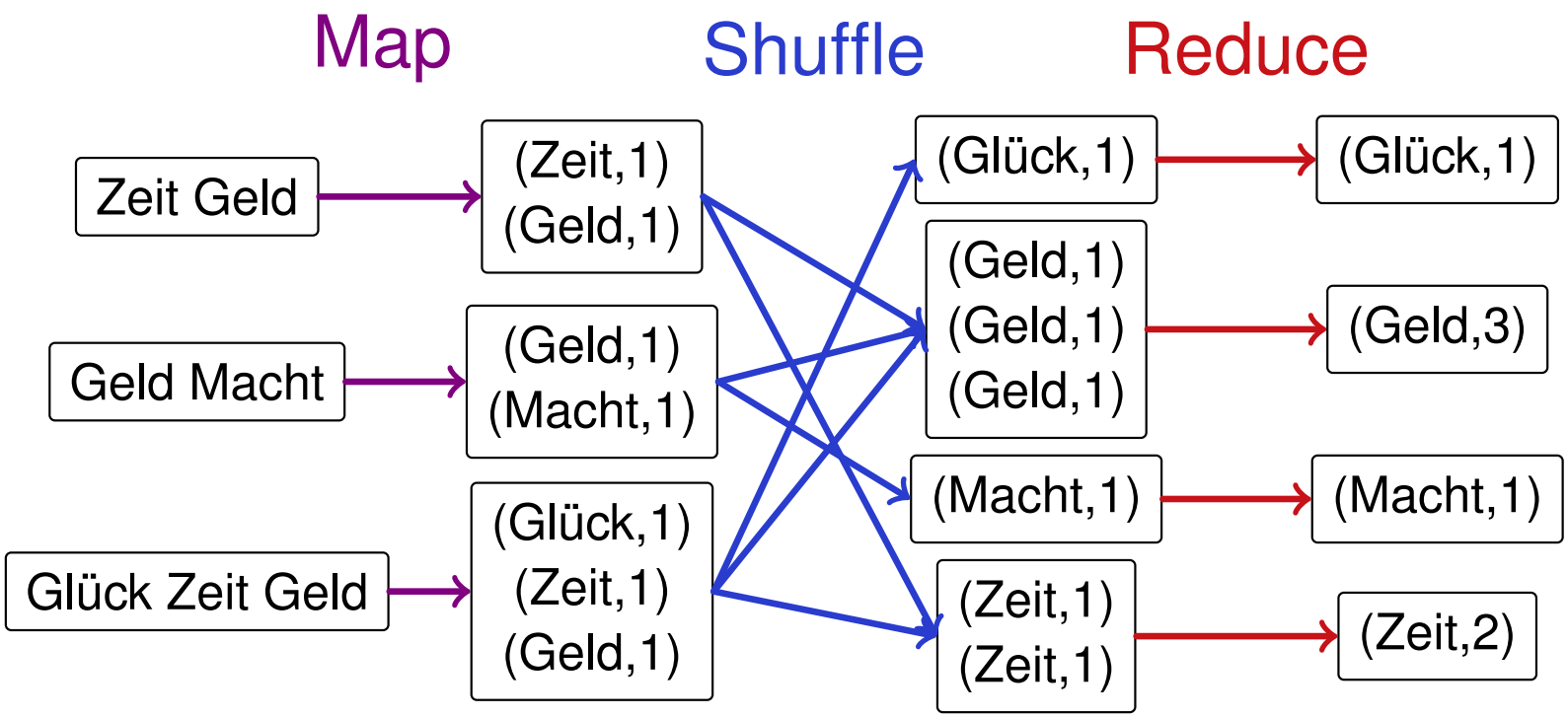
Karlsruher Institut für Technologie

Projektpraktikum: Verteilte Datenverarbeitung mit MapReduce

Timo Bingmann, Peter Sanders und Sebastian Schlag | 21. Oktober 2014 @ PdF Vorstellung

INSTITUTE OF THEORETICAL INFORMATICS – ALGORITHMICS

Was ist Map/Reduce?



Warum Map/Reduce?

- Eine einfache softwaretechnische Abstraktion mit
  - automatischer Parallelisierung von unabhängigen Operationen (map) und Aggregationen (reduce),
  - automatischer Verteilung der Daten und Arbeit,
  - automatischer Fehlertoleranz gegenüber Ausfall von Hardware.

$\Rightarrow$ **MapReduce-Framework**

Warum Map/Reduce?

- Eine einfache softwaretechnische Abstraktion mit
  - automatischer Parallelisierung von unabhängigen Operationen (map) und Aggregationen (reduce),
  - automatischer Verteilung der Daten und Arbeit,
  - automatischer Fehlertoleranz gegenüber Ausfall von Hardware.

$\Rightarrow$ **MapReduce-Framework**

Wo ist Map/Reduce?

- Programmiermodell für ein verteiltes System.
- Implementierung, z.B. in 

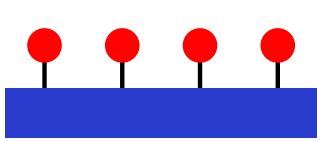
und weiteren experimentellen Frameworks (u.a. auch in C++): (Boost.MapReduce, Sector/Sphere, mapreduce-lite).
- zur Berechnung von: PageRank (spare Matrixmultiplikation), parallele Bildverarbeitung, Aggregation von Statistiken, maschinelles Lernen, etc.
- NICHT das gleiche wie verteilte Dateisysteme oder (verteilte) Datenbanken  

.

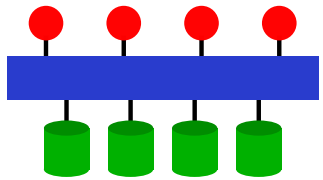
Was kann man da noch machen?

Ziel: Ausprobieren und Entwickeln von besseren Algorithmen für die Kernprobleme.

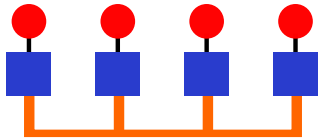
- 1. Etappe: eine prototypische, kanonische Implementierung eines MapReduce Frameworks für eines der Speichermodelle:



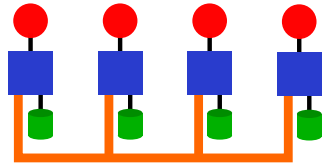
shared,



external,



distributed,



oder distributed external.

- (Prototyp, aber: einfache Applikationen sollen laufen.)
- 2. Etappe: performance-relevante Teilprobleme fokussieren.

Mögliche Verbesserungen

- Bessere Lokalität der Shuffle Operation durch
  - Erkennung eindeutiger Elemente durch Hashing.
  - Anpassung der Shuffles mit Teilreduktionen an die Netzwerk Topologie.
- Weniger Overhead bei der Fehlertoleranz durch
  - intelligente Quersummen, um nicht alles zu duplizieren.
  - (Hadoop: alles 3-fach gespeichert).
- Scheduling der Datenverarbeitung und -redistribution
  - und dabei Optimierung für mehrere MapReduce Schritte,
  - Pipelining von Operationen und Netzwerktransfers,
  - oder sogar der Energieeffizienz.

Voraussetzungen

Wir suchen: 3 – 5 engagierte Studenten, mit

- Spaß am Entwickeln und Analysieren verteilter Algorithmen,
- Grundkenntnissen in C++, Linux und Lernbereitschaft für mehr,
- und guten Noten in PSE und Algorithmen 1 & 2, oder algorithmischem Talent.

Wir bieten:

- Ein Forschungsprojekt im top-aktuellen “Big Data” Bereich.
- Unterstützung und Erfahrung bei der Umsetzung des Projekts.

Meldet Euch bei uns, bis **Montag, 27.10.2014**:

- Timo Bingmann und Sebastian Schlag
  Raum 222 und 210, Infobau – Mail: `{vor.nachname}@kit.edu`.

Weitere Lehrveranstaltungen

Vertiefungsfach:

- Algorithmentechnik (VF2)
- Parallelverarbeitung (VF5)

Empfohlene Vorlesungen:

- Parallele Algorithmen (WS)
- Algorithm Engineering (SS)

Weitere interessante Vorlesungen:

- Randomisierte Algorithmen (WS)
- Parallelrechner und Parallelprogrammierung (SS)
- Modelle der Parallelverarbeitung (SS)

Technologie-Stack Bingo

Vtune	ProtoBuf	ØMQ	gmock	OpenMP
BDB	SDSL	Boost	STXXL	TBB
ASIO	cmake	C++11	perf	MCSTL
Ceph	MsgPack	git	LZ4	mem cache
dSBF	UDP	doxygen	Jenkins	gtest

Karlsruher Institut für Technologie

Projektpraktikum: Verteilte Datenverarbeitung mit MapReduce

Timo Bingmann, Peter Sanders und Sebastian Schlag | 21. Oktober 2014 @ PdF Vorstellung

INSTITUTE OF THEORETICAL INFORMATICS – ALGORITHMICS