**Technische Hochschule Brandenburg**
University of Applied Sciences
**Fachbereich Informatik und Medien**

# Exposé zur Bachelorarbeit

"Einfluss von Datenstrukturen auf die Performanz des Cachings mit Redis"

von

Klaus Schwarz

| | |
|---|---|
| Datum: | Mai 2017 |
| Fachbetreuer: | Prof. Dr.-Ing. Sven Buchholz |
| Themengeber: | Technische Hochschule Brandenburg |
| Studienbereich: | Informatik und Medien |
| Matrikelnummer: | 20140006 |

# Motivation

In den letzten Jahren war eine Verschiebung vom Supercomputer-Computing hin zum Cloud-Computing zu beobachten. Infolgedessen hat sich die horizontale Skalierbarkeit erhöht, aber auch die Komplexität ist gewachsen. In großen verteilten Systemen reicht es nicht mehr aus, das Caching von hoch frequentierten Daten in einzelne Anwendungen oder Systeme einzubetten. Da zukünftige Datenanfragen aus der horizontal verteilten Cloud aus allen Richtungen kommen können und keinem Determinismus unterliegen, bedarf es eines Objektcaches.

Ein Objekt-Cache ist eine dedizierte Anwendung, die für die Speicherung von teuren Berechnungsergebnissen verantwortlich ist. Zum Beispiel kann in einer Anwendung zur Textanalyse ein Wort besonders viele Querreferenzen besitzen. Die Berechnung und Indexierung dieser Referenzen hat bereits bei der Erstellung einiges an Rechenzeit und Leistung gekostet. Eine erneute Referenzierung bei jeder Anfrage würde nicht nur unnötig Last erzeugen, es könnte große Analysen zeitmäßig schlichtweg unmöglich machen. Stattdessen wird die Berechnung nur ein einziges mal ausgeführt und das Ergebnis im Objektcache gespeichert. Nachfolgende Anfragen rufen die Ergebnisse aus dem Cache ab und vermeiden die Kosten der Berechnung.

# Aufgabenstellung

Das Hauptaugenmerk dieser Arbeit liegt nun in den Datenstrukturen, mit denen diese Objektcaches umgesetzt werden und deren Einfluss auf die Performanz in modernen verteilten Systemen. Dazu wird in dieser Arbeit die In-Memory-Datenbank Redis wie beschrieben genutzt um zu evaluieren, ob und wie sich die verfügbaren Strukturen in Diskrepanz zur Performanz auf das Caching auswirken. Durch das Aggregieren großer Mengen einfacher zu cachender Daten, soll durch das sukzessive Verändern der Datenstruktur selbiger herausgefunden werden, wie sich im Verhältnis dazu die Performanz in Form der Antwortzeit auf eine möglichst hohe Anfrage-Dichte ändert und ob es dabei eine Relation zur Speicherbreite der jeweiligen Struktur gibt.

# Erwartetes Ergebnis

Da Redis kein reiner Key-Value Store ist, kann es zu großen Unterschieden in der Performanz der einzeln genutzten Datenstruktur kommen. Zwar werden einzelne Datensätze streng nach dem Prinzip eines Key-Value Stores abgespeichert, die interne Umsetzung variiert aber stark. So unterstützt Redis die Speicherung zum Beispiel in reinen binary-safe Strings ohne weitere Indexierung, Sortierung oder Score Wert. Es ist anzunehmen, dass eine derartige Struktur eine sehr gute Performanz bei der Verarbeitung der Daten während der Aggregation erreicht. Bezweifelt werden kann jedoch, dass so gespeicherte Daten in der Masse besonders schnell durch redis gefunden werden. Somit sollte sich für diese Datenstruktur eine sehr schlechte Performanz bei der Antwortzeit einer Anfrage ergeben und das, obwohl es sich dabei um die kleinst mögliche Struktur handelt, die Redis zu bieten hat. Weiterhin ist zu erwarten, dass sogenannte "sorted Sets", bei denen es sich um Listen handelt die mit einem Score Wert versehen wurden bei bestimmten Anfragen, nämlich solchen, bei dem die Macht der Score Zahl ausgespielt werden, kann eine sehr gute Performanz abliefern ungleich zu ihrer Größe im Vergleich zu reinen Strings. Allgemein wird erwartet, dass sich alle in Redis möglichen Strukturen relativ zur O'Notation jener Datenstruktur verhalten nach deren Vorbild sie implementiert wurden. Allgemein auszuschließen ist, dass sich durch das Ändern der Datenstruktur eine Änderung der Antwortzeit ergibt, welche sich fest in Relation zur Speichernutzung der jeweiligen Struktur verhält. Da sich die Speicherbreite je Struktur stark unterscheidet könnte eine ebenso stark schwankende Performanz durch selbiges bedingt erwartet werden. Durch den Vorteil einiger Datenstrukturen in der Suchgeschwindigkeit gegenüber anderen sollte das gesamte Performanz-Bild jedoch in Diskrepanz zum Speicherverbrauch durch diesen Faktor dominiert werden.