BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

Exposé zur Masterarbeit im Fachbereich VI – Informatik und Medien – der Beuth Hochschule für Technik Berlin zur Erlangung des akademischen Grades **Master of Science (M.Sc.)** im Studiengang **Medieninformatik-Online (Master)**

Vergleich von Streamingframeworks

STORM, KAFKA, FLUME, S4

Eduard Bergen

Matrikelnummer: 769248

s40907@beuth-hochschule.de

Berlin, 27.04.2014

1. Betreuer Prof. Dr. Stefan Edlich
Gutachter Prof. Christoph Knabe

1.1 Einleitung

Mit der enormen Zunahme von Nachrichten durch unterschiedliche Quellen wie Sensoren (RFID) oder Nachrichtenquellen (RFD newsfeeds) wird es schwieriger Informationen beständig abzufragen. Um die Frage zum Beispiel zu klären, welcher Rechner am häufigsten über TCP frequentiert wird, werden unterstützende Systeme notwendig. An dieser Stelle helfen Methoden aus dem Bereich des Complex Event Processing (CEP). Im Spezialbereich Stream Processing von CEP wurden Streaming Frameworks entwickelt, um die Arbeit in der Datenflussverarbeitung zu vereinfachen. In den nächsten Unterkapiteln werden die Ziele der Abschlussarbeit, verwandte Arbeiten, ein Gliederungsentwurf, ein grober Zeitplan und ein vorläufiges Literaturverzeichnis vorgestellt.

1.2 Ziele der Arbeit

Hauptziel dieser Arbeit ist ein Vergleich zwischen den Streaming Frameworks Apache Storm, Apache Kafka, Apache Flume und Apache S4. Dabei sollen die Stärken und Schwächen der einzelnen Streaming Frameworks erarbeitet und gegenübergestellt werden. Zudem soll eine Herangehensweise für den Nutzen in einer Entwicklungsumgebung, sowie in einer produktiven Umgebung gezeigt werden. Außerdem sollen mögliche Anwendungsfälle für die Streaming Frameworks vorgestellt und eingeordnet werden. Eine Analyse und ein Belastungstest mit einer Diskussion sollen die Abschlussarbeit abrunden. Im folgenden Unterkapitel werden Arbeiten im ähnlichen Bereich vorgestellt.

1.3 Verwandte Arbeiten

Neben den genannten Streaming Frameworks befinden sich Anwendungen wie Apache Spark mit Spark Streaming der amplab UC Berkley, STREAM von Stanford, Rainbird von Twitter, Puma von Facebook, IBM InfoSphere, Streambase oder Microsoft StreamInsight in einem frühen Stadium der Entwicklung oder sind bereits kommerziell im Einsatz. Das anstehende Unterkapitel Gliederungsentwurf zeigt eine erste Gliederung des Dokuments zum Abschlussthema "Vergleich von Streaming Frameworks: Storm, Kafka, Flume und S4"

1.4 Gliederungsentwurf

1. Titel/Deckblatt
2. Zusammenfassung
3. Abstract
4. Inhaltsverzeichnis
5. Abbildungsverzeichnis
6. Tabellenverzeichnis
7. Quelltextverzeichnis
8. Einleitung
   (a) Aufgabenstellung und Motivation
   (b) Zielsetzung
   (c) Aufbau

9. Technische Grundlagen
10. Related Work
11. Analyse
12. Zieldefinition
13. Streaming Frameworks
    (a) Apache Storm
    (b) Apache Kafka
    (c) Apache Flume
    (d) Apache S4
14. Anwendungsfall und Prototyp
15. Auswertung
    (a) Benchmark Ergebnisse
    (b) Erkenntnis
16. Schlussbetrachtung
    (a) Zusammenfassung
    (b) Ausblick
    (c) Einschränkungen
17. Anhänge
    (a) Abkürzungsverzeichnis
    (b) Glossar
    (c) Literaturverzeichnis
    (d) Datenträger
18. Eigenständigkeitserklärung separat

Im folgenden Unterkapitel wird der Zeitplan grob gesetzt und im abschließenden Unterkapitel wird das vorläufige Literaturverzeichnis vorgestellt.

1.5 Grober Zeitplan

Datum	*Aufgabe*
28.04.2014	Recherche & Analyse
05.05.2014	Recherche & Zieldefinition
12.05.2014	Recherche & Einordnung
19.05.2014	Apache Storm Vorstellung & API
26.05.2014	Apache Storm Hands on & Vor- Nachteile
02.06.2014	Apache Kafka Vorstellung & API
09.06.2014	Apache Kafka Hands on & Vor- Nachteile
16.06.2014	Apache Flume Vorstellung & API
23.06.2014	Apache Flume Hands on & Vor- Nachteile
30.06.2014	Apache S4 Vorstellung & API
07.07.2014	Apache S4 Hands on & Vor- Nachteile
14.07.2014	Anwendungsfall Implementation & Messung
21.07.2014	Vorstellung & Diskussion Ergebnisse
28.07.2014	Review I
04.08.2014	Nachbearbeitung
11.08.2014	Schlussbetrachtung
18.08.2014	Review II
25.08.2014	Nachbearbeitung, Druck & Erstellung Präsentation
01.09.2014	Abgabe
08.09.2014	Kolloquium
15.09.2014	
22.09.2014	
29.09.2014	
06.10.2014	
13.10.2014	
20.10.2014	
27.10.2014	
28.10.2014	spätester Abgabetermin

1.6 Vorläufiges Literaturverzeichnis

[1] Gul Agha. *Actors: a model of concurrent computation in distributed systems*. MIT Press, Cambridge, MA, USA, 1986.

[2] A. Beckmann, S. Karabekyan, and J. Pflüger. A flexible and testable software architecture: Applying presenter first to a device server for the doocs accelerator control system of the european xfel. In *PCaPAC2012: Proceedings of the 9th International Workshop on Personal Computers and Particle Accelerator Controls*, volume 9, pages 131–133, December 2012.

[3] Dhruba Borthakur, Jonathan Gray, Joydeep Sen Sarma, Kannan Muthukkaruppan, Nicolas Spiegelberg, Hairong Kuang, Karthik Ranganathan, Dmytro Molkov, Aravind Menon, Samuel Rash, Rodrigo Schmidt, and Amitanand Aiyer. Apache hadoop goes realtime at facebook. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, SIGMOD '11, pages 1071–1080, New York, NY, USA, 2011. ACM.

[4] Surajit Chaudhuri and Umeshwar Dayal. An overview of data warehousing and olap technology. *SIGMOD Rec.*, 26(1):65–74, March 1997.

[5] Jagmohan Chauhan, Shaiful Alam Chowdhury, and Dwight J. Makaroff. Performance evaluation of yahoo! s4: A first look. In Fatos Xhafa, Leonard Barolli, and Kin Fun Li, editors, *3PGCIC*, pages 58–65. IEEE, 2012.

[6] Mitch Cherniack, Hari Balakrishnan, Magdalena Balazinska, Don Carney, Uğur Çetintemel, Ying Xing, and Stan Zdonik. Scalable distributed stream processing. In *CIDR 2003 - First Biennial Conference on Innovative Data Systems Research*, Asilomar, CA, January 2003.

[7] Allen Goldberg and Robert Paige. Stream processing. In *Proceedings of the 1984 ACM Symposium on LISP and functional programming*, LFP '84, pages 53–62, New York, NY, USA, 1984. ACM.

[8] Jan-Hinrich Hauer, Vlado Handziski, Andreas Köpke, Andreas Willig, and Adam Wolisz. A component framework for content-based publish/subscribe in sensor networks. In *Proceedings of the 5th European conference on Wireless sensor networks*, EWSN 2008, pages 369–385, Berlin, Heidelberg, 2008. Springer-Verlag.

[9] Patrick Hunt, Mahadev Konar, Flavio P Junqueira, and Benjamin Reed. Zookeeper: wait-free coordination for internet-scale systems. In *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, volume 8, pages 11–11, 2010.

[10] Ken Kennedy and Kathryn S. McKinley. Maximizing loop parallelism and improving data locality via loop fusion and distribution. In *Languages and Compilers for Parallel Computing*, volume 768 of *Lecture Notes in Computer Science*, pages 301–320. Springer-Verlag, 1994.

[11] Jay Kreps, Neha Narkhede, and Jun Rao. Kafka: A distributed messaging system for log processing. In *Proceedings of 6th International Workshop on Networking Meets Databases (NetDB), Athens, Greece*, 2011.

[12] Leonardo Neumeyer, Bruce Robbins, Anish Nair, and Anand Kesari. S4: Distributed stream computing platform. In *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, ICDMW '10, pages 170–177, Washington, DC, USA, 2010. IEEE Computer Society.

[13] Zhengping Qian, Yong He, Chunzhi Su, Zhuojie Wu, Hongyu Zhu, Taizhi Zhang, Lidong Zhou, Yuan Yu, and Zheng Zhang. Timestream: reliable stream computation in the cloud. In *Proceedings of the 8th ACM European Conference on Computer Systems*, EuroSys '13, pages 1–14, New York, NY, USA, 2013. ACM.

[14] Ariel Rabkin, Matvey Arye, Siddhartha Sen, Vivek Pai, and Michael J. Freedman. Making every bit count in wide-area analytics. In *Proceedings of the 14th USENIX conference on Hot Topics in Operating Systems*, HotOS'13, pages 6–6, Berkeley, CA, USA, 2013. USENIX Association.

[15] Shariq Rizvi. Complex event processing beyond active databases: Streams and uncertainties. Technical Report UCB/EECS-2005-26, Electrical Engineering and Computer Sciences University of California at Berkeley, December 2005.

[16] Michael Stonebraker, Chuck Bear, Uğur Çetintemel, Mitch Cherniack, Tingjian Ge, Nabil Hachem, Stavros Harizopoulos, John Lifter, Jennie Rogers, and Stan Zdonik. One size fits all? – part 2: benchmarking results. In *In CIDR*, 2007.

[17] Chengwei Wang, Infantdani Abel Rayan, Greg Eisenhauer, Karsten Schwan, Vanish Talwar, Matthew Wolf, and Chad Huneycutt. Vscope: Middleware for troubleshooting time-sensitive data center applications. In Priya Narasimhan and Peter Triantafillou, editors, *Middleware*, volume 7662 of *Lecture Notes in Computer Science*, pages 121–141. Springer, 2012.