

Datenbankanalyse

1. MongoDB

Für den kleinen Prototypen wurde sich für MongoDB entschieden, da diese eine der bekanntesten NoSQL Datenbanken ist. Zur Auswahl stand ebenfalls eine MySQL- Datenbank, da die Produkte ebenfalls tabellarisch gespeichert werden könnten. Jedoch ist einer der Vorteile von MongoDB, dass die Dokumente kein festes Schema haben und somit die Tabellen nicht umstrukturiert werden müssen, wenn neue Felder eingeführt werden. Ebenfalls können Verbindungen zwischen Datenbanken hergestellt werden, durch Referenzen. Falls dies Mal nötig sein sollte und im Gegensatz zu beispielsweise MySQL muss der Primärschlüssel dann nur im Dokument einzigartig sein. Des Weiteren ist MongoDB gut für Webprojekte geeignet (vgl. <https://www.ionos.de/digitalguide/websites/web-entwicklung/mongodb-vorstellung-und-vergleich-mit-mysql/> , letzter Aufruf 30.12.2020). Darüber hinaus ist aus der Vorlesung bekannt, dass die Verarbeitung und Speicherung schneller möglich ist und auch Kopien besser sind, an dieser Stelle sind die Stichworte Skalierbarkeit, Verfügbarkeit und Flexibilität zu nennen. Achtung: MongoDB hat all diese Vorteile, aber bei Recherchen wurde das Problem deutlich, dass durch die Verwendung von Spring, die Änderungen weniger leicht zu machen sind. Diese Problematik beruht jedoch allen Anschein nach, allein auf Spring. (vgl. Data migration with MongoDB n Spring Boot, Stephan Kuligin, 12.05.2019, <https://medium.com/@kuligin.ste/data-migration-with-mongodb-%D0%B8-spring-boot-773a79f38e4d>, letzter Aufruf 10.01.2021)

2. Cassandra

Der Grund für die Cassandra- Auswahl ist ein recht simpler: Die Installation von InfluxDB machte unerwartet viele Probleme, während Cassandra sich sehr schnell installieren ließ. Diese Datenbank ist selbstverständlich für die clickstreams.

3. InfluxDB

Im zweiten Versuch (die Gründe dafür stehen im Verlauf des Dokuments) versuchte man InfluxDB anzubinden.

Beide Datenbanken hatten anfängliche Probleme bei der Einbindung in das Javaprojekt, so konnte erst nicht auf die Database von MongoDB zugegriffen werden auf Grund von Authentifikationsproblemen, die mit Hilfe eines neuen Nutzers und Passworts gelöst werden konnten. Bei Cassandra gab es ebenfalls Probleme beim Einbinden, da nicht auf die Daten zugegriffen werden konnte.

Durch die Verwendung von Docker hatte man bei der Speicherung von Clickstreams und der IP zunächst eine falsche IP angezeigt bekommen, aber die IP wurde richtig angezeigt, wenn man von einem Mobilgerät auf die Website Zugriff (im lokalen Netz).

Die Auswahl der clickstream Datenbank wurde bei der Verbindung zu Grafana etwas bereut. Grafana kann direkt mit InfluxDB zusammenarbeiten, unterstützt jedoch nicht Cassandra. Um diese Schwierigkeit zu meistern, gibt es im Netz viele verschiedene Lösungsansätze. Alle laufen darauf

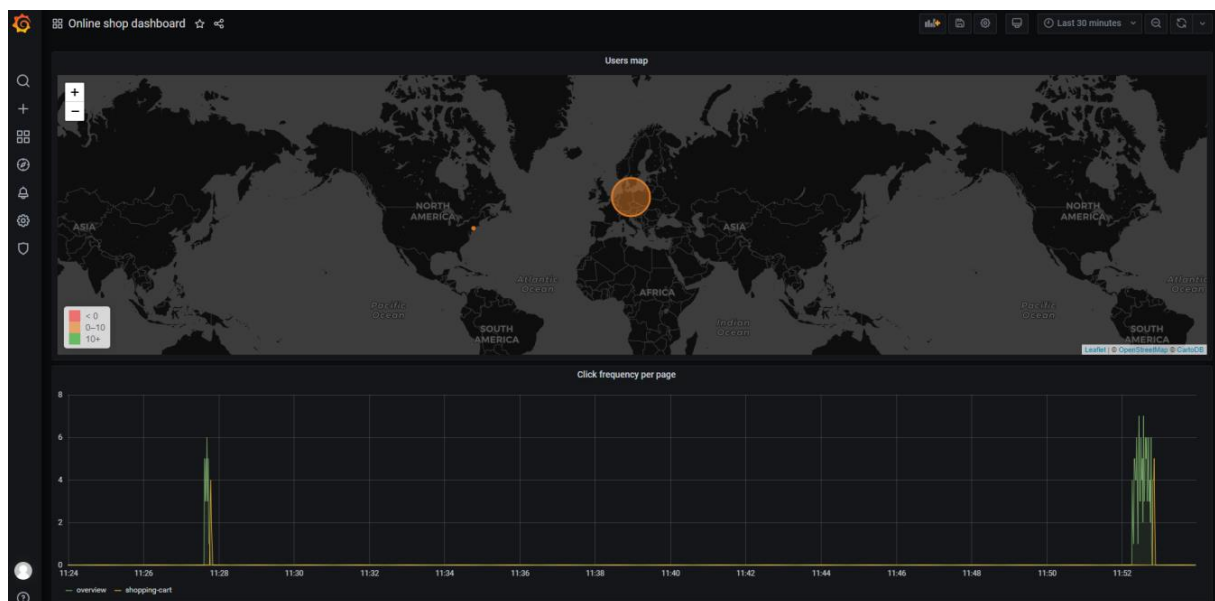
hinaus, dass man eine Datenbank zwischen Grafana und Cassandra hat und in dieser die Daten aus Grafana gespeichert werden.

Wir entschieden uns für Prometheus. Dies ist jedoch eine zweifelhafte Entscheidung gewesen aus dem Grund, dass Prometheus insbesondere Graphen zum Performance Monitoring lieferte und nichts Brauchbares für die Business Intelligence. Im späteren Verlauf, wenn man über diesen Prototypen hinausdenkt, wäre die Anwendung jedoch durchaus sehr sinnvoll, um genau zu beobachten, wann wo Probleme auftreten und so Rückschlüsse ziehen zu können.

Da dies nicht sonderlich hilfreich war für die Aufgabe, wurde anstelle von Cassandra InfluxDB angebunden als zweiter Versuch für Grafana. Dafür wurde in Java die Anmeldedaten für Cassandra zu jenen von InfluxDB geändert und letztendlich die Daten in Influx gespeichert. Grafana bietet eine direkte Schnittstelle für InfluxDB an, weshalb die Anbindung deutlich leichter war.

Weiterhin würde man noch einen Useragent implementieren, um beispielsweise den Webbrowser auswerten zu können.

Die Analysen:



Die obere Analyse „Users Map“ zeigt an aus welchen Teilen der Welt die meisten Zugriffe stammen. Für diese Abgabe sind es lediglich simulierte Daten.

„Click frequency per page“ zeigt als untere Analyse an, wie oft wann welche Seite aufgerufen wurde. Beispielsweise Shopping-cart.