Analyse von Log-Files eines Web-Servers mithilfe eines Rechnerverbunds

Steffen Hafner und Daniel Landler-Gärtner

Hochschule
Ravensburg-Weingarten
Technik | Wirtschaft | Sozialwesen

PROJEKTARBEIT

Systemadministration

in der Hochschule Ravensburg-Weingarten

im April 2017

Erklärung

Wir erklären eidesstattlich, dass wir die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

der Hochschule Ravensburg-Weingarten, am 5. April 2017

Steffen Hafner und Daniel Landler-Gärtner

Inhaltsverzeichnis

Erklärung i

1 Einleitung 1

2 Problem 2
2.1 Webserver Log-File Analyse . . . 3
2.2 Log-Files des HRW Webserver . . . 3
2.3 Geplante Umsetzung . . . 4

3 Anforderungsanalyse 5

4 Lösungsvorschläge 6
4.1 Framework . . . 6
4.1.1 Hadoop . . . 6
4.1.2 Spark . . . 7
4.2 Datenbank . . . 8
4.2.1 HBase . . . 8
4.2.2 Cassandra . . . 8
4.3 Virtualisierungssoftware . . . 9
4.3.1 Docker . . . 9
4.3.2 VirtualBox . . . 9

Kapitel 1

Einleitung

Die Verfügbarkeit von Daten hat sich in den vergangenen zehn Jahren drastisch verändert. Die Anzahl verschiedener Datenquellen steigt stetig durch die zunehmende Verbreitung mobiler und internetfähiger Geräte. Dadurch sehen sich Unternehmen heute mit sehr viel größeren Datenmengen konfrontiert. Diese gilt es zu erfassen, zu speichern und auszuwerten. Dabei ist es nicht nur die Datenmenge selbst, die den Unternehmen Probleme bereitet, sondern darüber hinaus auch die Struktur und die Art der Daten, sowie die Geschwindigkeit, mit der sie anfallen.

Der Begriff Big-Data ist in den letzten Jahren vom reinen Buzz-Word hin zu einem greifbaren technischen Begriff gereift. Big-Data sind Datenmengen, die zu groß für traditionelle Datenbanksysteme sind sowie eine hohe Schnelllebigkeit besitzen. Diese Datenmengen sind entweder unstrukturiert oder semi-strukturiert und enstsrpechen somit nicht den Richtlinien herkömmlicher Datenbanksysteme. Die Herausforderung liegt darin die Daten dennoch zu speichern und zu verarbeiten, damit neue Informationen gewonnen werden können. Mögliche neue Informationen sind z.B. empfohlene Kontakte in sozialen Netzwerken, passende Produktempfehlungen in E-Commerce Lösungen oder Artikelvorschläge auf Nachrichtenseiten.

Eine treffende Definition von Big-Data lässt sich am Besten durch die drei V veranschaulichen. Volume (Speichergröße und Umfang), velocity (die Geschwindigkeit mit der Datenmengen generiert und transferiert werden) und variety (Bandbreite der Datenquellen)[1]. Diese Definition kann durch value und validity ergänzt werden, welche für einen unternehmerischen Mehrwert und die Sicherstellung der Datenqualität stehen[2].

Auf Grund des hohen Aufwands von Echtzeit-Analysen großer Datenmengen wird Big-Data häufig in Verbindung zu Cluster Computing gebracht. Um Cluster Computing zu realisieren ist ein Framework nötig um die Verarbeitung der Daten auf eine große Anzahl an Computern zu verteilen.

Die Visualisierung der gewonnen Informationen erfordert die Bildung von Korrelationen zwischen den einzelnen Datensätzen, um diese in Abhängigkeit voneinander präsentieren zu können. Dies erfordert, im Gegensatz zu normalisierten Daten von relationalen Datenbanken, bei Plain-Text-Analysen einen erheblichen Mehraufwand.

[1] Gartner IT Glossary: http://www.gartner.com/it-glossary/big-data

[2] Big Data – Fluch oder Segen? – Unternehmen im Spiegel gesellschaftlichen Wandels

Kapitel 2

Problem

Bereits vor dem Aufkommen von Big-Data haben Unternehmen Log-Files zur Gewinnung von Einblicken genutzt. Jedoch ist das Problem, dass durch das exponentielle Wachstum aller Datenquellen die Verwaltung und Analyse der Log-Files zu einer immer größeren Herausforderung wird. Log-Files enthalten eine große Menge an Informationen, wobei nicht alle für den jeweiligen Betreiber von gleicher Bedeutung sind. Zudem liegen die Informationen innerhalb der Log-Files in einem schlecht leserlichen Format vor, weshalb eine Analyse von relevanten Informationen und Korrelationen sehr aufwendig ist.

Serverlogs können abhängig von dem Log-Level der jeweiligen Architektur sehr groß ausfallen, wodurch die manuelle Verwaltung und Analyse nahezu unmöglich wird. Andererseits ist die Analyse von System-Logs notwendig um beispielsweise potenzielle Sicherheitsrisiken und Netzwerkfehler erkennen zu können.

Es gibt prinzipiell zwei Arten von Log-Files:

1. **Ereignis-Logs** – ermöglichen einen umfassenden Überblick über die Funktionalität des Systems sowie allen Komponenten zu einem bestimmten Zeitpunkt.
2. **Benutzer-Logs** – ermöglichen einen detailierten Einblick in das Nutzerverhalten wie z.B. auf Webseiten. Durch die Analyse der Benutzer-Logs können genauere Informationen über das Verhalten der Benutzer gewonnen werden als mit gewöhnlichen Webanalyse-Diensten wie Google Analytics oder Omniture.

Ein effizienter und automatisierter Prozess wird benötigt damit schnell und akkurat Muster erkannt werden können sowie die großen Datenmengen der Serverlogs erfolgreich bewältigt werden können. Andererseits laufen Unternehmen Gefahr wertvolle Informationen in der riesen Datenflut zu verlieren und dadurch einen datengestützen Wettbewerbsvorteil zu verlieren.

2.1 Webserver Log-File Analyse

Dig Logfile-Analyse bezeichnet den Prozess der gezielten Überprüfung und Auswertung eines Logfiles. Durch die Auswertung von Webserver Logs können allgemeine Informationen über das Verhalten und die Aktivitäten der Seitenbesucher gewonnen werden.

Webserver Log-Files enthalten folgende Informationen:

- IP-Adresse und Hostname
- Zugriffszeitpunkt
- Vom User verwendeter Browser
- Vom User verwendetes OS
- Herkunftslink bzw. -URL
- Verwendete Suchmaschine inklusive genutzter Keywords
- Verweildauer
- Anzahl aufgerufener Seite
- Zuletzt geöffnete Seite vor dem Verlassen der Webseite

Eines der größten Probleme der Webserver-Logfile-Analyse wird durch das zustandslose HTTP Protokoll verursacht. Durch die separate Behandlung der Anfragen, behandelt der Webserver zwei verschiedene Seitenaufrufe eines Clients als zwei unterschiedliche Instanzen. Wodurch eine Analyse des Nutzerverhaltens deutlich erschwert wird.

Um diesen Problemen entgegen zu wirken gibt es zwei gängige Lösungsmöglichkeiten:

1. **Vergabe einer Session-ID:** Die Session-ID ist eine serverseitig generierte ID, die im Browser des Nutzers gespeichert wird. Alle folgenden Anfragen eines Nutzers werden durch die vergebene ID kenntlich gemacht.
2. **Nutzeridentifikation via IP-Adresse:** Nutzer werden über ihre eindeutige IP-Adresse erkannt und bei allen folgenden Anfragen durch diese identifiziert. Voraussetzung dafür ist die Zustimmung des Nutzers zur Erhebung seiner vollständigen IP-Adresse zu Analysezwecken. Ein weiteres Problem ergibt sich aus der dynamischen Vergabe von IP-Adressen oder durch die mehrfache Nutzung der gleichen IP-Adresse.

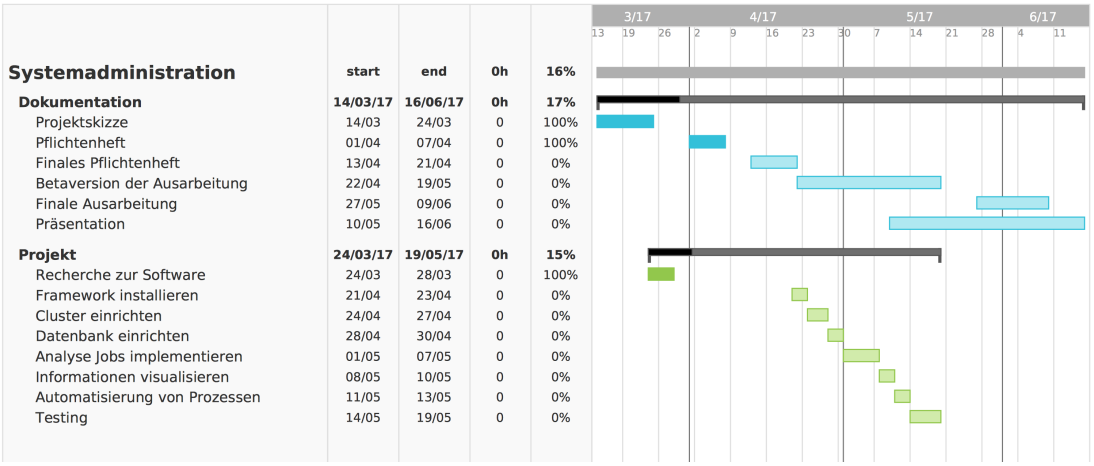
2.2 Log-Files des HRW Webserver

Für die Veranschaulichung des genannten Problems verwenden wir die Log-Files des Webservers der Hochschule Ravensburg-Weingarten. Zum aktuellen Zeitpunkt liegen uns die gewünschten Log-Files noch nicht vor, da dass Rechenzentrum mit dem Datenschutzbeauftragten noch in Rücksprache ist. Daher können wir auf den Inhalt und die geplante Analyse der Log-Files nicht detailliert eingehen. Einige Beispiele dafür wären:

- Welchen Browser wurde benutzt?
- Auf welcher Seite stand der Link, mit dem der Nutzer auf die Seite gekommen ist?
- Welche Suchmaschine wurde verwendet?
- Wie unterscheidet sich Surfverhalten von externen und internen Nutzern?
- Wie lang blieb Nutzer auf der Website?
- Korrelationen zwischen den Hochschul-Websiten, wie z.B. LSF, elearning etc.

2.3 Geplante Umsetzung

Aufgrund der hohen Komplexität eines realen Computer-Clusters soll in diesem Projekt ein Prototyp entstehen, der ein virtuelles Computer-Cluster simuliert und dadurch die möglichen Funktionen eines realen Computer-Clusters veranschaulicht. Zudem soll der Prototyp einfach portierbar, skalierbar und nutzbar sein.



Systemadministration	start	end	0h	16%
Dokumentation	**14/03/17**	**16/06/17**	**0h**	**17%**
Projektskizze	14/03	24/03	0	100%
Pflichtenheft	01/04	07/04	0	100%
Finales Pflichtenheft	13/04	21/04	0	0%
Betaversion der Ausarbeitung	22/04	19/05	0	0%
Finale Ausarbeitung	27/05	09/06	0	0%
Präsentation	10/05	16/06	0	0%
Projekt	**24/03/17**	**19/05/17**	**0h**	**15%**
Recherche zur Software	24/03	28/03	0	100%
Framework installieren	21/04	23/04	0	0%
Cluster einrichten	24/04	27/04	0	0%
Datenbank einrichten	28/04	30/04	0	0%
Analyse Jobs implementieren	01/05	07/05	0	0%
Informationen visualisieren	08/05	10/05	0	0%
Automatisierung von Prozessen	11/05	13/05	0	0%
Testing	14/05	19/05	0	0%

Abbildung 2.1: Gantt-Diagramm

Kapitel 3

Anforderungsanalyse

Die geplanten Anforderungen sind aufgeteilt in *Muss* und *Kann* Kriterien.

Muss-Kriterien

- Verarbeitung erfolgt in virtuellem Cluster
- Open-Source Framework für skalierbare und verteilt arbeitende Software
- Persistente Speicherung der analysierten Daten in einer Datenbank
- Open-Source Datenbank, die mit gewähltem Framework kompatibel ist
- Gewonnen Informationen sollen Nutzer in einer Textdatei zur Verfügung stehen
- Prototyp soll plattformunabhängig und ohne aufwendige Konfiguration/Installation testbar sein
- Prototyp soll in isolierter Umgebung arbeiten, sodass keine Manipulation an Hostsystem stattfindet

Kann-Kriterien

- Gewonnene Informationen werden auf einer Website dargestellt
- Automatisierte Ausführung der Analyse-Jobs innerhalb des Clusters
- Automatisierte Instanziierung/Installation des Prototypen auf anderen Rechnern

Kapitel 4

Lösungsvorschläge

4.1 Framework

Gesucht ist ein Framework mit dem alle *Muss-Kriterien* bestmöglich umgesetzt werden können. Des Weiteren sollen alle Tasks im Gantt-Diagramm 2.1 fristgerecht erledigt werden können. Um das Projekt im genannten Zeitraum durchführen zu können, beschränken wir uns bei der Suche auf die bekanntesten Frameworklösungen auf dem derzeitigen Markt.

Die zwei bekanntesten Framworks für skalierbare, verteilt arbeitende Software im Zusammenhang mit großen Datenmengen sind *Hadoop* und *Spark*. Sowohl Hadoop als auch Spark werden unter Linux entwickelt und verwenden native Linux Libraries. Daher begrenzt sich unsere Auswahl des zu verwendenden Betriebssystems auf Linux Distributionen. Dies erleichtert zum einen das Einrichten und zum anderen die Wartung des Frameworks.

4.1.1 Hadoop

Hadoop ist ein Java-Framwork der Apache Software Foundation zum verteilten Speichern von Daten und zu deren parallelen Verarbeitung. Hadoop wird dabei in einem horizontal skalierbaren Cluster betrieben, das auf einfachstem Weg wie gewünscht skaliert werden kann. Große Unternehmen wie Yahoo betreiben so Cluster mit über 4000 Knoten[1]. Statt der Anschaffung neuer, schnellerer Hardware (Scale Up) wird beim Betrieb von Hadoop vielmehr die Erweiterung des Clusters (Scale Out) um weitere Knoten empfohlen.

Zu den Basiskomponenten von Hadoop, die bei der Installation mitgeliefert werden gehören:

- **Hadoop Distributed File System (HDFS):** Ein über das gesamte Cluster verteiltes Dateisystem zur Speicherung der zu verarbeitenden Daten.
- **Map-Reduce:** Ein Programmierframework zur verteilten Verarbeitung von Daten gemäß der zweiphasigen Verarbeitung durch Mapper- und Reducer-Klassen.
- **YARN:** Verwaltet die Resourcen eines Clusters dynamisch für verschiedene Jobs.

[1] Referenzzahlen für Unternehmen, die Hadoop einsetzen: http://wiki.apache.org/hadoop/PoweredBy

Zudem verfügt Hadoop über ein großes Ökosystem, das zahlreiche Technologien enthält, die ergänzend zu den genannten Technologien, installiert werden können.

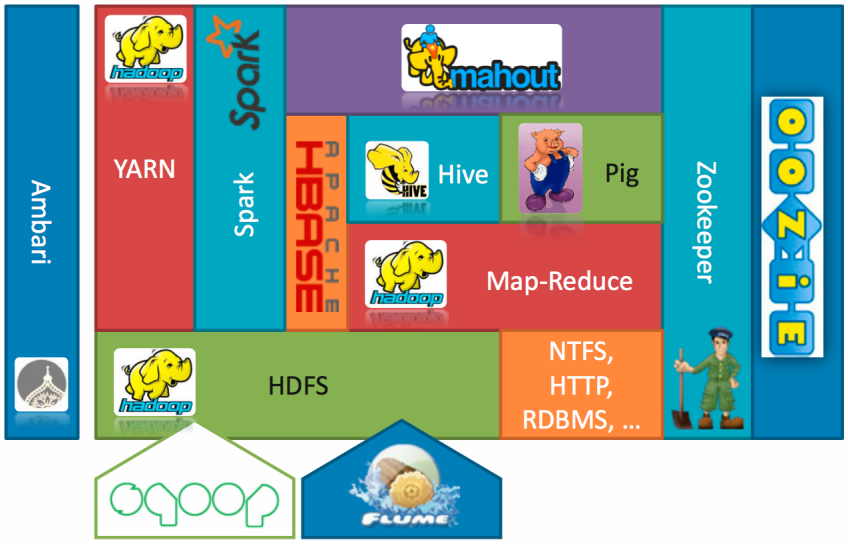


Abbildung 4.1: Hadoop-Ökosystem

4.1.2 Spark

Die Daten-Analyse Plattform Spark für clustergestützte Berechnungen wird hauptsächlich für die schnelle Ausführung von Jobs genutzt. Mit Apache Spark können Daten transformiert, fusioniert sowie mathematischen Analysen unterzogen werden. Spark ist darauf ausgelegt die Daten dynamisch im RAM des Server-Clusters zu halten und dort zu verarbeiten. Die sogenannte In-Memory-Technologie gewährleistet eine extrem schnelle Auswertung riesiger Datenmengen.

Die besondere Stärke ist das beinhaltete maschinelle Lernen (Machine Learning) mit den Zusätzen MLib (Machine Learning Bibliothek) sowie SparkR(Direkte Verwendung von R-Bibliotheken unter Spark). Dadurch lassen sich iterative Schleifen sehr gut verarbeiten was eine wichtige Vorrausetzung für Machine Learning Algorithmen darstellt.

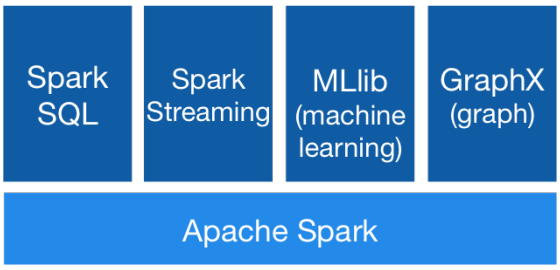


Abbildung 4.2: Apache Spark Framework

4.2 Datenbank

4.2.1 HBase

Apache HBase ist eine quelloffene, spaltenorientierte NoSQL-Datenbank, die sich in ihrer Architektur und ihrem Aufbau an Google's *BigTable* orientiert. Eine Besonderheit von HBase ist, dass ein Datensatz beliebig viele Spalten haben kann, auch mehr oder weniger als der vorige oder folgende Datensatz. Diese Eigenschaft hilft bei der schnellen persistenten Speicherung von Daten, da diese zuvor nicht normalisiert werden müssen. HBase ist Teil des Hadoop-Ökosystems 4.1 und kann daher im *verteilten Modus* ausgeführt werden, in dem sie auf Hadoop aufsetzt und das HDFS nutzt, um ihre Daten darin zu speichern. Der Vorteil beim verteilten Speichern von Daten liegt wie auch beim HDFS darin, besonders große Datenmengen unterzubringen und auf Wunsch zu skalieren, indem man weitere Knoten dem Cluster hinzufügt, wenn Performance oder Speicherkapazitäten knapp werden. Bekannte Unternehmen, die HBase verwenden sind: Adobe, Facebook, Netflix, Spotify, Yahoo! uvm.

4.2.2 Cassandra

Apache Cassandra ist ebenfalls eine spaltenorientierte NoSQL-Datenbank, die als skalierbares, ausfallsicheres System für den Umgang mit großen Datenmengen in verteilten Systemen (Clustern) konzipiert wurde. Sie entstand ebenfalls nach dem Vorbild von Google's *BigTable*. Cassandra wird häufig zusammen mit dem Framework Spark verwendet und bildet mit zusätzlichen Technologien den SMACK-Stack, bestehend aus **S**park, **M**esos, **A**kka, **C**assandra und **K**afka. Daher wird Cassandra eher mit den genannten Technologien verwendet und harmoniert weniger gut mit Hadoop. Bekannte Unternehmen, die Cassandra verwenden sind: Twitter, Digg und Reddit. Auch Facebook nutzte bis 2011 Cassandra, bis diese durch eine Kombination von HBase und HDFS ersetzt wurde.

4.3 Virtualisierungssoftware

4.3.1 Docker

Die Open-Source-Technologie Docker ermöglicht es Anwendungen mithilfe von Betriebssystemvirtualisierung in Containern zu isolieren. Durch die Verwendung von Containern können die Applikationen erstellt, ausgeführt, getestet und verteilt werden.

Durch das Verpacken von Anwendungen in standardisierte Einheiten erfolgt eine Trennung sowie Verwaltung der auf dem Rechner verwendeten Ressourcen. Diese Einheiten enthalten alle Komponenten die eine Software für die Ausführung benötigt wie z.B. Code, Laufzeitmodul, System-Tools und Systembibliotheken. Dadurch ermöglicht Docker eine schnelle, zuverlässige und einheitliche Bereitstellung von Anwendungen - unabhängig von der Umgebung.

Die Isolation der Anwendungen erfolgt bei Docker auf Software Ebene.

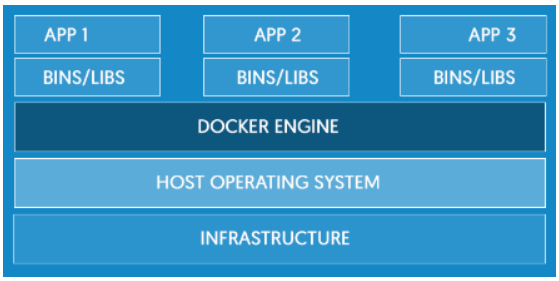


Abbildung 4.3: Aufbau Docker

4.3.2 VirtualBox

Die Virtualisierungssoftware von Oracle ermöglicht die Installation von virtuellen Maschinen auf einem Host System. Dabei greift VirtualBox auf die Hardware-Ressourcen des Hostsystems zurück und stellt einen Teil dem Gastsystem zur Verfügung. Die Isolation der Virtuellen Maschinen erfolgt auf Hardware Ebene. Jede Virtuelle Maschine besitzt somit ein eigenes Betriebssystem.

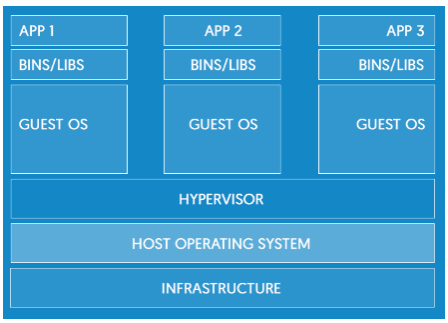


Abbildung 4.4: Aufbau VirtualBox