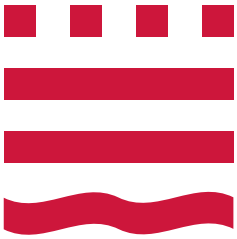

Technische Hochschule Brandenburg
University of Applied Sciences

BACHELORARBEIT

Serverless / Serverlose Architekturen für Konventionelle Webanwendungen

Vorgelegt von: Dragoljub Milasinovic
Matrikelnummer: 20140076

am: XX. Monat XXXX

zum
Erlangen des akademischen Grades

BACHELOR OF SCIENCE
(B.Sc.)

Erstbetreuer: Prof. Dr.-Ing. Schafföner
Zweitbetreuer: Jonas Brüstel, M.Sc.

Inhaltsverzeichnis

1 Quellen

OpsWorks AWS :: Deployment Strategy Cloud Design patterns :: Profi Patterns Man Trade Offs Arch :: Auswertung + Design guide <- self-adaptive arch?? AWS Sol. Arch :: Best Practices AWS vs Patterns general Amazon Web Services in Action :: Best Practices Arch Impl Cloud Design-Patterns for AWS :: Patterns list Serverless Arch AWS :: Main :: lambda: compute as a back end

2 Einleitung

Beispielhaftes Bild Abbildung 2.1

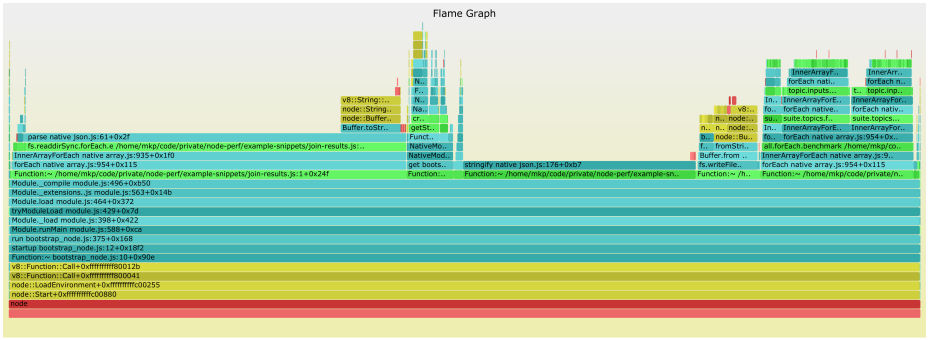

Abbildung 2.1: Beispiel Flame-Graph eines Node.js Skripts

Beispielhaftes Code-Snippet siehe Listing 2.1.

Listing 2.1: Aufnahme der „real“-Zeit

```
START=$(date +%s.%N)
node ${JS_FILE}
END=$(date +%s.%N)
DIFF=$(echo "$END - $START" | bc)
```

Hier kommt eine Bibliography-Referenz: [BME+07]

2.1 Motivation

Ausnutzung der Architektonischen -Arch- Gestalltungsmittel von AWS.

Start : MVP Minimal Viable Product Start::Chars: Arch + Domän Flexibilität - Schnelle Arch Änderungen

2.2 Ziel

Umsetzung der Kernfunktionalität einer Beispielanwendung mit ausschließlich „Serverlosen“Architekturen. wenn Zeit: Identifizieren von unverzichtbare Generische Funktionen für Serverless Anwendungen.

2.3 Aufbau der Arbeit

3 Grundlagen

Software Architektur: "What's important". Frühe, un-/schwer- veränderbare Entscheidungen. p.6 Studies in computational intlligen Ontologies: Level 4 SaaS : Scalable, Configurable, and Multitenant

Design: Lambda Orchestrator -> Pool of Lambdas to use

3.1 KOMA

Beispiel Anwendung „KOMA"ist ein Akronym für Kompetenz-Matrix. Die Umsetzung der Anwendung soll die von einem Individuum erworbene und zu erwerbenden Fertigkeiten, Kompetenzen und deren Niveau nachvollziehen. <– ?

Das Modell von KOMA basiert auf dem Grundmodell von „European Qualifications Framework Semantics"und dem deutschen Qualifikationsrahmen.

Ein Nutzungs-Fall aka. Use-Case: -Als Professor, will ich eine Auflistung der erworbene Fertigkeiten einer Klassenstufe abrufen können.

-Als Professor, will ich das Kompetenzniveau einer Kompetenz und derer Fertigkeiten abrufen können.

3.2 Datenhaltung Analyse und Auswahl

ontology: Formale Darstellung von Wissen durch eine Menge von Konzepten innerhalb eines Domänes und dessen Beziehungen -zwischen Konzepten-.

Semantics: relationships between signifiers De-notation: precise literal meaning of signifier Con-notation: associated meanings of signifier

Da die Konnotationen von beispielsweise Schlusselkompetenzen von Kontext zu Kontext unterschiedlich sind, bietet sich eine Ontologische Datenspeicherung an. Extensible, Migriation ok. Consistency ko.

Um die Konsistenz der Semantic des Grundmodells zu bewahren wird ein Relationale schema für die Ontologie benutzt.

3.3 Functionen

Einloggen: 0Auth Google gibt token, der wird in Lambda überprüft, Session in oauth.com verwaltet Query: SparQL ?x, ?y, ?z WHERE ... Datenspecherung Architektur: DynamoDB: speichert :individual als Schlussel und seine relative URL Ś3: speichert die .owl Dateien.

Lambda Funktion: Maps zwischen S3 und DynamoDB.

3.4 Patterns

Valet Key

Static Content Hosting ok

Sharding ok

Compute Resource Consolidation

Command and Query Responsability Segregation CQRS <- readS3UpdateDynamo.js

4 Ergebnis und Auswertung

cons: kurzlebige konfigurationen herausfinden ?? tracking? viel Konfiguration, kaum Konveniton -> .json 4 everything local testing braucht event-symulation.json

Entwurfsmuster von Frameworks wie JEE durch eine Annotation generisch eingesetzt sind neu zu implementieren in der Cloud

5 Zusammenfassung und Ausblick

Listings

Abbildungsverzeichnis

Abkürzungen

GC Garbage Collection

„Garbage Collection“ bezeichnet die automatische Speicherwaltung zur Minimierung des Speicherbedarfes eines Programmes. Garbage Collection (GC) wird zur Laufzeit durch Identifikation von nicht mehr benötigten Speicherbereichen ausgeführt. Im Vergleich zur manuellen Speicherverwaltung benötigt GC mehr Ressourcen.

Literaturverzeichnis

[BME+07] G. Booch, R. Maksimchuk, M. Engle, J. Conallen, K. Houston, and B.Y.P. D. *Object-Oriented Analysis and Design with Applications*. Pearson Education, 2007.

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die von mir eingereichte Masterarbeit selbstständig verfasst, ausschließlich die angegebenen Hilfsmittel benutzt und sowohl wörtliche, als auch sinngemäße entlehnte Stellen als solche kenntlich gemacht habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Brandenburg an der Havel, XX. Monat 2017

Vorname Nachname