Evaluation von NoSQL-Datenbanksystemen für den Einsatz in einem Framework zur Analyse großer unstrukturierter Textmengen

Max Kießling

Ausgangslage & Problemstellung

Mit der immer größer werdenden Menge frei verfügbarer Daten im Internet, steigt auch die Möglichkeit diese Daten unter verschiedensten Gesichtspunkten zusammenzufassen und auszuwerten, um so zu neuen Erkenntnissen zu gelangen. Aufgrund der schieren Menge verfügbarer, sowie benötigter Daten, ist es kaum möglich die Verarbeitung manuell zu betreiben. Daher helfen immer fortschrittlichere Algorithmen automatisch Zusammenhänge innerhalb dieser, meist unstrukturierten, Daten zu erkennen und neues Wissen zu extrahieren.
Diese Systeme benötigen jedoch eine flexible Basis, welche die Akquise und Bereitstellung der benötigten Informationen betreut.
Ein solches System ist das auf *Ruby* basierende *Ciwor*. Ciwor kümmert sich dabei um den kompletten Verarbeitungsprozess. Es verwaltet die Datenquellen, ruft deren Inhalt in dynamischen Zeitintervallen ab und verarbeitet diesen (zum Beispiel mittels *Named Entity Recognition*). Die aktuelle Implementierung von Ciwor nutzt dabei zur Datenhaltung ausschließlich das relationale SQL-*Datenbanksystem* (DBS) MySQL.
Allerdings ergeben sich aus der Nutzung des relationalen Datenbanksystems einige Probleme.
So wächst das Volumen der gespeicherten Daten täglich um etwa 350MB. Dies stellt prinzipiell keine problematischen Anforderungen an das Datenbanksystem, da moderne relationale DBS gut skalierbar sind und daher auch mit großen Datenvolumina umgehen können. Durch ihre stark strukturierte Datenhaltung, sind sie jedoch nicht dafür geeignet große Mengen unstrukturierter Inhalte zu speichern und effizient zu verwalten.
Ein weiteres Problem entsteht beim Auffinden und der Modellierung komplexer Zusammenhänge zwischen Entitäten. Zwar bietet SQL theoretisch gute Ansätze um einfach 1:n und n:m Beziehungen zwischen Relationen zu modellieren. Sobald diese Verknüpfungen vielschichtiger werden, steigt die Komplexität der benötigten Datenstrukturen sehr stark an. Dies führt wiederum zu sehr umfangreich und schwierig zu formulierenden Datenbankabfragen. Gleichermaßen steigt auch deren Berechnungsdauer. So benötigen einige der von uns durchgeführten Anfragen bis zu 20 Minuten Rechenzeit.

Aufgabenstellung

Betrachtet man die momentane Situation, zeigt sich, dass eine Neugestaltung des Systems unumgänglich ist. Aufgabe dieser Arbeit ist daher eine Überarbeitung des Systems CIWOR. Dabei soll eine modulare Software entstehen, welche die Auswertung vorhandener Daten einfacher und effizienter gestaltet. Hierbei steht vor allem die Art der Datenhaltung im

Vordergrund. Verschiedene moderne *NoSQL* Systeme, wie *Neo4J* und *MongoDB* sollen auf ihre Eignung für die Projektanforderungen hin untersucht werden. Anhand dessen kann anschließend entschieden werden, welche Systeme eingesetzt werden sollen. Weitere wichtige Anforderungen an das Design sind die Möglichkeit der Anbindung an die vorhandenen Strukturen, sowie die Interoperabilität der Komponenten.

Abschließend soll das Konzept testweise implementiert und mit dem Momentanzustand im Hinblick auf Geschwindigkeit, Skalierbarkeit und Bedienbarkeit verglichen werden.

Umsetzungskonzept

Betrachtet man die aktuellen Probleme, so fällt auf, dass Ciwor drei Aufgaben erfüllt:

- Quellenverwaltung, Crawling und Jobmanagement
- Speicherung und Verwaltung der Rohdaten
- Datenverarbeitung und Informationsextraktion

Da all diese Aufgabenbereiche unterschiedliche Anforderungen an das System stellen, ist es sinnvoll sie modular und individuell zu konzipieren und umzusetzen.
Das Kernsystem (Quellverwarltung, Crawling, Jobmanagement) kann unverändert bleiben, weil die bisherige Implementierung den Anforderungen sehr gut gerecht wird. Die Speichereinheit und die Verarbeitungseinheit müssen von der jetzigen Anwendung entkoppelt und den Anforderungen angepasst werden.

Speicherung und Verwaltung der Rohdaten

Dieser Bereich der Anwendung dient dazu, die gecrawlten Rohdaten, sowie vorverarbeitete Datensätze zu verwalten. Dabei handelt es sich zumeist um unstrukturierte Inhalte, weshalb es vermutlich sinnvoll ist einen Dokumentenstore wie MongoDB oder aber das Filesystem für die Datenhaltung zu nutzen.

Datenverarbeitung und Informationsextraktion

Im letzten Schritt des Verarbeitungsprozesses steht die Aufbereitung und Strukturierung der gefundenen Daten. Je nach Art der Daten bzw. der zu extrahierenden Information, gibt es eine große Anzahl verschiedener, zum Teil aufeinander aufbauender, Verarbeitungsschritte. Daher ist es wichtig, dass es einfach ist neue Prozesse, sogenannte *Worker,* zu implementieren und am System anzumelden. Da die Ergebnisse stark von den extrahierten Daten abhängen, muss die Datenhaltung für die Worker individuell betrachtet werden. Es ist jedoch davon auszugehen, dass es sich zu meist um strukturierte, stark zusammenhängende Daten handelt, wo durch sich eine Datenhaltung in Graphdatenbanken wie Neo4J empfielt.