Exposé zur Diplomarbeit
Parallelisierung von Text Mining Workflows in einer Cloud

(November 2010 – April 2011)

Betreuer: Prof. Ulf Leser
Bearbeitung: Erik Dießler

1. Motivation

Die Menge an vorhandenen Informationen, zum Beispiel in Form von Bildern, Videos, Audiodateien und textbasierten Dokumenten nimmt seit Jahren unaufhörlich zu. So stieg die Menge an digitalisierten Informationen weltweit auf 800.000 Petabytes, von denen circa 95% in unstrukturierter Form vorliegen [GR07], womit eine maschinelle Verwendung und Verarbeitung deutlich erschwert wird. Die Strukturierung dieser Daten, beispielsweise durch das Versehen mit Metainformationen, stellt eine dringende Notwendigkeit dar, um der Datenflut Herr zu werden. Unglücklicherweise liegt auch die weitaus größte Menge des vorhandenen Wissens, zum Beispiel

Forschungsberichte, Veröffentlichungen, Artikel und Studien in unstrukturierter und damit schwer zugänglicher Form vor. Jedoch ist gerade im Bereich der wissenschaftlichen Veröffentlichungen eine möglichst starke Strukturierung der Informationen für eine weitere automatisierte Verarbeitung notwendig. Diese automatisierte Verarbeitung kann durch die Verknüpfung von vorhandenem Wissen aus unterschiedlichen Quellen die Basis für neue Forschung und Erkenntnisse schaffen.

Ein Beispiel sind Sammlungen von medizinischen Texten. PubMed [PM10] ist eine bibliographische Fachdatenbank aus dem Bereich der Biomedizin. Sie umfasst circa 20 Millionen Zitate, meist inklusive der kompletten Abstracts und Links auf die Volltexte. Jährlich wächst der Umfang von PubMed um etwa 500.000 weitere Dokumente. Viele dieser Veröffentlichungen enthalten Informationen über die Beziehungen von Proteinen. Diese Zusammenhänge können textübergreifend auf Grund der Anzahl an Dokumenten jedoch schwerlich durch Lesen und Herstellen möglicher Beziehungen „per Hand“ ermittelt werden. An diesem Punkt setzt das Text Mining an, welches vor allem die Suche von Bedeutungsstrukturen und Zusammenhängen in eben diesen schwach strukturierten Texten unterstützt [AT99, YYM09]. Meist handelt es sich bei Text Mining Anwendungen um eine Kombination von verschiedenen Teilprogrammen für Information Retrieval, Natural Language Processing und Extraktions [HW06]. Diese Teilprogramme führen jeweils eine ganz spezifische Aufgabe wie zum Beispiel das Part-Of-Speech-Tagging aus. Die Kombination der Teilprogramme stellt zusammengenommen einen Workflow dar, welcher von einem Textdokument komplett durchlaufen wird. Je nach Anforderung an die Text Mining Anwendung werden unterschiedliche Komponenten zu spezifischen Workflows kombiniert. Für die Modellierung, Kombination und Ausführung von Workflows existiert eine Vielzahl von unterschiedlichen Frameworks, wie Apache UIMA [UIMA10], U-Compare [UC10, KBM09], OpenNLP [NLP10], Gate [GATE10], LingPipe [LP10] und andere.

Ein Bereich aus der Biomedizin, der hier beispielhaft genannt werden kann, ist die Suche, das Verständnis und die Katalogisierung von Protein-Protein-Interaktionen (PPI). Hierbei handelt es sich um biochemische Prozesse in lebenden Organismen, bei denen zwei oder mehr Proteine zusammen eine biologische Funktion, wie die DNA Replikation, ausführen. Das Verständnis dieser Prozesse ermöglicht den Eingriff von außen im Falle von Krankheiten wie Krebs, Alzheimer oder Diabetes.

Zusammenhänge zwischen den PPIs herstellen zu können, ist für die Erfassung dieser komplexen Prozessen unerlässlich.
Mit der steigender Anzahl an Veröffentlichungen und der damit verbundenen Vergrößerung der Datenbasis steigt der Bedarf an Zeit und Ressourcen, die bei der Verarbeitung der Texte benötigt werden. Die automatisierte, dabei aber trotzdem einfache und kostengünstige Akquirierung von zusätzlichen Ressourcen sowie die effizientere Gestaltung der automatisierten Prozesse zur Strukturierung und Suche bekommen somit eine immer größere Bedeutung. Auf die Frage nach kostengünstigen und leicht verfügbaren Ressourcen zur Abarbeitung von Workflows lautet eine der möglichen Antworten Cloud Computing [WTK08]. Das Cloud Computing stellt Ressourcen wie beispielsweise pure Rechenleistung oder umfangreiche Speichermöglichkeiten on demand zur Verfügung. Ist ein Problem parallelisierbar und verlangt keine übermäßig schnelle (mit einem Cluster vergleichbare) Anbindung zwischen den rechnenden Instanzen, ist eine Cloud Architektur eine geeignete Umgebung zu dessen Bearbeitung [EH08]. Aktuelle öffentliche Implementierungen wie Amazon EC2 [AZ07] oder Microsofts Azure [WA10] bieten diese Leistungen zu relativ günstigen Preisen an.

2. Ziele

Ziel dieser Diplomarbeit ist es, die Ausführung von Text Mining Workflows zu beschleunigen. Um dies zu erreichen, soll auf zwei Ebenen der Workflowausführung angesetzt werden. Der erste Ansatzpunkt bezieht sich auf die Parallelisierung von Workflows. Wie bereits beschrieben besteht ein Workflow aus mehreren Teilschritten, die nacheinander auf einem Textdokument ausgeführt werden. Soll dieser Workflow über einer Million Dokumente ausgeführt werden, so ist offensichtlich, dass eine Parallelisierung des Workflows zur Geschwindigkeitssteigerung beitragen wird. Die Datenbasis in Form von einzelnen Textdokumenten, welche jeweils individuell durch einen Workflow abgearbeitet werden, macht eine Verteilung besonders attraktiv. Hier bieten sich die oben bereits angesprochenen Cloud Computing Architekturen an.
Der zweite Ansatzpunkt stellt sich dem Problem einer effizienteren Ausführung von Workflows. Häufig werden Workflows mit nur leicht veränderten Parametern neu gestartet. Diese Parameteränderungen haben mitunter nur Einfluss auf einzelne

Teilschritte des gesamten Workflows, aber dennoch wird der komplette Workflow erneut ausgeführt. Hier besteht ein Einsparpotential, wenn es möglich wäre, bereits vorhandene Ergebnisse von vorherigen Workflowausführungen in die Ausführung eines neuen Workflows mit einfließen zu lassen. So könnte beispielsweise ein Workflow bereits existierende Zwischenergebnisse verwenden, um die Anzahl der Workflowschritte und damit die Laufzeit des gesamten Workflows zu reduzieren. Dieser Ansatz verspricht insbesondere bei geringfügig veränderten Parametern oder ausgetauschten Komponenten am Ende eines Workflows Einsparpotentiale.
Die Funktionsfähigkeit der zu entwickelnden Lösung soll anhand von zwei Fallstudien demonstriert werden.

3. Vorgehen

Für die Erstellung von Workflows sollen die Natural Language Processing / Text Mining Frameworks U-Compare [UC10, KBM09] und Apache UIMA [UIMA10] verwendet werden. Unstructured Information Management Architecture (UIMA) stellt einen OASIS [OA09] Standard für den Zugriff auf unstrukturierte Informationen dar. Mit Apache UIMA existiert eine Open Source Implementierung dieses Standards. Das Framework umfasst unter anderem die Entwicklung und das Management von Workflowkomponenten und kompletten Workflows. Als Alternativen können OpenNLP [NLP10], GATE [GATE10] oder LingPipe [LP10] genannt werden. Auf Grund der Standardisierung und der Existenz von Apache UIMA-Wrappern für OpenNLP und GATE soll im Rahmen dieser Arbeit Apache UIMA verwendet werden. U-Compare basiert auf UIMA und stellt eine GUI zur Erstellung von Workflows aus bereits existierenden oder selbst konzipierten auf UIMA basierenden Komponenten zur Verfügung.
Als Cloud Computing Umgebung soll auf Amazon Elastic Compute Cloud (EC2) zurückgegriffen werden. Diese bietet eine sehr gute Toolunterstützung, eine recht große Community und einen sehr guten Funktionsumfang, welcher sich von reinen Rechenleistungen über Load Balancing und verschiedenen Storagelösungen bis hin zu Datenbankdiensten erstreckt. Als Alternativen zu den Amazon Web Services können zum Beispiel die Windows Azure Plattform und GoGrid [GG10] genannt werden.

Ein weiterer Grund für den Einsatz von EC2 ist die geplante Verwendung des experimentellen Task-Scheduling Frameworks Nephele [WBE10]. Das Framework wird bisher experimentell auf Eucalyptus [EU10] eingesetzt. Eucalyptus ist eine Software, welche das Interface von EC2 implementiert und die private Erstellung einer Cloud ermöglicht. Für die Verwendung in einer nicht privaten Cloud, ist somit im EC2 Rahmen mit den geringsten Problemen zu rechen. Nephele soll das Management und die Verteilung der zu bearbeitenden Textsammlungen, das Scheduling der Teilschritte und die Überwachung der Teilergebnisse übernehmen. Eines der wichtigsten Ziele von Nephele ist die dynamische Allokierung von Ressourcen. Für jeden Teilschritt des Prozesses werden nur die gerade notwendigen Ressourcen verwendet. Aktuell ist Nephele noch nicht in der Lage, die Menge der benötigten Rechenleistung selbstständig zu bestimmen. Dies muss noch manuell erfolgen, ist aber für jeden Teilschritt möglich. Als mögliche Alternativen können Pegasus [DSS05], Dryad [IBY07] oder Falkon [RZD07] genannt werden. Alle diese Frameworks orientieren sich jedoch eher an GRID Computing und während dies der Idee nach nicht weit vom Cloud Computing entfernt ist, bietet Nephele hier doch zwei entscheidende Vorteile. Zum einen orientiert sich Nephele an aktuellen Cloud Architekturen und zum anderen wird mit dem Ansatz zur dynamischen Allokierung von Ressourcen der Struktur kommerzieller Cloud Systeme Rechnung getragen.

Die Entwicklung einer Lösung zur Erreichung der Zielstellung soll in vier Schritten erfolgen. Zunächst erfolgt eine Evaluierung der einzusetzenden Komponenten sowohl im Bezug auf ihre Einsetzbarkeit für die Problemstellung als auch auf mögliche Einschränkungen, die mit ihrer Verwendung berücksichtig werden müssen. Der zweite Schritt umfasst die Erstellung einer Wrapper-Komponente, welche eine Kommunikation zwischen Apache UIMA und Nephele ermöglicht. Diese Komponente soll Nephele in die Lage versetzen, UIMA Workflows zu lesen, Ausführungspläne zu erstellen und diese verteilt auszuführen. Hierzu wird es notwendig sein, dass Nephele die einzelnen Teilschritte eines Workflows erkennt und unter Einbeziehung von Nutzerinformationen den Ausführungsgraphen sowie die zu verwendende Anzahl von Ressourcen bestimmt. Die Wrapper-Komponente muss also UIMA Workflows „verstehen“, indem sie zum Beispiel das UIMA Austauschformat PEAR verarbeiten kann. Zusätzlich soll die Wrapper-Komponente eine Konfigurationsdatei erstellen, die der Nutzer mit zusätzlichen Informationen, wie der Anzahl zu nutzender

Instanzen pro Teilschritt des Workflows, versehen kann und welche er letztlich Nephele zur Verarbeitung übergeben kann.
Ein dritter Schritt wird sich mit der Vergleichbarkeit und effizienteren Ausführung von Workflows beschäftigen. Die Komponente soll in der Lage sein, einen neuen Workflow mit den Informationen von bereits berechneten Workflowausführungen zu vergleichen. Dies könnte derart gestaltet sein, dass ein Graph verwendet wird, um die Workflows vergleichen zu können. Der Graph würde aus einem root-Element bestehen, welches für den Text-Corpus steht. Es werden Kanten für jeden Workflowschritt und Knoten für die Zwischenergebnisse eingefügt. Soll ein neuer Workflow ausgeführt werden, kann dieser vom root-Element ausgehend mit dem Graphen verglichen werden. Besteht bis zu einem n-ten Schritt Gleichheit, so kann das entsprechende Zwischenergebnis verwendet und der Workflow verkürzt werden. Gerade die Frage der Gleichheit von Workflowschritten muss an dieser Stelle besonders betrachtet werden, denn je größer die tolerierten Abweichungen sein dürfen, desto größer ist die Wahrscheinlichkeit, dass übereinstimmende Workflows gefunden werden können.
Der vierte und letzte Schritt beschäftigt sich mit der Evaluierung der entwickelten Lösung. Es sollen besonders drei Aspekte untersucht werden:

1. Funktionsfähigkeit der Lösung mit Hilfe der benannten Fallstudien unter Berücksichtigung der folgenden Punkte demonstrieren:
    a. Die Zuverlässigkeit/Stabilität der Lösung muss geprüft werden.
    b. Die Datenintegrität im Vergleich zu nicht parallelisierten Workflows muss geprüft werden
    c. Wie hoch sind die entstehenden Kosten (Cloud-Ressourcen) bei Umsetzung der Fallstudien?

2. Die Performance der Lösung unter Berücksichtigung der folgenden Punkte messen und bewerten:
    a. Können mit der entwickelten Lösung Performancegewinne erreicht werden? Skaliert die Lösung?
    b. Welchen Einfluss hat der entstehende Verwaltungsoverhead und ab welcher Corpusgröße und/oder welchem Workflowumfang macht der Einsatz Sinn?

3. Wie wirkt sich der Einsatz der Komponente zum Vergleich von Workflows und der Wiederverwendung von Zwischenergebnissen aus?
    a. Besteht Datenintegrität bei der Nutzung von Zwischenergebnissen gegenüber der kompletten Ausführung eines Workflows?
    b. Werden vergleichbare Workflows zuverlässig erkannt?
    c. Wird durch den Vergleich und die Anwendung der Vergleichs-Komponente ein Performancegewinn erzielt?

4. Verwandte Arbeiten

Der Bereich der verwandten Arbeiten gliedert sich in zwei Bereiche, zum einen der Frage nach der verteilten Ausführung von Text Mining Workflows auf Cloud Systemen und zum anderen dem Problem der Wiederverwendbarkeit von Zwischenergebnissen der Workflows.

Zur Nutzung von Cloud Systemen konnten keine Veröffentlichungen gefunden werden. Einzig GATE Cloud [GC10] ist ein Projekt, welches im Rahmen von GATE die Verwendung von Cloud Systemen anstrebt. Aktuell befindet es sich im Alpha-Status und genauere Informationen sind lediglich für Partner und Sponsoren verfügbar. Im Bereich des Grid Computing als nächster „Verwandter“ des Cloud Computing gibt es jedoch einige Arbeiten, die sich im Sinne der Beschleunigung von Text Mining Workflows mit den Möglichkeiten des Grid Computing auseinandersetzen. So beschreiben Kumpf et al. in ihrer Arbeit [KMW07] das Mapping und die Implementierung eines Text Mining Workflows auf D-Grid zur Analyse von PubMed Texten. Andere Ansätze setzen auf bereits vorhandenen Grid Computing Projekten wie discovery net [CGG02] oder myGrid [MG10] auf und erweitern diese [GCG05]. Ghanem et al. ergänzen die Möglichkeiten, die discovery net bietet, um eine Architektur, die es ermöglicht, komplexe Data Mining und Text Mining Workflows erstellen zu können.

Die Wiederverwendbarkeit von Workflows kann aus mehreren Blickwinkeln betrachtet werden. So widmen sich Xiang und Madey in [XM07] der Frage der erneuten Verwendung von bereits erstellten Teilen von Workflows oder kompletten Workflows. Wombacher [W10] schlägt eine Lösung vor, die die Verwendung von

Zwischenergebnissen in verschiedenen Workflows ermöglicht. Und schließlich, bezogen auf eine Teilfragestellung dieser Arbeit, gibt es den Blickwinkel der „smart“ reruns von Workflows, also der Verwendung von bereits vorhandenen Zwischenergebnissen für einen erneuten Workflowdurchlauf bei veränderten Parametern oder einem fehlerhaften Abbruch. Mehrere Arbeiten [ABJ06, CA08] rund um das Projekt Kepler [K10] beschäftigen sich mit dieser Frage. Kepler ist ein open Source Framework zu Erstellung wissenschaftlicher Workflows und bietet unter anderem die Option von „smart“ reruns.

5. Literatur

[ABJ06] Ilkay Altintas, Oscar Barney, Efrat Jaeger-Frank - Provenance Collection Support in the Kepler Scientific Workflow System, In *Proceedings of the International Provenance and Annotation Workshop*, 2006

[AT99] Ah-hwee Tan - Text Mining: The state of the art and the challenges, In *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, 1999

[AZ07] Amazon Web Services - Projekt Webseite: http://aws.amazon.com/ec2/, September 2010

[CA08] Daniel Crawl, Ilkay Altintas - A Provenance-Based Fault Tolerance Mechanism for Scientific Workflows, *In Intl. Provenance and Annotation Workshop*, 2008

[CGG02] V. Ćurčin, M. Ghanem, Y. Guo, M. Köhler, A. Rowe, J. Syed, P. Wendel - Discovery net: towards a grid of knowledge discovery, *In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 658 - 663*, Edmonton, Alberta, Canada, 2002

[DSS05] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz. – Pegasus: A framework for mapping complex scientic workflows onto distributed systems, In *Scientific Programming, 13(3) 219 -237*, 2005.

[EH08] Constantinos Evangelinos and Chris N. Hill – Cloud Computing for parallel Scientific HPC Applications: Feasibility of running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2. [Extended Abstract], Department of Earth, Atmospheric and Planetary Sciences, MIT 77 Massachusetts Ave. Cambridge, MA 02139, USA, 2008

[EU10] Eucalyptus, *Projekt Webseite: http://open.eucalyptus.com/*, September 2010

[GCG05] Ghanem, M.; Chortaras, A.; Guo, Y.; Rowe, A.; Ratcliffe, J. - A grid infrastructure for mixed bioinformatics data and text mining, In *The 3rd ACS/IEEE International Conference on Computer Systems and Applications*, 2005.

[GATE10] GATE – General Architecture for Text Engineering, *Projekt Webseite: http://gate.ac.uk/*, September 2010

[GC10] GATE Cloud – a New Way to Mine the Web, *Projekt Webseite: http://gatecloud.net/*, September 2010

[GG10] GoGrid, *Projekt Webseite: http://www.gogrid.com/*, September 2010

[GR07] John Gantz, David Reinsel et al, The Expanding Digital Universe - A Forecast of Worldwide Information Growth Through 2010, An IDC White Paper - sponsored by EMC, März 2007

[HW06] U. Hahn & J. Wermter, Levels of Natural Language Processing for Text Mining, In *S. Ananiadou & J. McNaught (Eds.), Text Mining for Biology and Biomedicine.* Boston, London: Artech House Books, pp.13-41, 2006

[IBY07] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly – Dryad: distributed data-parallel programs fromsequential building blocks. In *EuroSys '07: Proceedings of the 2nd ACM SIGOPS/EuroSysEuropean Conference on Computer Systems 2007,pages 59- 72*, New York, NY, USA, 2007

[K10] The Kepler Project, *Projekt Webseite: https://kepler-project.org/*, September 2010

[KBM09] Yoshinobu Kano, William A. Baumgartner Jr., Luke McCrohon, Sophia Ananiadou, K. Bretonnel Cohen, Lawrence Hunter and Jun'ichi Tsujii, U-Compare: share and compare text mining tools with UIMA, In *Bioinformatics. 25(15), pp. 1997-1998, doi:10.1093/bioinformatics/ btp289*, 2009

[KMW07] Kumpf, Kai; Mevissen, Theo; Wäldrich, Oliver; Ziegler, Wolfgang; Ginzel, Sebastian; Weuffel, Thomas - Multi-Cluster Text Mining on the Grid using the D-Grid UNICORE environment, *Präsentiert auf German e-Science Conference*, 2007

[LP10] LingPipe, *Projekt Webseite: http://alias-i.com/lingpipe/*, September 2010

[MG10] myGrid, Projekt *Webseite: http://www.mygrid.org.uk/*, September 2010

[NLP10] OpenNLP, *Projekt Webseite: http://opennlp.sourceforge.net/*, September 2010

[OA09] OASIS - Organization for the Advancement of Structured Information Standards, *Projekt Webseite: http://www.oasis-open.org/news/oasis-news-2009-03-19.php*, September 2010

[PM10] PubMed, *Projekt Webseite: http://www.ncbi.nlm.nih.gov/pubmed*, September 2010

[RZD07] I. Raicu, Y. Zhao, C. Dumitrescu, I. Foster, and M. Wilde – Falkon: a Fast and Light-weight tasKexecutiON framework. In *SC '07: Proceedings of the 2007 ACM/IEEE conference on Supercomputing, pages 1-12*, New York, NY, USA, 2007.

[UIMA10] Apache UIMA – Unstructured Information Management Architecture, *Projekt Webseite: http://uima.apache.org/index.html*, September 2010

[WA10] Windows Azure Plattform, *Projekt Webseite: http://www.microsoft.com/windowsazure/*, September 2010

[WBE10] Daniel Warneke, Dominic Battré, Stephan Ewen, Fabian Hueske, Odej Kao, Volker Markl – Nephele/PACTs: A Programming Model and Execution Framework for Web-Scale Analytical Processing Pages: 119-130 , In *Proceedings of the 1st ACM symposium on Cloud computing*, Indianapolis, Indiana, 2010

[W10] Andreas Wombacher - Data Workflow - A Workflow Model for Continuous Data Processing, *Internal Report*, University of Twente, Electrical Engineering, Mathematics and Computer Science, 2010

[WTK08] Lizhe Wang, Jie Tao, Marcel Kunze, Alvaro Canales Castellanos, David Kramer, Wolfgang Karl. Scientific Cloud Computing: Early Definition and Experience. In *proceedings of 10th IEEE International Conference on High Performance Computing and Communications (HPCC'08)*, Dalian, China, Sep. 2008.

[UC10] U-Compare, *Projekt Webseite: http://u-compare.org/index.html*, September 2010

[XM07] Xiaorong Xiang, Gregory Madey - Improving the Reuse of Scientific Workflows and Their By-products, In *IEEE International Conference on Web Services, 2007*, pages 792 - 799, Salt Lake City, 2007

[YYM09] Yanliang Qi, Yang Zhang, Min Song, "Text Mining for Bioinformatics: State of the Art Review," In *Computer Science and Information Technology, International Conference on, pp. 398-401*, 2009