lena	null
2	Linus	Freiburg

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Right Outer Join

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Right Outer Join

Benutzer ID	Name	Ort
0	Lars	Karlsruhe
2	Linus	Freiburg
3	null	Karlsruhe

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Anti Join

Join Typen

Benutzer ID	Name
0	Lars
1	Lena
2	Linus

Tabelle: Benutzer

Benutzer ID	Ort
0	Karlsruhe
2	Freiburg
3	Karlsruhe

Tabelle: Ort

Join: Benutzer + Ort über Benutzer ID

Anti Join

Benutzer ID	Name	Ort
1	Lena	null
3	null	Karlsruhe

Reduce Side Join

- Einfachste Art
- Viel Last auf Netzwerk
- Alle Join Typen möglich
- Verwendung
  - Beliebig viele Datenquellen
  - Beliebig große Datenquellen

Live Demo

Reduce Side Join - Erweiterung Bloom Filter

- Viel Last im Netzwerk durch Reduce Side Join
- Bloom Filter kann Last verringern
- Voraussetzung
  - Inner Join

Weitere Joins

- Replicated Join
  - Ein großer Datensatz
  - Mehrere kleine Datensätze (im Speicher)
  - Nur Map Phase
- Composite Join
  - Sortiert und Partitioniert nach Fremdschlüssel
  - Vorsortierung / -partitionierung notwendig
  - Nur Map Phase
- Kartesisches Produkt / Kreuzprodukt
  - Kombination aller Datensätze miteinander
  - Berechnungsaufwand $O(n^k)$ bei k Datensätzen

Gliederung

Daten Organisationsmuster

- Umwandlung der Datenorganisation
- Effizientere Verarbeitung

Beispiele

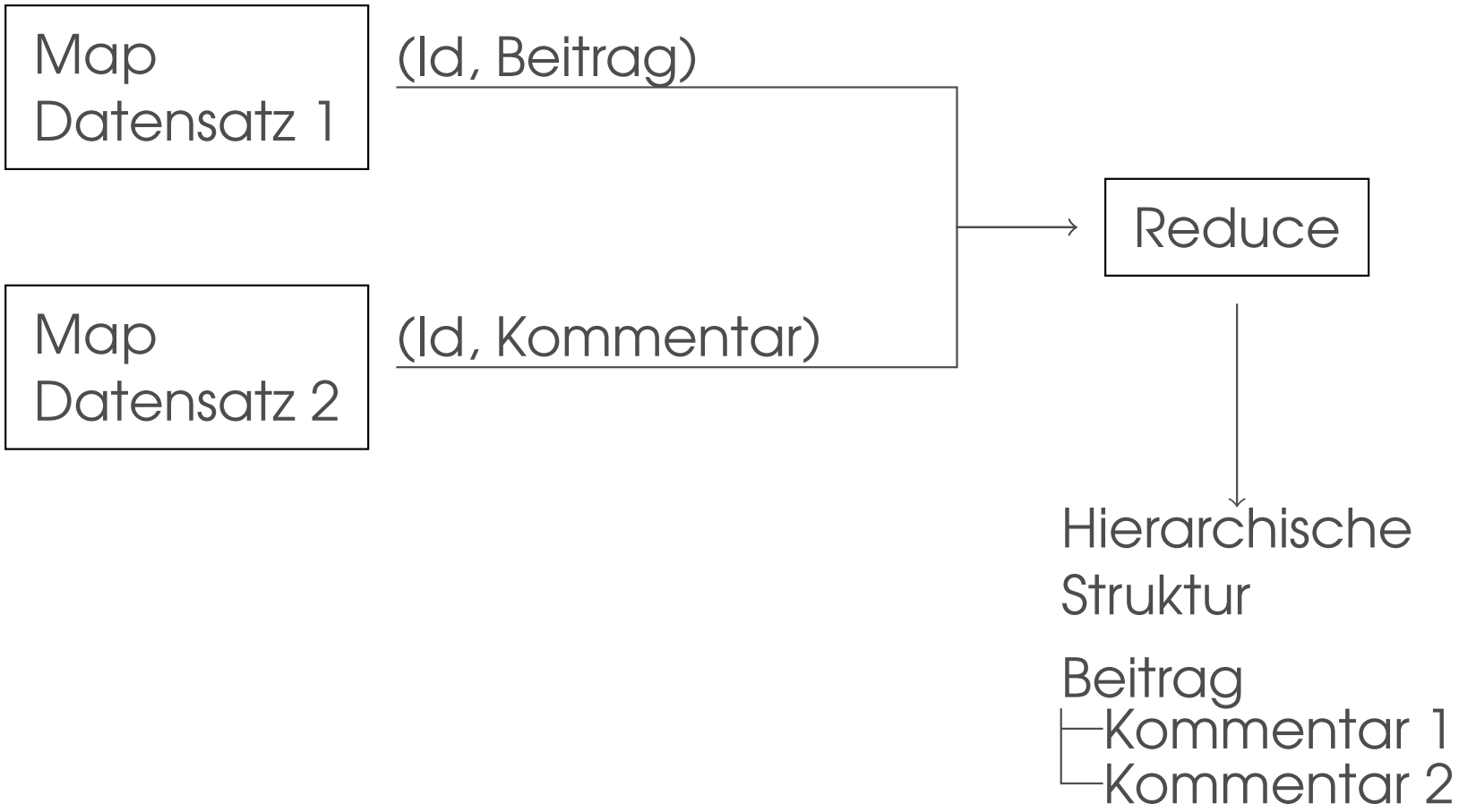
- Strukturiert zu Hierarchisch
- Partitionierung / Klasseneinteilung
- Sortierung

Strukturiert zu Hierarchisch

- Umwandlung von Daten aus RDBMS
- Vermeidung von Joins in späteren Berechnungen
- Voraussetzungen
  - Daten über Fremdschlüssel verknüpft
- Verwendung
  - Vorbereitung für NoSQL

$\Rightarrow$ Verwendet Reduce Side Join

Strukturiert zu Hierarchisch



Partitionierung / Klasseneinteilung

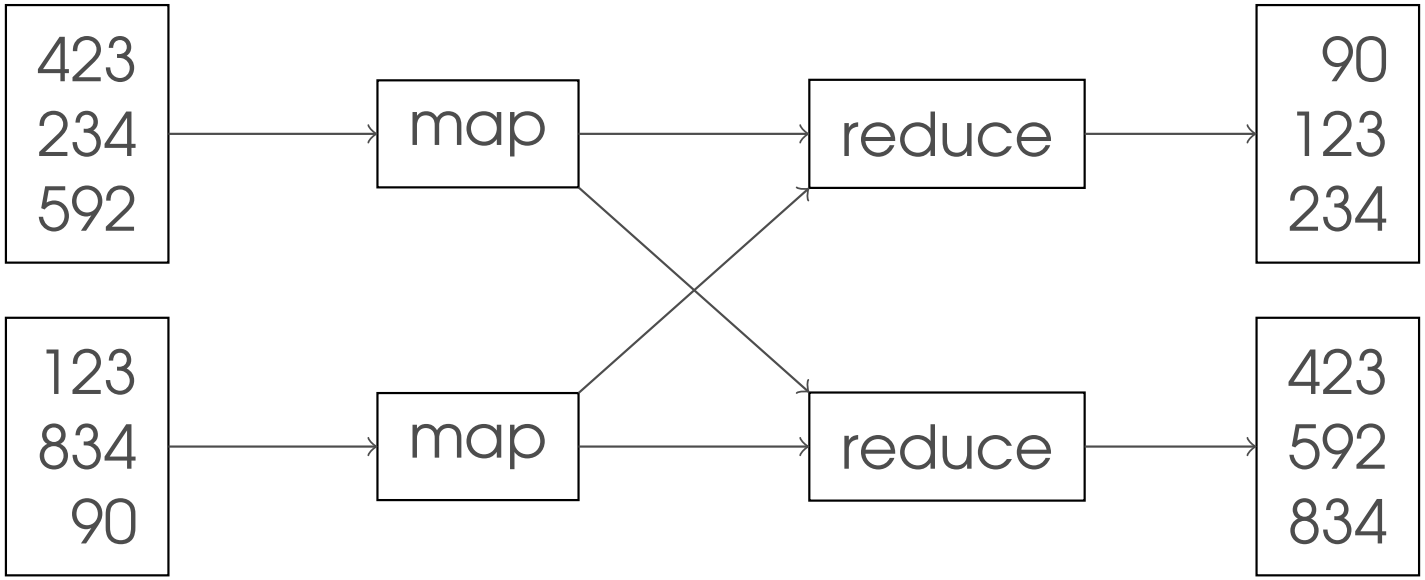
- Unterteilung der Daten
- Gruppierung von ähnlichen Daten
- Verwendet Partitioner
- Voraussetzung
  - Anzahl Partitionen bekannt
- Verwendung
  - Gruppierung von kontinuierlichen Werten (z.B. Zeit)
  - Gruppierung von Klassen (z.B. Länder)

Live Demo

Sortierung

- Abspeichern in verschiedenen Dateien
- Sortierte Liste bei aufeinanderfolgenden Dateien
- Voraussetzung
  - Sortierbare Daten
- Verwendung
  - Parallele Sortierung

Sortierung - Beispiel



Gliederung

Metamuster

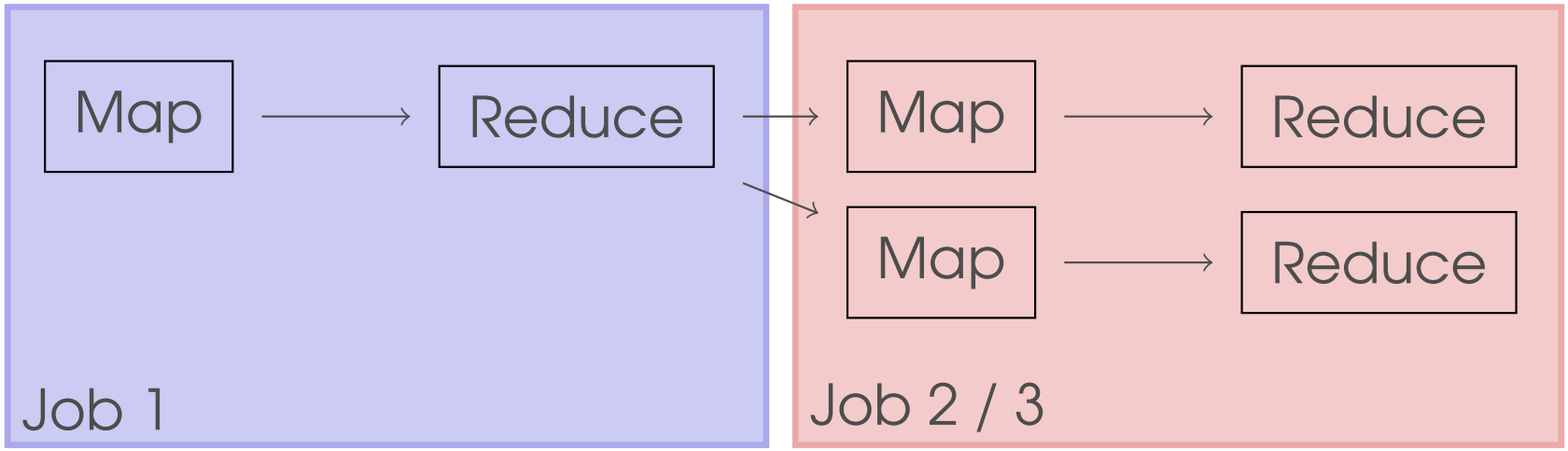
- Geschickte Kombination verschiedener Entwurfsmuster
- Steigerung der Effizienz
- Reduzierung der Rechenzeit

Beispiele

- Verkettung von Jobs
- Zusammenlegen von Jobs
- Mischen von Jobs

Verkettung von Jobs

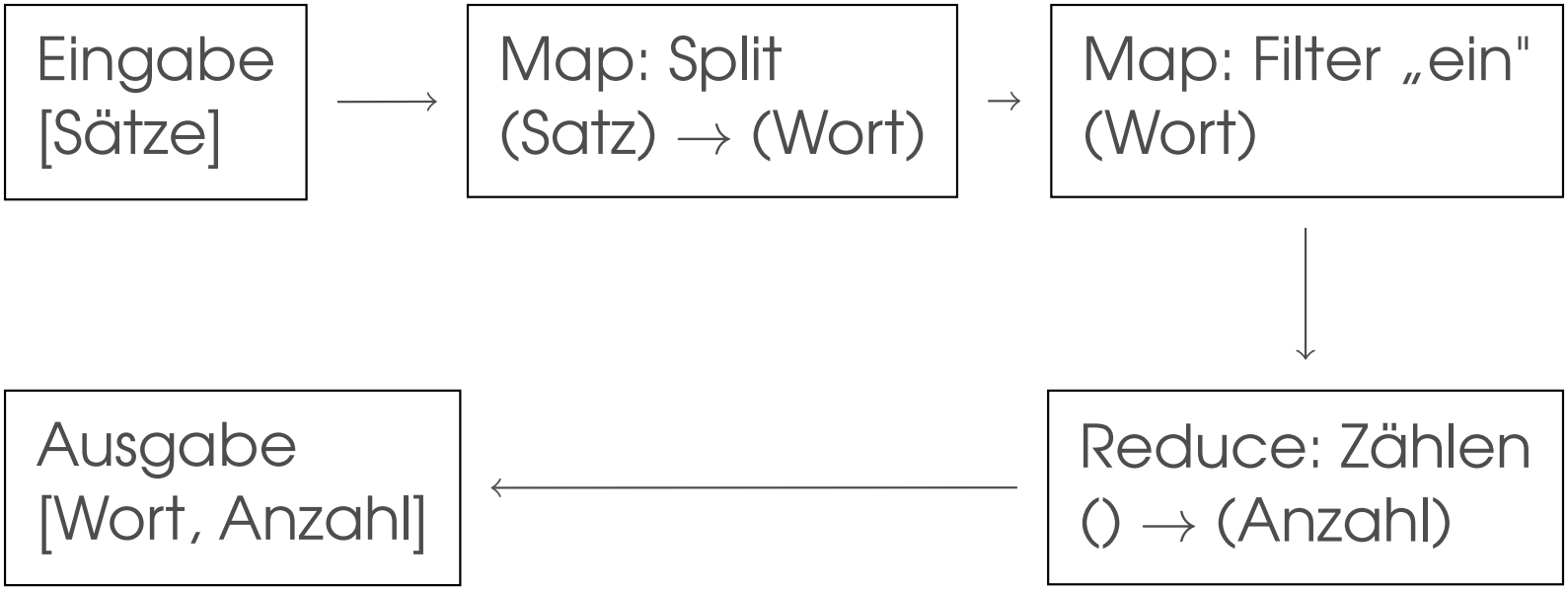
- Kombination mehrerer MapReduce Jobs
- Zwischenspeichern der Daten
- Zwischenergebnisse Löschen
- Fehlerbehandlung notwendig



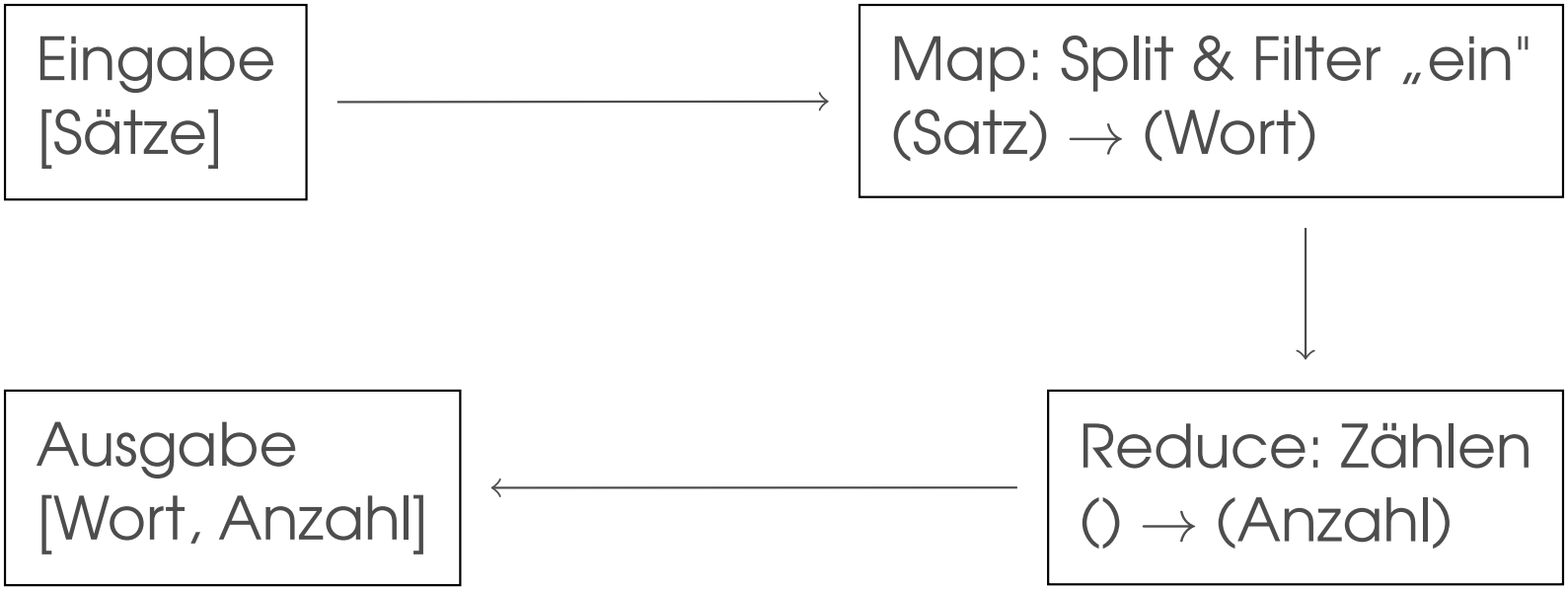
Zusammenlegen von Jobs

- Zusammenlegung von Map / Reduce Phasen
- Weniger Schreiboperationen
- Arten
  - Kombination mehrerer Map Phasen
  - Kombination einer Reduce und Map Phase

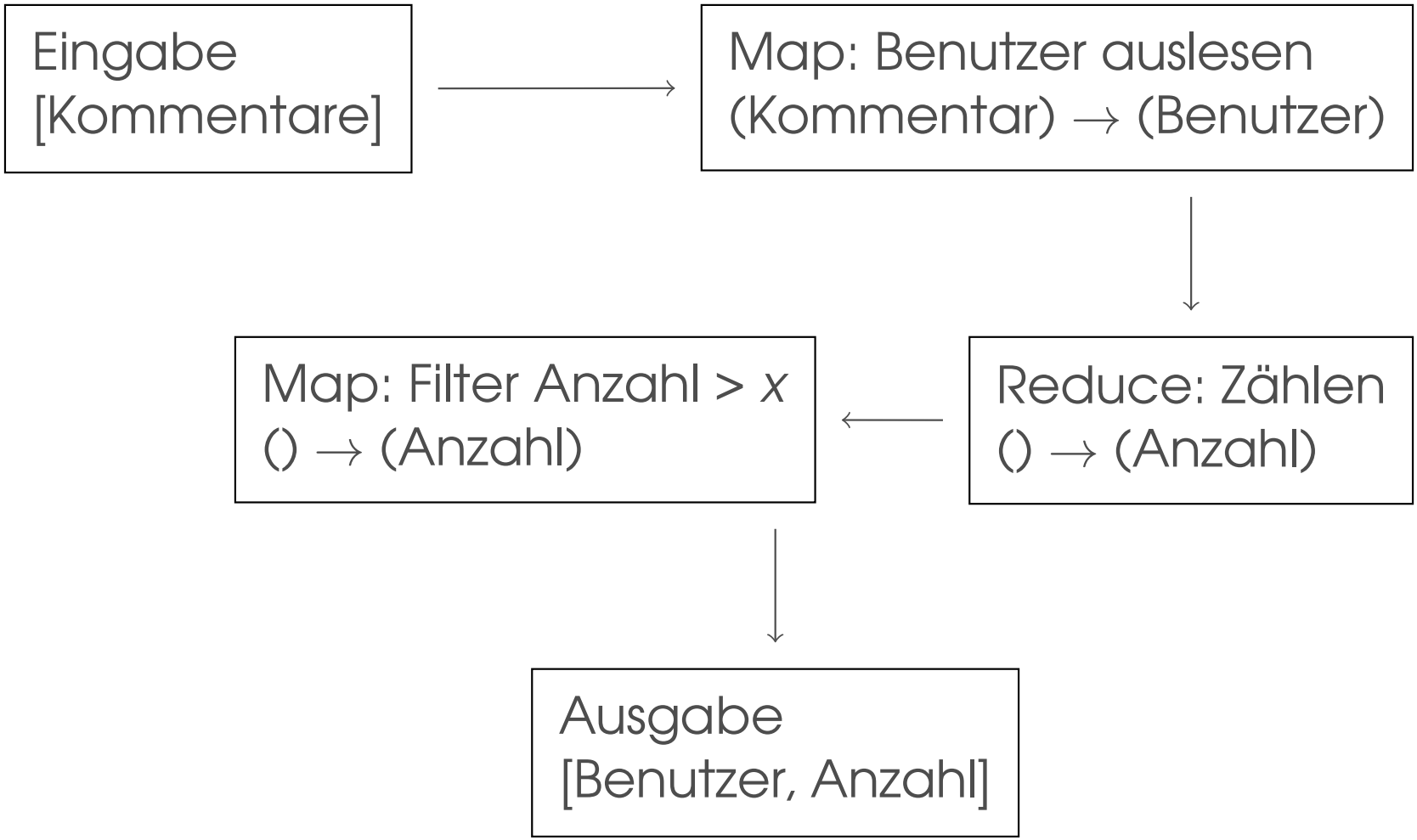
Zusammenlegen von Jobs - Beispiel: Kombination von Map Phase



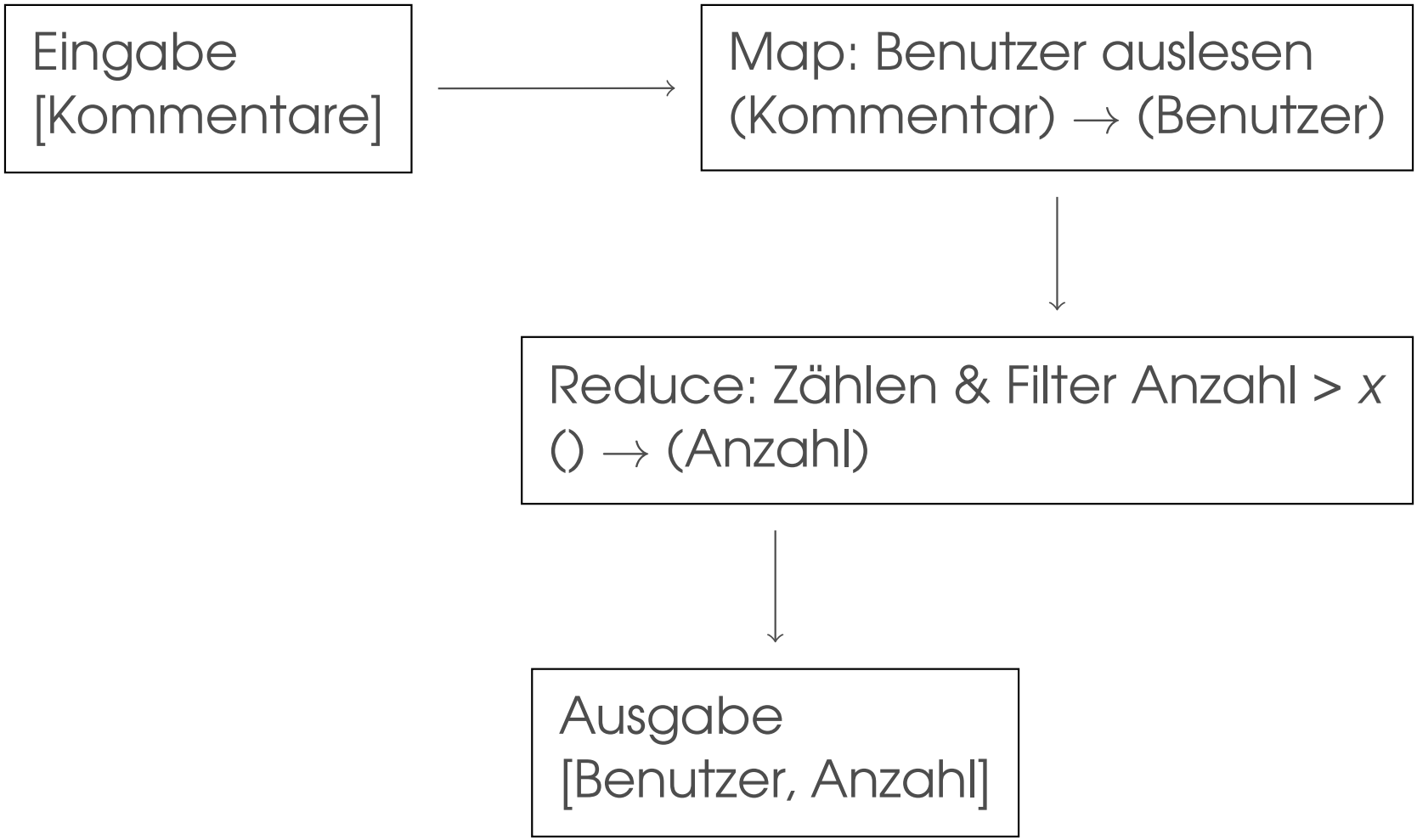
Zusammenlegen von Jobs - Beispiel: Kombination von Map Phase



Zusammenlegen von Jobs - Beispiel: Reduce & Map zu Reduce



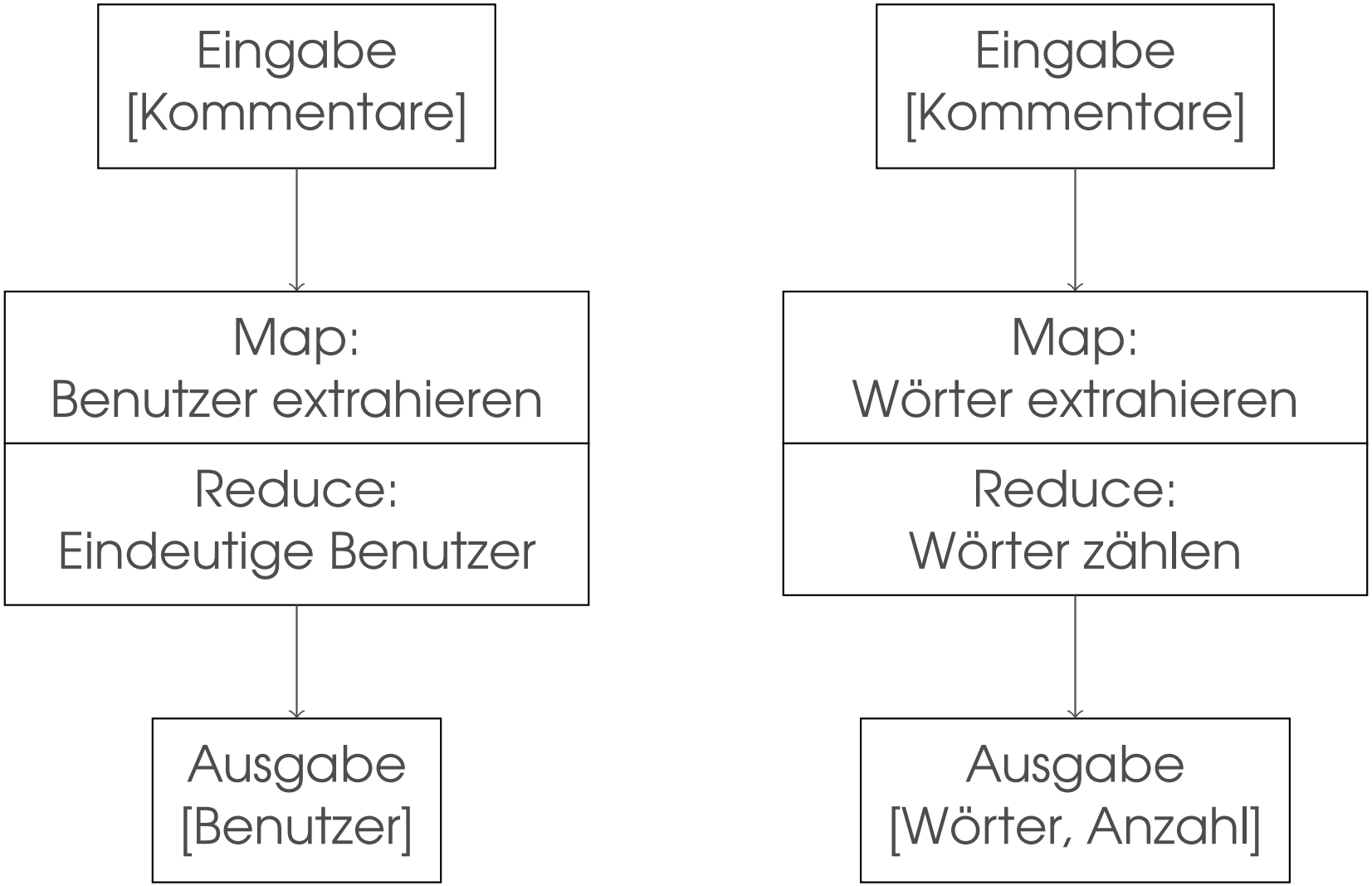
Zusammenlegen von Jobs - Beispiel: Reduce & Map zu Reduce



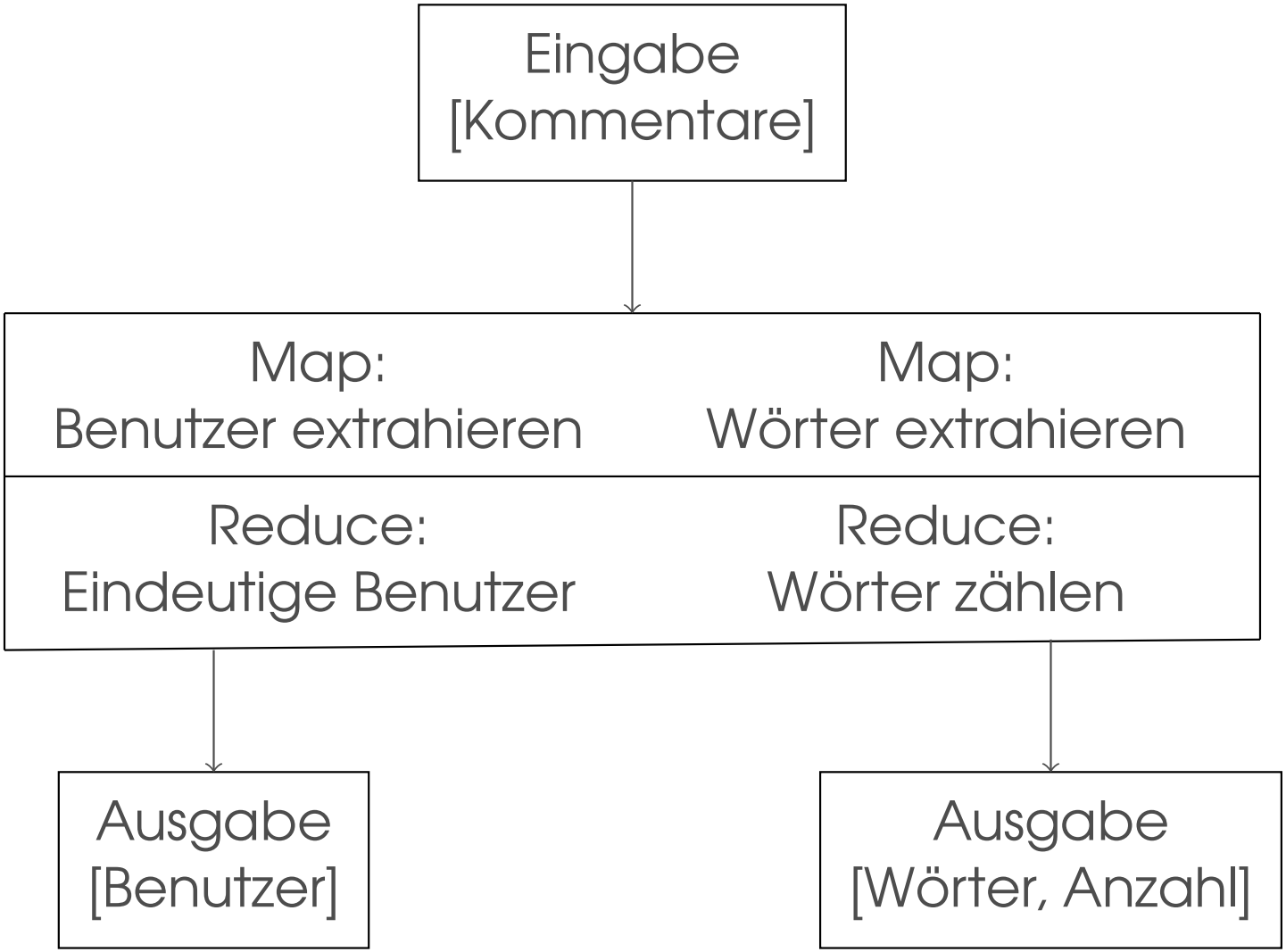
Mischen von Jobs

- Gleichzeitige Ausführung mehrerer MapReduce Jobs
  - Einmal Einlesen
  - Mehrere Mapper ausführen
- Weniger Aufwand zum Lesen / Parsen
- Ungünstig während Entwicklung
- Große Effizienzsteigerung bei langsamen Laufwerken (Bandlaufwerk, HDD)
- Reduktion von Last bei Datenbanken

Mischen von Jobs - Beispiel



Mischen von Jobs - Beispiel



Live Demo

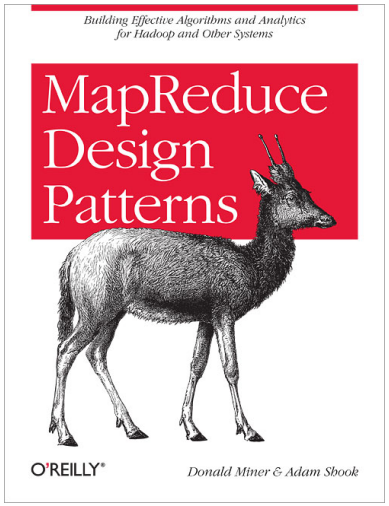
Gliederung

Ausblick

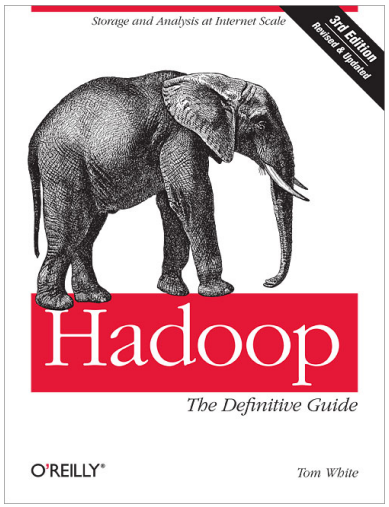
- Nur kleine Auswahl gängiger Probleme
- Eingabe / Ausgabe
  - Weitere Datentypen (Bild, Video, Audio, . . .)
  - Weitere Quellen (SQL Datenbank)
- Programmiersprachen haben MapReduce Paradigma eingebaut
  - Java Streams
  - C# Pipelines

Gliederung

Literatur

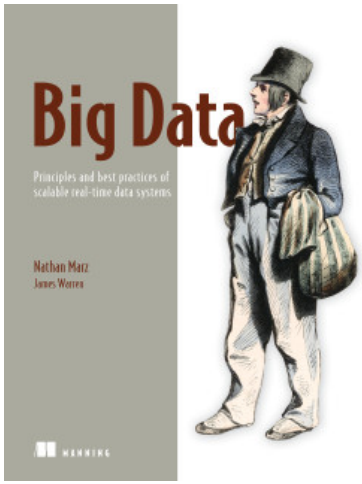


- MapReduce Design Patterns
  - Donald Miner, Adam Shook
  - O'Reilly
  - ISBN: 978-1449327170



- Hadoop: The Definitive Guide
  - Tom White
  - O'Reilly
  - ISBN: 978-1449311520

Literatur



- Big Data
  - Nathan Marz, James Warren
  - Manning
  - ISBN: 978-1617290343

Quellen

- Inhalt
  - MapReduce: simplified data processing on large clusters - Jeffrey Dean, Sanjay Ghemawat
  - manning.com
  - oreilly.com
- Daten
  - stackoverflow.com
  - dwd.de