



Graphalgorithmen in massiv parallelen Umgebungen

Heinrich Latreider

Grundseminar Vortrag am 27.06.2017

Betreuer: Prof. Dr. Olaf Zukunft



Graphalgorithmen in massiv parallelen Umgebungen

Inhaltsübersicht

- **Motivation**
- **Graphentheorie**
- **Big Data**
- **Graph Processing**
- **Ausblick**
- **Konferenzen**

Graphalgorithmen in massiv parallelen Umgebungen

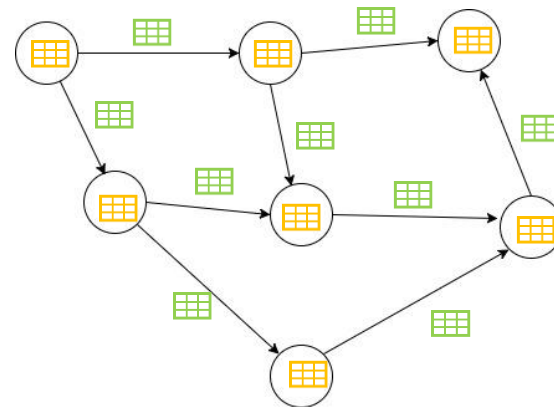
Motivation

- **Komplette Neuorientierung**
- **Bachelor AI**
 - Graphentheorie
- **Interesse an verteilter Datenverarbeitung**
- **Wie lassen sich Graphen parallel verarbeiten?**

Graphalgorithmen in massiv parallelen Umgebungen

Graphentheorie

- Teilgebiet der Mathematik
- Anfänge bis in das 18. Jhd.
- Knoten und Kanten als zentrale Elemente
- Viele Probleme auf Graphen zurückführbar
- Property Graph als Erweiterung



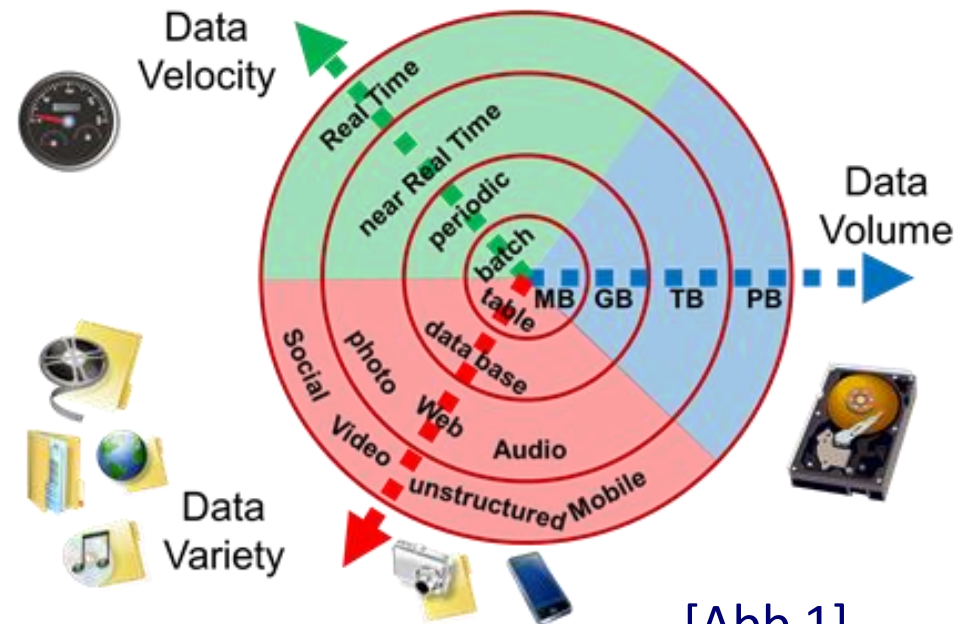


Graphentheorie - Algorithmen

- **Suchalgorithmen (Breiten- bzw. Tiefensuche)**
- **Kürzeste Pfade**
- **Minimale Spannbäume**
- **Flüsse**
- **Matching**
- **uvm.**

Big Data

- 3 V's
 - Velocity
 - Volume
 - Variety

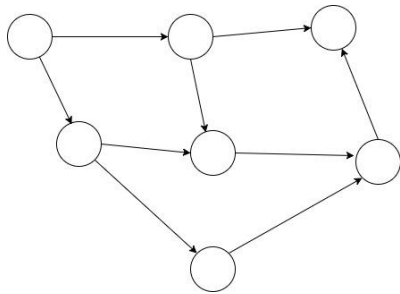


[Abb 1]

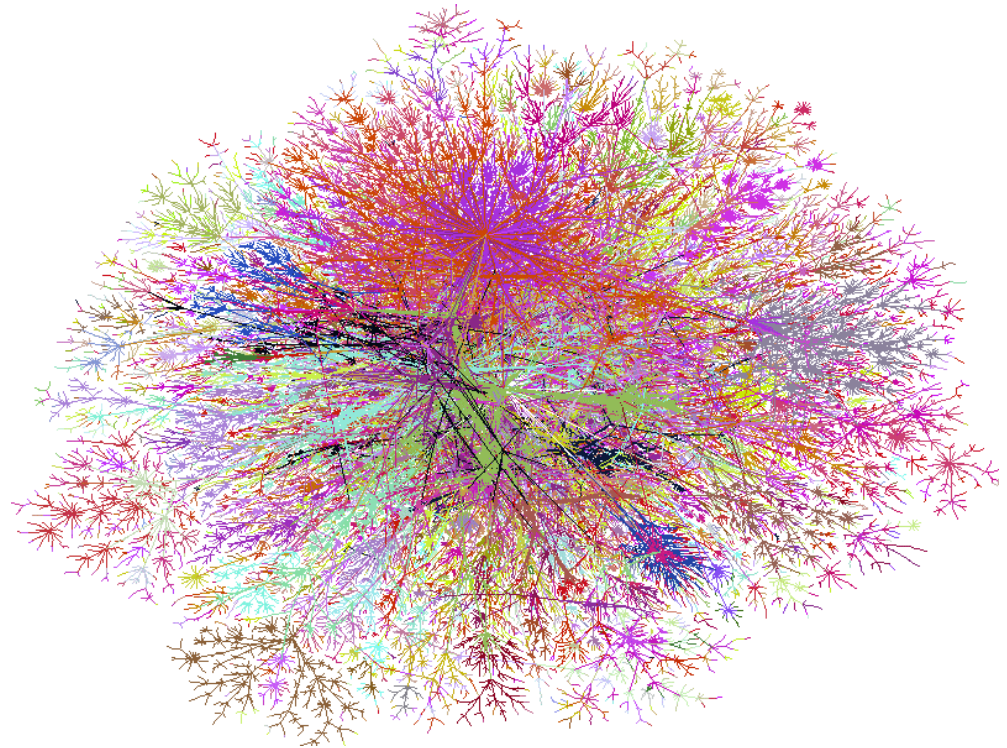
- Effiziente Verarbeitung mit „herkömmlichen“ Technologien nicht mehr möglich

Graphalgorithmen in massiv parallelen Umgebungen

Big Data + Graphen



VS.



[Abb 2]

Graphalgorithmen in massiv parallelen Umgebungen

Anwendungsgebiete

- **Social Media → Social Analysis**
 - Facebook
 - Twitter
- **Web Graphs → PageRank**
 - Google Search
 - Wikipedia
- **Recommendation → Collaborative Filtering**
 - Neighborhood Methode

Big Data Processing

MapReduce (am Beispiel Wordcount)

- **Eingabetext in n Teiltex te aufteilen**
- **Map:**
 - Paralleles Zählen der Wörter des Teiltex te \rightarrow Wörter abgebildet auf Vorkommen im Text
- **Shuffle**
 - Gruppierung der Teilergebnisse nach Wörtern
- **Reduce**
 - Wörter aus Teilergebnissen zusammenaddieren

Big Data Processing

Problem:

- Graphalgorithmen sind iterativ
 - Iterationen als Folge von MapReduce Jobs ineffizient (Netzwerklast + I/O)
 - MapReduce für Graphalgorithmen ungeeignet
- Daher andere Ansätze zur Graphverarbeitung nötig

Graph Processing – Paper

- **Pregel: A System for Large-Scale Graph Processing (2010)**
- **PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs (2012)**

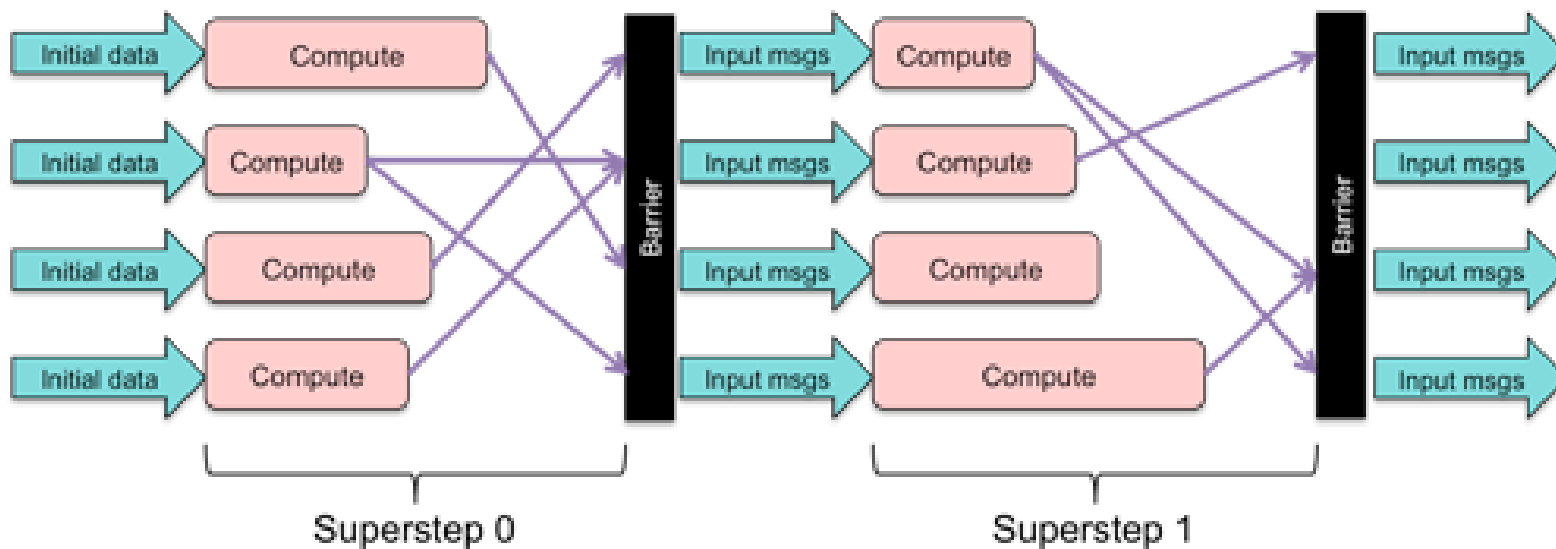
Graphalgorithmen in massiv parallelen Umgebungen

Pregel

- „Think like a Vertex“
- Inspiriert vom Bulk Synchronous Parallel Modell
- Ablauf: Folge von Supersteps
- Master / Worker Architektur
- Knoten haben zwei Zustände
 - Aktiv
 - Inaktiv

Pregel

Bulk Synchronous Parallel



[Abb 3]

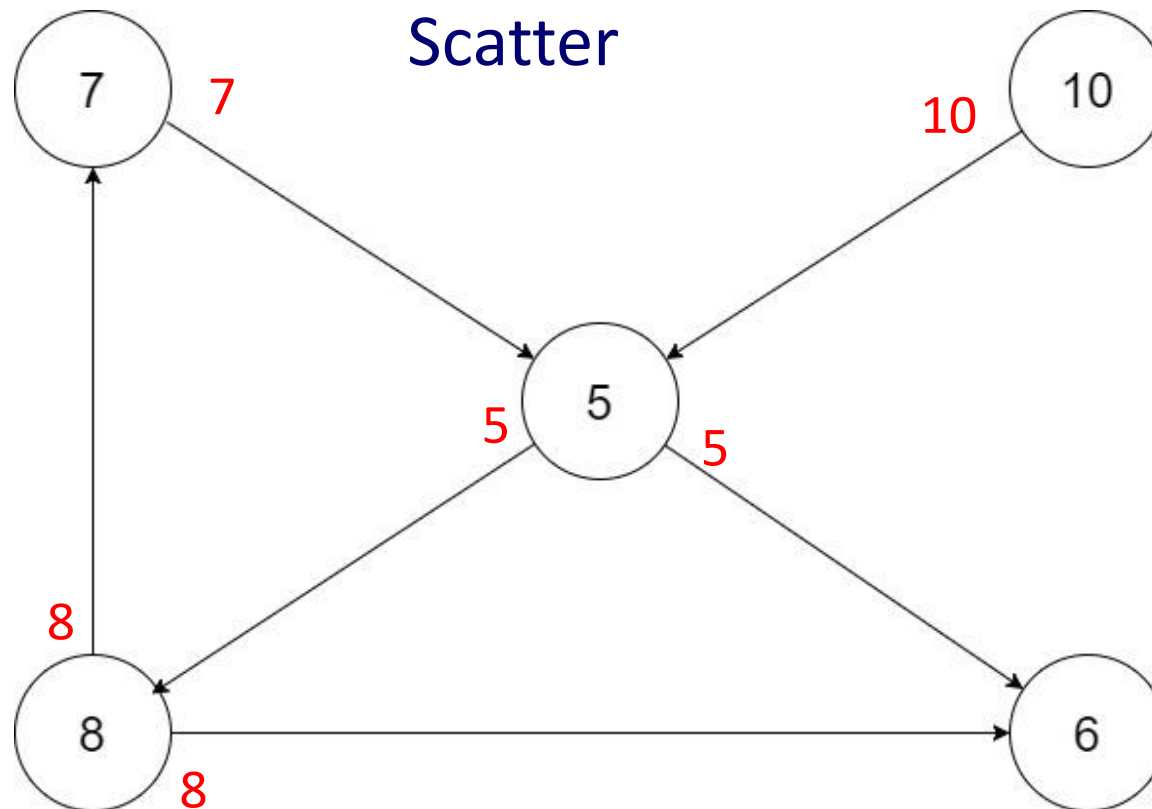
Graphalgorithmen in massiv parallelen Umgebungen

Pregel

Superstep – Phasen

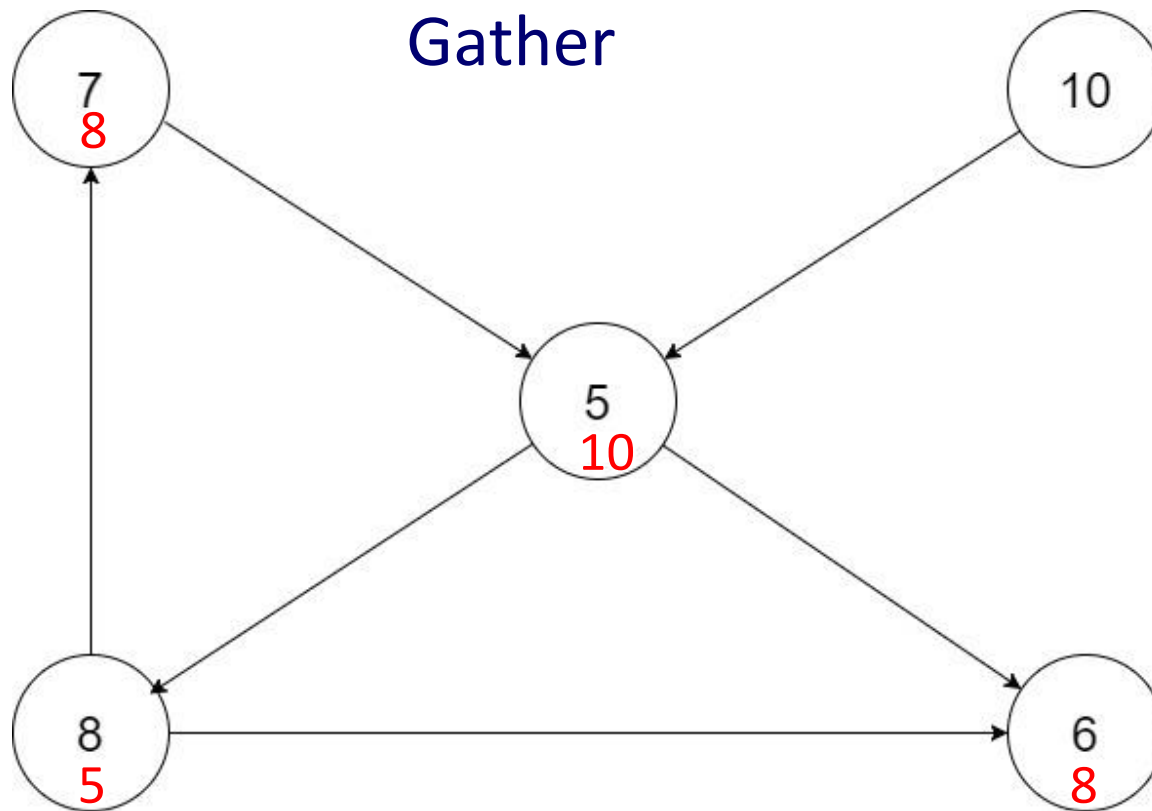
- **Gather:**
 - Alle eingehenden Nachrichten sammeln
 - Message Combiner zum Aggregieren
- **Apply:**
 - Evtl. eigenen Zustand anpassen
- **Scatter:**
 - Nachrichten über ausgehende Kanten senden

Pregel - Beispiel

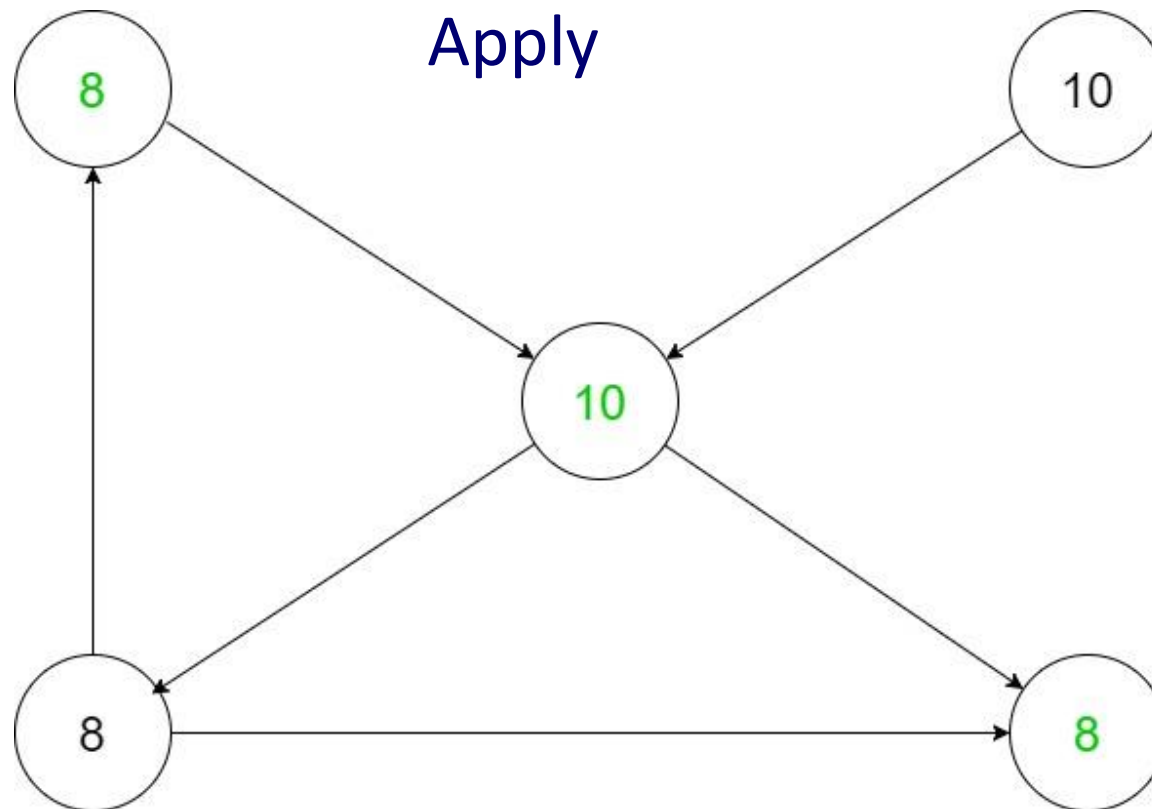


Graphalgorithmen in massiv parallelen Umgebungen

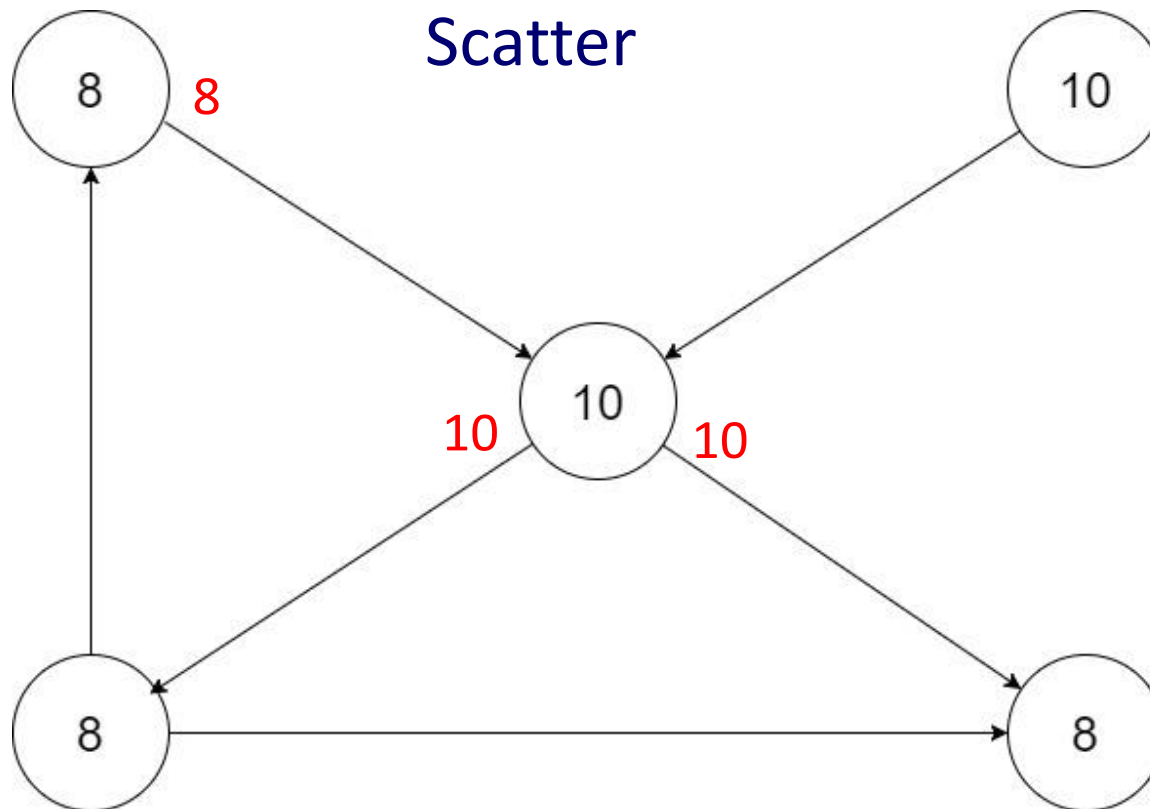
Pregel - Beispiel



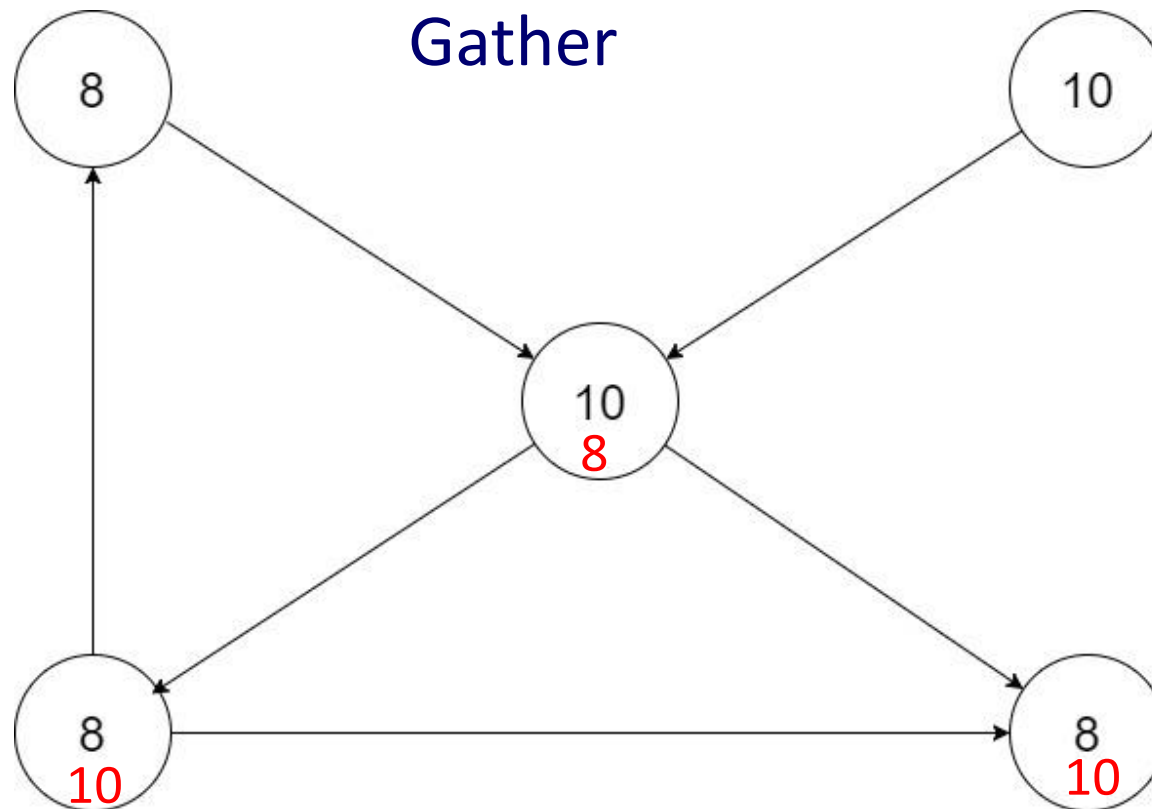
Pregel - Beispiel



Pregel - Beispiel

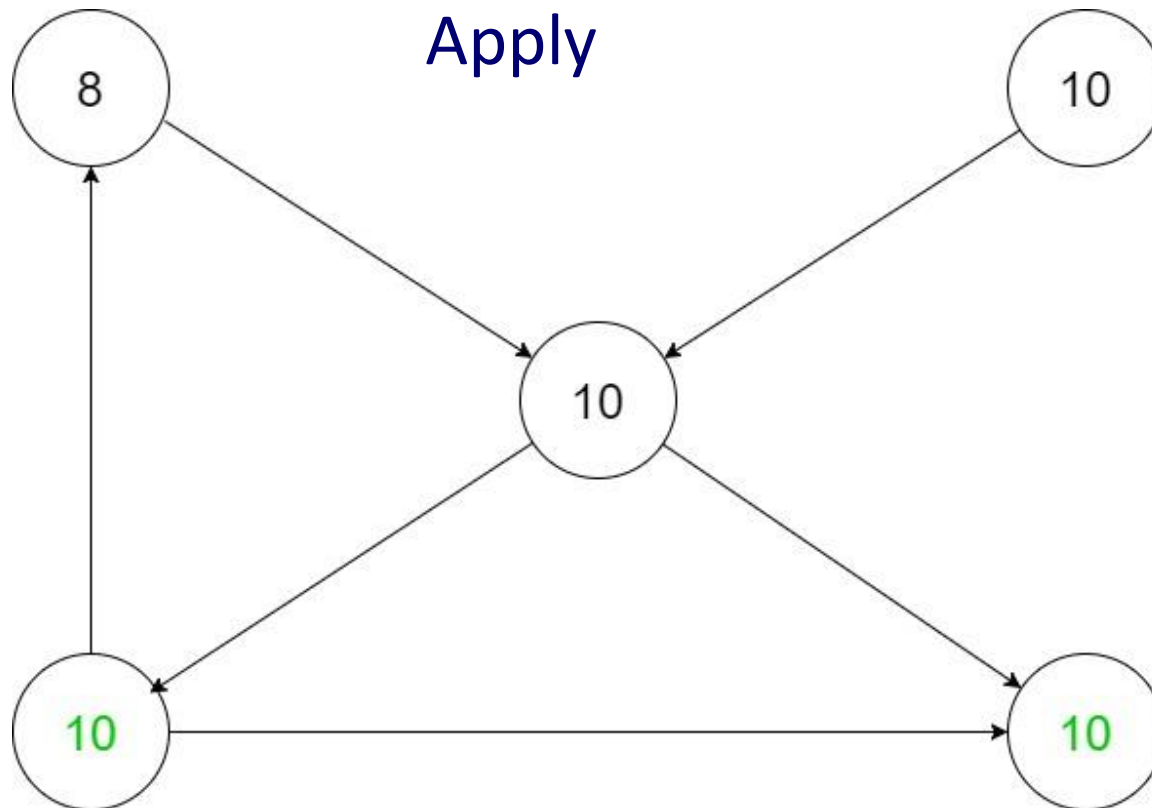


Pregel - Beispiel

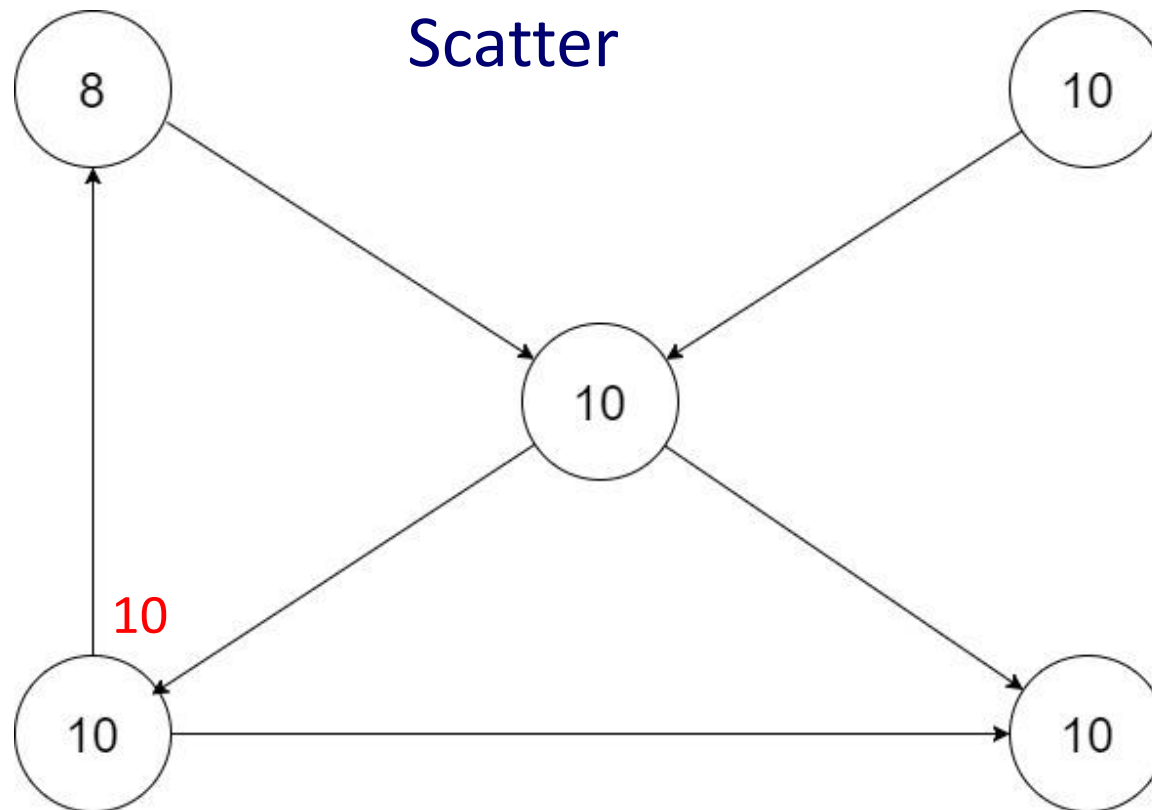


Graphalgorithmen in massiv parallelen Umgebungen

Pregel - Beispiel

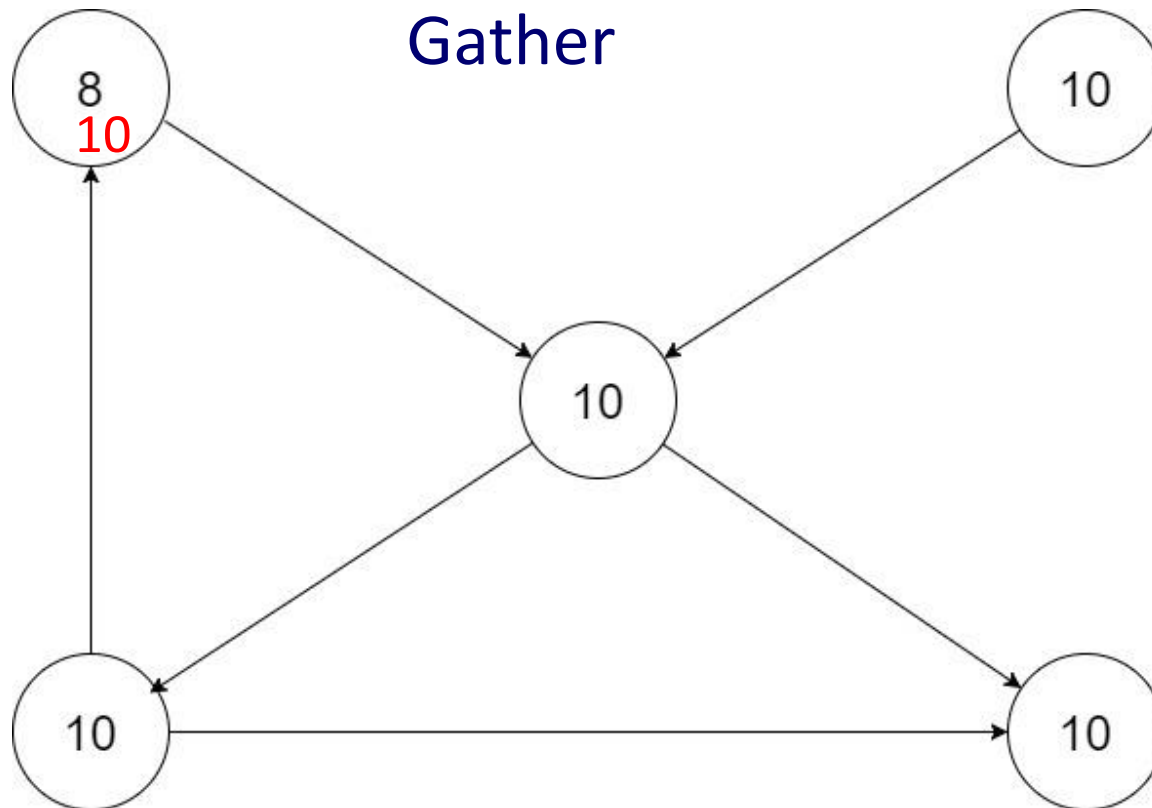


Pregel - Beispiel



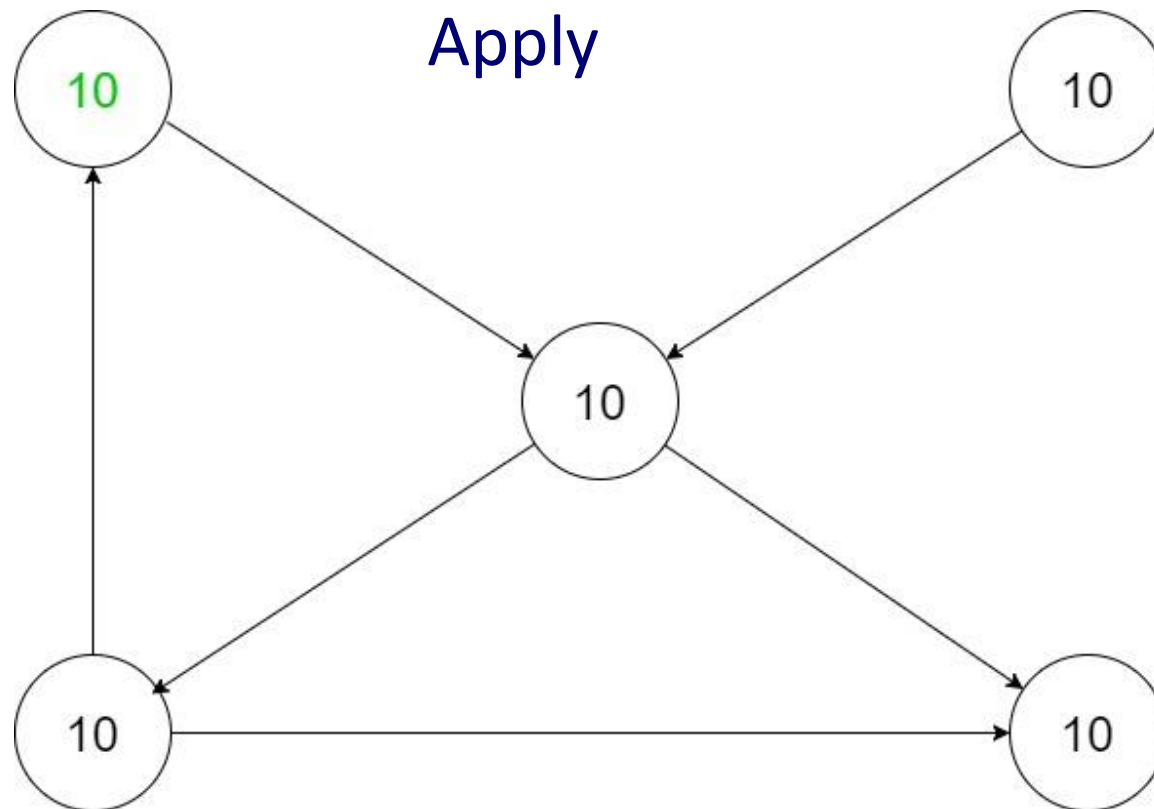
Graphalgorithmen in massiv parallelen Umgebungen

Pregel - Beispiel



Graphalgorithmen in massiv parallelen Umgebungen

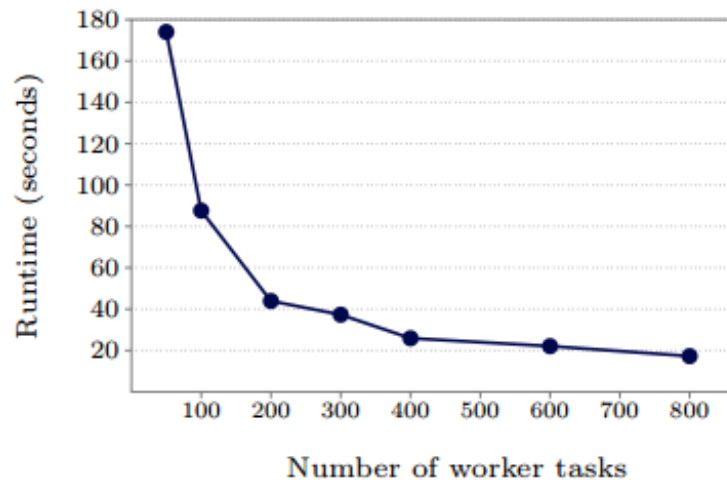
Pregel - Beispiel



Graphalgorithmen in massiv parallelen Umgebungen

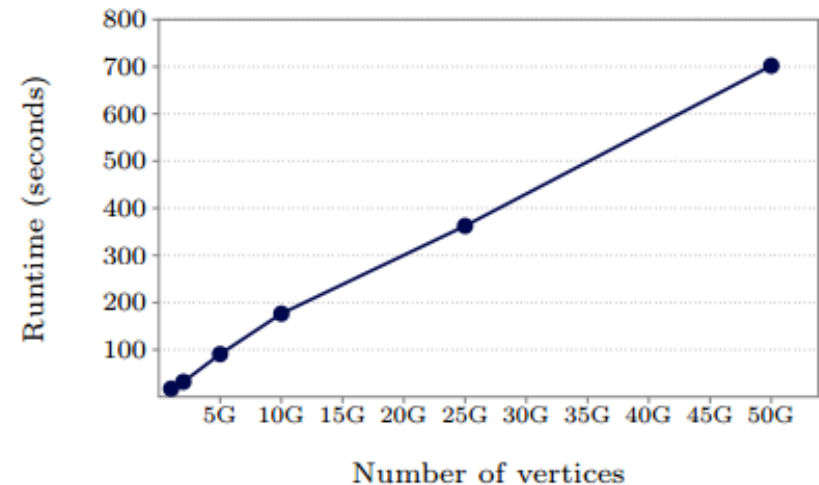
Pregel

Skalierbarkeit bzgl. Anzahl der Worker



$$|V| = 10^9$$

Skalierbarkeit bzgl. Größe des Graphen



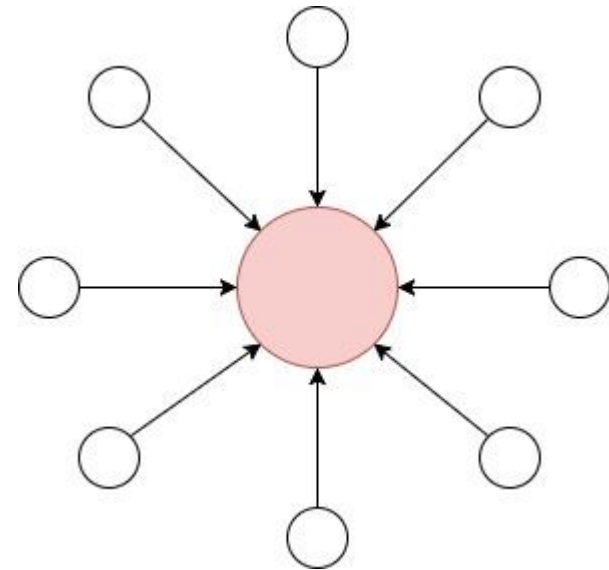
$$[1]$$

Graphalgorithmen in massiv parallelen Umgebungen

PowerGraph

Probleme:

- „Natürliche“ Graphen
- Power-Law Degree Distribution
 - Rechenaufwand unbalanciert
 - Partitionierung
 - Kommunikation
 - Begrenzte Ressourcen pro Maschine





PowerGraph

- **Verarbeitung „natürlicher“ Graphen**
- **Sowohl synchrone als auch asynchrone Graphverarbeitung**
- **Vertex Cut**

PowerGraph – Vertex Cut

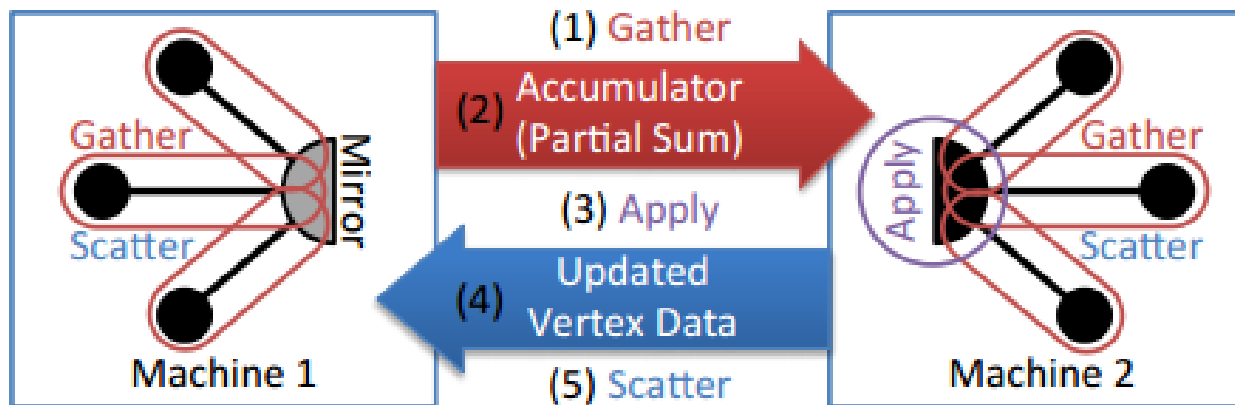
- **Wahrscheinlichkeit, dass eine Kante bei zufälliger Verteilung auf p Maschinen geschnitten wird:**

$$1 - \frac{1}{p} \quad [2]$$

- **Netzwerk- und Speicheroverhead**
- **Beispiel:**
 - Bei $p = 2$ sind 50% der Kanten geschnitten
 - Bei $p = 20$ sind 95% der Kanten geschnitten

PowerGraph – Vertex Cut

- Stattdessen High-Degree Vertices auf mehrere Maschinen verteilen



[2]

Graph Processing – Frameworks



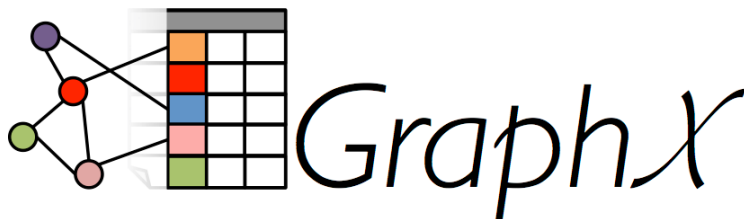
[Abb 4]



[Abb 5]



[Abb 8]



[Abb 6]



[Abb 7]

Graphalgorithmen in massiv parallelen Umgebungen

Ausblick

- **Grundseminar**
 - Überblick über existierende Ansätze und Algorithmen zur Graphverarbeitung verschaffen
- **Grundprojekt**
 - Einarbeitung in Graph Processing Frameworks
 - Erste Experimente
- **Weiteres Vorgehen:**
 - Weniger Fokus auf einen konkreten Algorithmus
 - ? Vergleich verschiedener Frameworks
 - ? Entwicklung einer Algorithmenbibliothek



Konferenzen

- IEEE International Conference on Big Data (IEEE Big Data)
- ACM SIGMOD – Management of Data
- USENIX Symposium on Operating Systems Design and Implementation (OSDI)

Graphalgorithmen in massiv parallelen Umgebungen

Quellen

- (1) Grzegorz Malewicz, Matthew H. Austern, Aart J.C Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. 2010. **Pregel: a system for large-scale graph processing**. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (SIGMOD '10)*. ACM, New York, NY, USA, 135-146. DOI= <http://dx.doi.org/10.1145/1807167.1807184>
- (2) Joseph E. Gonzalez, Yucheng Low, Haijie Gu, Danny Bickson, and Carlos Guestrin. 2012. **PowerGraph: distributed graph-parallel computation on natural graphs**. In *Proceedings of the 10th USENIX conference on Operating Systems Design and Implementation (OSDI'12)*. USENIX Association, Berkeley, CA, USA, 17-30.
- (3) Leslie G. Valiant. 1990. **A bridging model for parallel computation**. *Commun. ACM* 33, 8 (August 1990), 103-111. DOI=10.1145/79173.79181 <http://doi.acm.org/10.1145/79173.79181>
- (4) Jeffrey Dean and Sanjay Ghemawat. 2008. **MapReduce: simplified data processing on large clusters**. *Commun. ACM* 51, 1 (January 2008), 107-113. DOI= <http://dx.doi.org/10.1145/1327452.1327492>
- (5) Sergey Brin and Lawrence Page. 1998. **The anatomy of a large-scale hypertextual Web search engine**. In *Proceedings of the seventh international conference on World Wide Web 7 (WWW7)*, Philip H. Enslow, Jr. and Allen Ellis (Eds.). Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 107-117.
- (6) Big Data, <https://www.gi.de/service/informatiklexikon/detailansicht/article/big-data.html> (25.06.17)
- (7) Graphalgorithmen, <https://www-m9.ma.tum.de/Allgemeines/GraphAlgorithmen> (25.06.17)

Graphalgorithmen in massiv parallelen Umgebungen

Abbildungen

[Abb 1] <https://www.gi.de/service/informatiklexikon/detailansicht/article/big-data.html>

[Abb 2] <http://jameskaskade.com/?p=2093>

[Abb 3] <http://www.book-ebooks.com/products/reading-epub/product-id/4022208/title/Cloud-Computing%2Bf%25C3%25BCr%2BUnternehmen.html>

[Abb 4] <https://hama.apache.org/>

[Abb 5] <http://giraph.apache.org/>

[Abb 6] <https://spark.apache.org/docs/latest/graphx-programming-guide.html>

[Abb 7] <https://flink.apache.org/>

[Abb 8] <https://turi.com>

Danke für die Aufmerksamkeit!

Fragen?