High Performance Computing, Big Data und Machine Learning

Hochleistungs-Ein-/Ausgabe

Michael Kuhn

Wissenschaftliches Rechnen
Fachbereich Informatik
Universität Hamburg

2018-07-06

Big Data

- Big Data bezeichnet Daten, die zu groß oder komplex für traditionelle Analyse sind
  - U. a. Probleme beim Erfassen, Speichern, Analysieren und Teilen der Daten
- Ursprünglich über drei Konzepte definiert
  - Volume: Umfang der Daten
  - Velocity: Durchsatz beim Datentransfer
  - Variety: Datentypen und -formate
- Neuerdings zusätzliche Konzepte
  - Veracity: Rauschen bzw. Störungen in den Daten
  - Value: Wert der Daten
  - Validity, Volatility etc.

Big Data…

- Daten fallen in unterschiedlichsten Bereichen an
  - Überwachungssysteme
  - Kommunikation
  - Gesundheitssystem
  - Smart Home
  - Internet of Things
  - Behörden- und Unternehmensdaten

Volume

- Größe, die als Big Data bezeichnet wird, verändert sich ständig
  - Hängt außerdem von den Fähigkeiten der Benutzer und Werkzeuge ab
  - Man geht heutzutage von einer Untergrenze im Terabyte-Bereich aus
- Struktur der Daten spielt auch eine Rolle
  - Strukturierte Daten sind unter Umständen weniger aufwendig als unstrukturierte
- Beispiele
  - Wikipedia: Über 50 GB (nur englische Wikipedia, ohne History) [16]
  - CERN: Über 200 PB (Bandarchiv) [12]
  - The Internet Archive: Über 50 PB [15]

Velocity

- Durchsatz hängt auch vom aktuellen Stand der Technik ab
  - Je nach Daten Untergrenze im Megabyte- oder Gigabyte-pro-Sekunde-Bereich
- Muss nicht notwendigerweise gespeichert werden
  - Manchmal In-situ-Analyse bzw. Datenreduktion
  - Mehrere TB/s trotzdem schwierig zu handhaben
- Beispiele
  - CERN: 25 GB/s [4]
  - SKA: 100 PB/s (Speicherdurchsatz) [3]

Variety

- Je nach betrachtetem Bereich unterschiedliche Datentypen und -formate
  - Strukturierte Daten
    - Wissenschaftliche Daten (HDF, NetCDF etc.)
  - Semi-strukturierte Daten
    - XML, JSON etc.
  - Unstrukturierte Daten
    - Text (z. B. Wikipedia-Artikel) etc.
- Fokus sind unstrukturierte Daten
  - Strukturierte Daten bereits gut durch andere Ansätze abgedeckt

Veracity

- Qualität der Daten ist wichtig für mögliche Erkenntnisse
  - Analyse mit fehlerhaften Daten führt meist zu schlechten Ergebnissen
  - Andererseits sind auch fehlerfreie Daten bei fehlerhafter Analyse nutzlos
- Fehler können viele Ursachen haben
  - Fehlerhafte Hardware (z. B. Messgeräte)
  - Fehlerhafte Algorithmen
  - Menschliche Fehler
- Qualitätssicherung ist daher ein wichtiger Aspekt

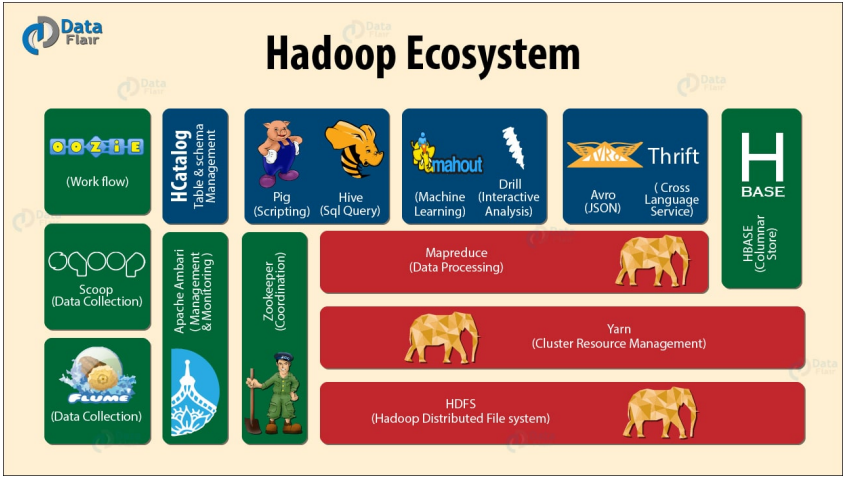
Value

- Wert der Daten von Benutzern und Umfeld abhängig
  - Wissenschaftliche Daten haben einen anderen Wert als kommerzielle
- Weiterverarbeitung führt üblicherweise zu einer Wertsteigerung
  - Z. B. wertlose Rohdaten, die weitergehend analysiert werden

Probleme

- Keine einheitliche Definition
  - Für jeden Anwender sind andere Aspekte wichtig
- Außerdem keine einheitliche Software-/Hardware-Umgebung
  - Big Data kann auch wissenschaftliche Daten im Hochleistungsrechnen bezeichnen
- Häufig synonym verwendet mit Googles MapReduce-Ansatz
  - Meistens wird Software-Stack des Apache-Projektes (z. B. Hadoop) genutzt

Software-Umgebung [5]



Big Data vs. Kommandozeile

- Big-Data-Werkzeuge sind nicht immer für die Lösung eines Problems geeignet [1]
  - MapReduce zur Berechnung des Gewinn/Verlust-Verhältnisses von Schachspielen
- Archiv hat eine Größe von 1,75 GB und enthält zwei Millionen Schachspiele

Big Data vs. Kommandozeile

- Big-Data-Werkzeuge sind nicht immer für die Lösung eines Problems geeignet [1]
  - MapReduce zur Berechnung des Gewinn/Verlust-Verhältnisses von Schachspielen
- Archiv hat eine Größe von 1,75 GB und enthält zwei Millionen Schachspiele
  - MapReduce-Job brauchte ca. 26 Minuten (entspricht einem Durchsatz von 1,14 MB/s)

Big Data vs. Kommandozeile

- Big-Data-Werkzeuge sind nicht immer für die Lösung eines Problems geeignet [1]
  - MapReduce zur Berechnung des Gewinn/Verlust-Verhältnisses von Schachspielen
- Archiv hat eine Größe von 1,75 GB und enthält zwei Millionen Schachspiele
  - MapReduce-Job brauchte ca. 26 Minuten (entspricht einem Durchsatz von 1,14 MB/s)
- Alternatives Archiv hat eine Größe von 3,46 GB
  - Kommandozeilenwerkzeuge lösen das gleiche Problem in ca. 12 Sekunden (270 MB/s)

Big Data vs. Kommandozeile

- Big-Data-Werkzeuge sind nicht immer für die Lösung eines Problems geeignet [1]
  - MapReduce zur Berechnung des Gewinn/Verlust-Verhältnisses von Schachspielen
- Archiv hat eine Größe von 1,75 GB und enthält zwei Millionen Schachspiele
  - MapReduce-Job brauchte ca. 26 Minuten (entspricht einem Durchsatz von 1,14 MB/s)
- Alternatives Archiv hat eine Größe von 3,46 GB
  - Kommandozeilenwerkzeuge lösen das gleiche Problem in ca. 12 Sekunden (270 MB/s)

```
find . -type f -name '*.pgn' -print0 | xargs -0 -n4 -P4 mawk '/Result/ { split($0, a,
    ↪ "-"); res = substr(a[1], length(a[1]), 1); if (res == 1) white++; if (res ==
    ↪ 0) black++; if (res == 2) draw++ } END { print white+black+draw, white, black,
    ↪ draw }' | mawk '{games += $1; white += $2; black += $3; draw += $4; } END {
    ↪ print games, white, black, draw }'
```

Software-Umgebungen [6, 2]

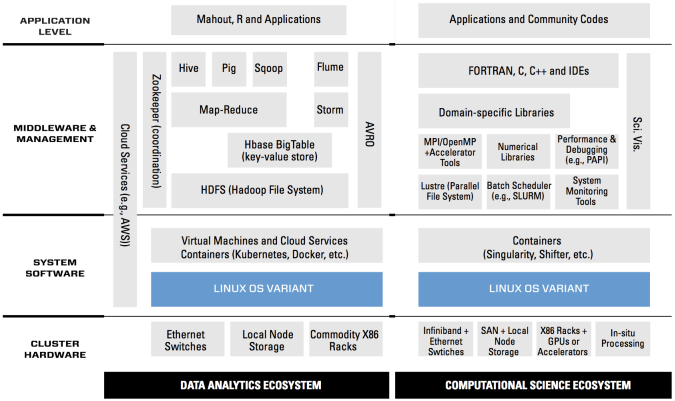
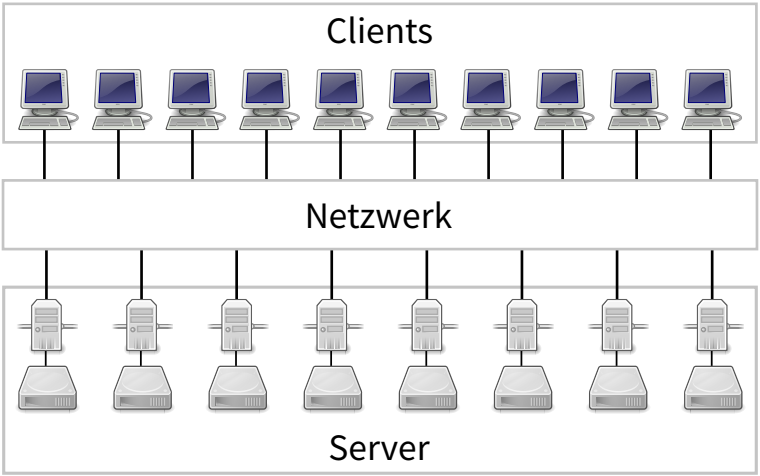


Figure 1: Different software ecosystems for high-end Data Analytics and for traditional Computational Science. [Credit: Reed and Dongarra [66]]

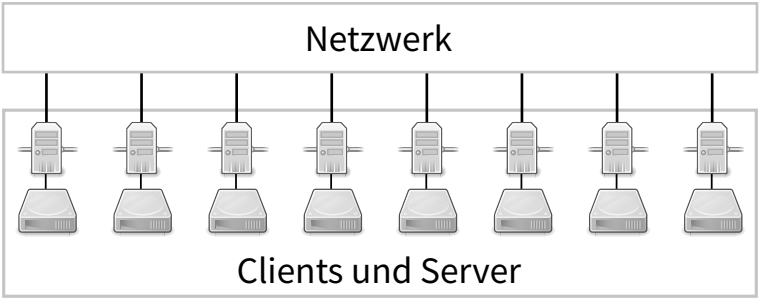
Software-Umgebungen…

- Big Data wird immer wichtiger
  - Häufig leistungstechnisch suboptimal
- Hadoop nutzt normalerweise HDFS
  - Daten werden auf lokale Speichergeräte kopiert
  - Kommunikation über HTTP
- Zunehmend Unterstützung für Big-Data-Anwendungen
  - Lustre, OrangeFS etc.

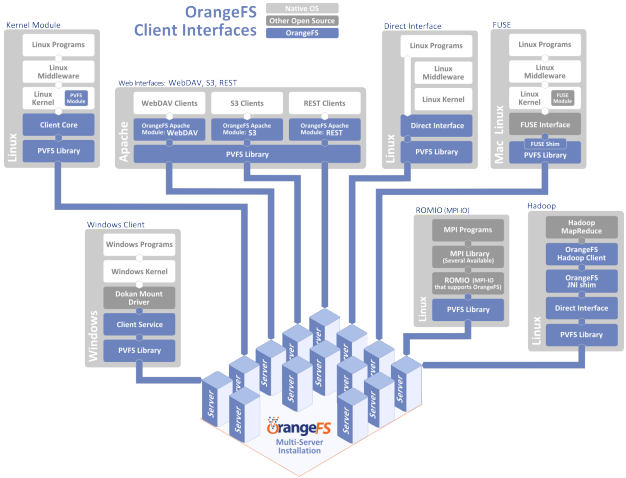
Paralleles verteiltes Dateisystem



Hadoop-Cluster



Schnittstellen [13]



Hadoop auf OrangeFS [9]

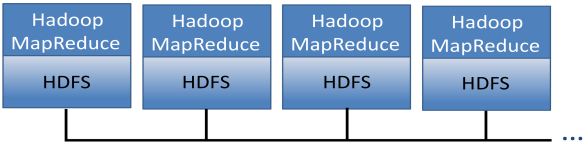


Fig. 2: Typical Hadoop with HDFS local storage (HDFS in short).

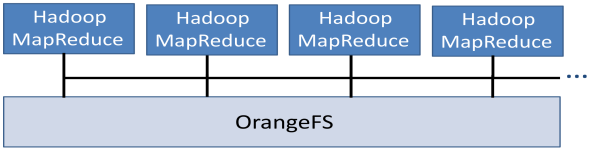
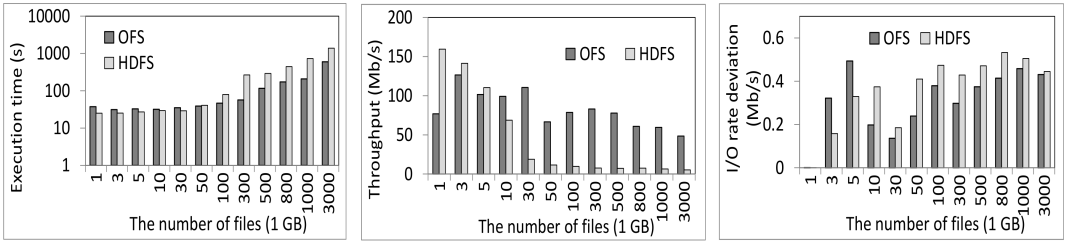


Fig. 3: Hadoop with the OrangeFS dedicated storage (OFS in short).

- HDFS: Daten liegen bestenfalls lokal
- OrangeFS: Daten liegen immer auf entfernten Servern

Hadoop auf OrangeFS… [9]

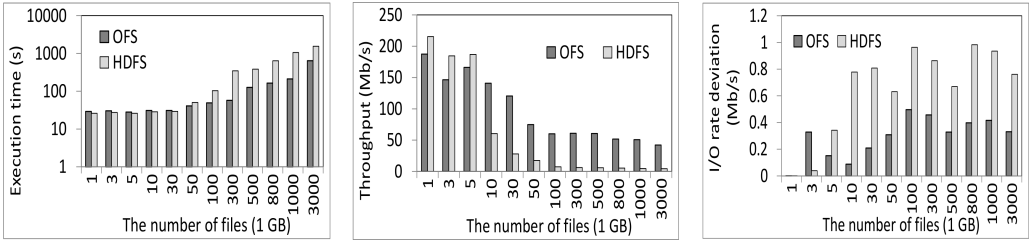


(a) Execution time. (b) Throughput. (c) I/O rate deviation.

Fig. 4: Performance of I/O-intensive application *TestDFSIO* write test.

- HDFS ist für kleine Dateizahlen schneller
  - Leistung bricht mit vielen Dateien massiv ein

Hadoop auf OrangeFS… [9]



(a) Execution time. (b) Throughput. (c) I/O rate deviation.

Fig. 5: Performance of I/O-intensive application *TestDFSIO* read test.

- HDFS und OrangeFS auch mit wenigen Dateien fast gleichauf

Hadoop auf OrangeFS… [9]

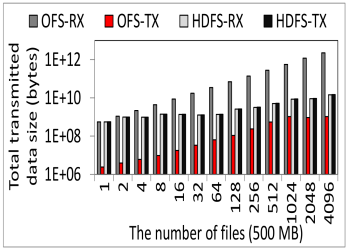


Fig. 10: Total transmitted data size of data-intensive *Grep*.

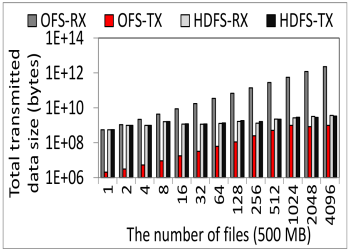


Fig. 11: Total transmitted data size of data-intensive *Wordcount*.

- Deutlich mehr empfangene Daten in OrangeFS, da entfernte Zugriffe stattfinden
- HDFS versendet mehr Daten, da Eingabe unter den Mappern ausgetauscht wird

Big Data…

- Andererseits interessante Ansätze aus dem Big-Data-/Cloud-Umfeld
  - Elastizität zur dynamischen Anpassung des Dateisystems
  - Zuschalten von zusätzlichen Dateisystem-Servern bei Bedarf
- Nutzung von Object Stores
  - Viele Anwendungen benötigen keine POSIX-Dateisysteme
  - MPI-IO-Funktionalität kann auf Object Stores abgebildet werden
- Wird im Rahmen des BigStorage-Projektes untersucht
  - Beispiel: Týr ist ein Blob-Speicher mit Unterstützung für Transaktionen [11]
  - Siehe `http://bigstorage-project.eu`

Big Data… [10]

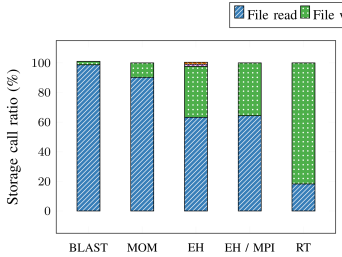


Fig. 1. Measured relative amount of different storage calls to the persistent file system for HPC applications

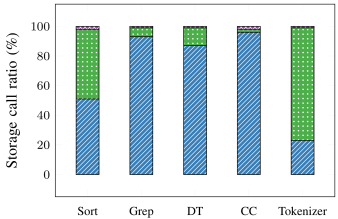


Fig. 2. Measured relative amount of different storage calls to the persistent file system for Big Data applications

- Fast ausschließlich Lese- und Schreiboperationen in HPC-Anwendungen
  - MPI-IO hat sehr eingeschränkten Funktionsumfang
- Wenige Verzeichnisoperationen in Big-Data-Anwendungen
  - Hauptsächlich durch Spark-Framework verursacht

Machine Learning

- Ein Unterbereich der künstlichen Intelligenz
  - Anwendungen lernen und verbessern sich mithilfe von Daten
- Grobe Unterteilung in zwei Ansätze
  - Supervised Learning
    - Lernen mithilfe von Eingabe- und Ausgabedaten
  - Unsupervised Learning
    - Lernen nur mit Eingabedaten
- Beispielhafte Anwendungen
  - Klassifikation
    - Einordnung in eine oder mehrere vorgegebene Klassen
  - Clustering
    - Eigenständige Einordnung in Gruppen

Machine Learning…

- Machine Learning nutzt Modelle
  - Entscheidungsbäume
    - Entscheidung wird aufgrund mehrerer Faktoren getroffen
  - Neuronale Netze
    - Neuronen sind über gewichtete Kanten verbunden und haben Aktivierungsfunktionen

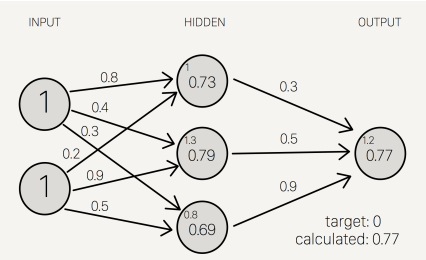


Abbildung: Neuronales Netz [14]

Machine Learning für HPC und Big Data [7]

- Machine-Learning-Ansätze nützlich für HPC- und Big-Data-Probleme
  - Passende Hardware ist in vielen HPC-Systemen bereits vorhanden (GPUs)
  - Große Datenmengen eignen sich sehr gut für Training
- Neuronale Netze können effizient Vorhersagen treffen
  - Training sehr aufwendig, Vorhersagen mit Integer-Arithmetik
- Beispiele
  - LIGO Signal Processing (NCSA): 5000 Mal schneller
  - Analyzing Gravitational Lensing (SLAC Stanford): Millisekunden statt Wochen
  - Tracking Neutrinos (Fermilab): Erkennungsquote um 33 % erhöht

Machine Learning für HPC und Big Data… [7]

1. Anpassung von Simulationen oder Experimenten zwischen Iterationen
   - Dadurch z. B. schnellere Konvergenz
2. Verbesserung existierender Simulationen
   - Simulationsdaten werden für das Training verwendet
3. Ersetzen von numerischen Simulationen durch Vorhersagen
   - Erfordert Umdenken und Umstrukturieren der Anwendungen

Gelernte Indizes [8]

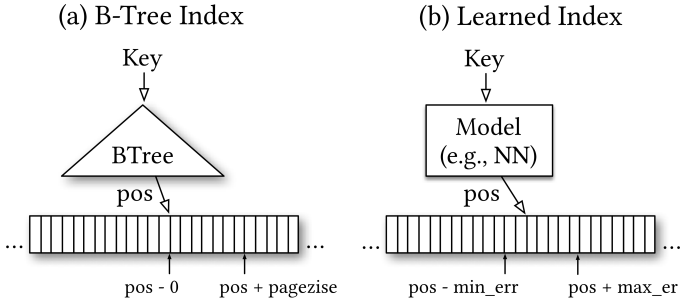


Figure 1: Why B-Trees are models

- B-Baum liefert Speicherseite, die die Daten enthält
- Neuronales Netz liefert ungefähre Position

Gelernte Indizes… [8]

		Map Data			Web Data			Log-Normal Data		
Type	**Config**	Size (MB)	Lookup (ns)	Model (ns)	Size (MB)	Lookup (ns)	Model (ns)	Size (MB)	Lookup (ns)	Model (ns)
Btree	page size: 32	52.45 (4.00x)	274 (0.97x)	198 (72.3%)	51.93 (4.00x)	276 (0.94x)	201 (72.7%)	49.83 (4.00x)	274 (0.96x)	198 (72.1%)
	page size: 64	26.23 (2.00x)	277 (0.96x)	172 (62.0%)	25.97 (2.00x)	274 (0.95x)	171 (62.4%)	24.92 (2.00x)	274 (0.96x)	169 (61.7%)
	page size: 128	13.11 (1.00x)	265 (1.00x)	134 (50.8%)	12.98 (1.00x)	260 (1.00x)	132 (50.8%)	12.46 (1.00x)	263 (1.00x)	131 (50.0%)
	page size: 256	6.56 (0.50x)	267 (0.99x)	114 (42.7%)	6.49 (0.50x)	266 (0.98x)	114 (42.9%)	6.23 (0.50x)	271 (0.97x)	117 (43.2%)
	page size: 512	3.28 (0.25x)	286 (0.93x)	101 (35.3%)	3.25 (0.25x)	291 (0.89x)	100 (34.3%)	3.11 (0.25x)	293 (0.90x)	101 (34.5%)
Learned Index	2nd stage models: 10k	0.15 (0.01x)	98 (2.70x)	31 (31.6%)	0.15 (0.01x)	222 (1.17x)	29 (13.1%)	0.15 (0.01x)	178 (1.47x)	26 (14.6%)
	2nd stage models: 50k	0.76 (0.06x)	85 (3.11x)	39 (45.9%)	0.76 (0.06x)	162 (1.60x)	36 (22.2%)	0.76 (0.06x)	162 (1.62x)	35 (21.6%)
	2nd stage models: 100k	1.53 (0.12x)	82 (3.21x)	41 (50.2%)	1.53 (0.12x)	144 (1.81x)	39 (26.9%)	1.53 (0.12x)	152 (1.73x)	36 (23.7%)
	2nd stage models: 200k	3.05 (0.23x)	86 (3.08x)	50 (58.1%)	3.05 (0.24x)	126 (2.07x)	41 (32.5%)	3.05 (0.24x)	146 (1.79x)	40 (27.6%)

Figure 4: Learned Index vs B-Tree

- Es werden mehrstufige Modelle benutzt
  - Das Modell auf Stufe 1 wählt das Modell auf Stufe 2 aus
- Gelernte Indizes sind in allen Fällen kleiner und schneller

Gelernte Indizes… [8]

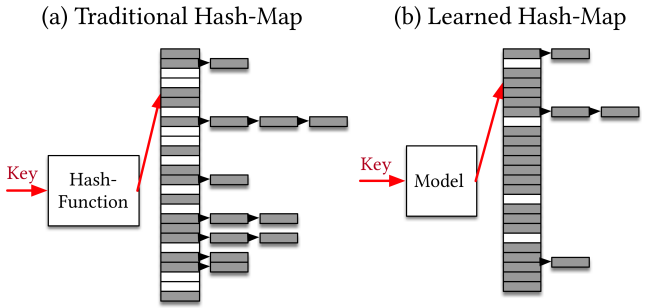


Figure 7: Traditional Hash-map vs Learned Hash-map

- Ansatz kann für unterschiedliche Datenstrukturen genutzt werden
  - Z. B. um die Hash-Funktionen einer Hash-Tabelle zu ersetzen

Gelernte Indizes… [8]

	% Conflicts Hash Map	% Conflicts Model	Reduction
Map Data	35.3%	07.9%	77.5%
Web Data	35.3%	24.7%	30.0%
Log Normal	35.4%	25.9%	26.7%

Figure 8: Reduction of Conflicts

- Gelernte Indizes können die Anzahl an Konflikten deutlich reduzieren
- In manchen Fällen allerdings höhere Lookup-Zeiten
  - In verteilten Architekturen trotzdem sinnvoll, da die Lookup-Zeit vernachlässigbar ist

Zusammenfassung

- Big Data wird zunehmend wichtiger
  - Begriff ist allerdings nicht eindeutig definiert
- HPC und Big Data nutzen unterschiedliche Technologien
  - Parallele verteilte Dateisysteme für HPC, HDFS für Big Data
  - Big-Data-Technologien sind häufig ineffizient
- Machine Learning bietet interessante Perspektiven für HPC und Big Data
  - Verbesserte Leistung und verringerter Speicherverbrauch
- HPC-, Big-Data- und Machine-Learning-Technologien konvergieren
  - Erlaubt vielseitigere und effizientere Systeme

Quellen I

[1] Adam Drake. Command-line Tools can be 235x Faster than your Hadoop Cluster. https://adamdrake.com/command-line-tools-can-be-235x-faster-than-your-hadoop-cluster.html.

[2] J.-C. Andre, G. Antoniu, M. Asch, R. Badia Sala, M. Beck, P. Beckman, T. Bidot, F. Bodin, F. Cappello, A. Choudhary, B. de Supinski, E. Deelman, J. Dongarra, A. Dubey, G. Fox, H. Fu, S. Girona, W. Gropp, M. Heroux, Y. Ishikawa, K. Keahey, D. Keyes, W. Kramer, J.-F. Lavignon, Y. Lu, S. Matsuoka, B. Mohr, T. Moore, D. Reed, S. Requena, J. Saltz, T. Schulthess, R. Stevens, M. Swany, A. Szalay, W. Tang, G. Varoquaux, J.-P. Vilotte, R. Wisniewski, Z. Xu, and I. Zacharov. Big Data and Extreme-Scale Computing: Pathways to Convergence. Technical report, EXDCI Project of EU-H2020 Program and University of Tennessee, 01 2018.

Quellen II

http://www.exascale.org/bdec/sites/www.exascale.org.bdec/files/whitepapers/bdec2017pathways.pdf.

[3] Peter J. Braam. Performance engineering for the SKA telescope. In *Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering, ICPE 2018, Berlin, Germany, April 09-13, 2018*, page 1, 2018.

[4] CERN. Processing: What to record? https://home.cern/about/computing/processing-what-record.

[5] DataFlair Team. Hadoop Ecosystem and their Components – A complete Tutorial. https://data-flair.training/blogs/hadoop-ecosystem-components/.

Quellen III

[6] John Russell. New Blueprint for Converging HPC, Big Data. https://www.hpcwire.com/2018/01/18/new-blueprint-converging-hpc-big-data/.

[7] Karl Freund. What's Hot At SC17: The Synthesis Of Machine Learning & HPC. https://www.forbes.com/sites/moorinsights/2017/11/14/whats-hot-at-sc17-the-synthesis-of-machine-learning-hpc/.

[8] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 489–504, 2018.

Quellen IV

[9] Zhuozhao Li, Haiying Shen, Jeffrey Denton, and Walter Ligon. Comparing application performance on hpc-based hadoop platforms with local storage and dedicated storage. In *2016 IEEE International Conference on Big Data, BigData 2016, Washington DC, USA, December 5-8, 2016*, pages 233–242, 2016.

[10] Pierre Matri, Yevhen Alforov, Alvaro Brandon, Michael Kuhn, Philip H. Carns, and Thomas Ludwig. Could blobs fuel storage-based convergence between HPC and big data? In *2017 IEEE International Conference on Cluster Computing, CLUSTER 2017, Honolulu, HI, USA, September 5-8, 2017*, pages 81–86, 2017.

Quellen V

[11] Pierre Matri, Alexandru Costan, Gabriel Antoniu, Jesús Montes, and María S. Pérez. Týr: blob storage meets built-in transactions. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2016, Salt Lake City, UT, USA, November 13-18, 2016*, pages 573–584, 2016.

[12] Mélissa Gaillard. CERN Data Centre passes the 200-petabyte milestone. https://home.cern/about/updates/2017/07/cern-data-centre-passes-200-petabyte-milestone.

[13] OrangeFS Development Team. OrangeFS Documentation. http://docs.orangefs.com/.

Quellen VI

[14] Steven Miller. Mind: How to Build a Neural Network (Part One). https://stevenmiller888.github.io/mind-how-to-build-a-neural-network/.

[15] The Internet Archive. Petabox. http://archive.org/web/petabox.php.

[16] Wikipedia. Wikipedia:Size of Wikipedia. https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia.