Entwicklung eines Realtime Webserver Monitoring Systems

Vorwort

Durch meine aktuelle Tätigkeit als System Administrator auf freiberuflicher Basis (SkyLime GbR) betreue ich sehr viele Web-Cluster. Unter anderem fällt hier die komplette Download Server Infrastruktur der Avira Operations GmbH & Co. KG.

Ein einzelner Web-Server im Cluster-Verbund verarbeitet meist mehrere tausende Verbindungsanfragen pro Sekunde. Diese Verbindungen werden in Protokolldateien vermerkt.

Für diese Web-Cluster ist eine Protokolldatei-Analyse für System Administratoren meist unerlässlich. Der System Administrator muss unter anderem Entscheiden ob weitere Server im Cluster-Verbund benötigt werden oder ein DDOS-Angriff auf die Server stattfindet.

Problem

Wie im Vorwort beschrieben, erstellt jeder Web-Server im Cluster-Verbund eine Protokolldatei. Diese Dateien enthalten eine Vielzahl wichtiger Informationen.

Aufwendiges Data-Mining kann Stunden, Tage oder Wochen dauern.

Um die aktuelle Auslastung des Clusters zu überwachen existieren Tools wie Munin, MRTG und Cacti. Diese Tools liefern jedoch nur Werte für einzelne Server und meist mit einer Verzögerung von 5 Minuten.

Hierbei geht für den System Administrator kostbare Zeit verloren um schnellstmöglich auf Probleme reagieren zu können.

Thesis

In einer Bachelor Thesis soll ein System entwickelt werden, welches Daten wie:

- Anzahl der Anfragen
- Anzahl der Anfragen je HTTP Status Code (200, 320, 4xx, 5xx)
- Dauer bis Anfrage beantwortet (Best, Worst, Average und Standard deviation)
- übertragene Bytes pro Anfrage (Min, Max, Average und Standard deviation)

von den einzelnen Servern abruft und aggregiert Darstellt.

Die Daten sollen nicht älter als zwei Sekunden sein (evtl. im Millisekundenbereich). Eine so hohe Auflösung ermöglicht ein schnelles reagieren bei z.B. einem unerwartet auftretenden Ansturm vieler Besucher.

Hierzu soll eine Architektur entworfen werden, welche die Daten von den Webservern zu einem Zentralen Monitoring System überträgt. Des weiteren soll eine Weboberfläche programmiert werden auf der die gesammelten Daten grafisch präsentiert werden. Um die Daten an den Browser zu senden sollen Websockets eingesetzt werden.