



Cloud Computing

Klausur an der Hochschule Karlsruhe – University of Applied Sciences
Wintersemester 2023/24, Montag 29.01.2024, 14:00 Uhr

- Name: _____ Punkte: _____ / 100 (40 zum Bestehen) Note: _____
- **Disclaimer:**
- Der Lösungsweg muss bei allen Aufgaben ersichtlich sein
 - Keine Hilfsmittel

Aufgabe 1: Begriffswelt

___/10

___/10 Punkte

Es ist kalt in Deutschland, daher haben Sie entschieden, die Gunst der Stunde zu nutzen und bauen *Hola!* eine Plattform für Cloud Computing auf Gran Canaria auf, mit Meerwasserkühlung, Mitarbeitern, die auch im Süden arbeiten wollen und günstigem Strom.

Für die häufig anzutreffenden digitalen Nomaden, die Ihnen bei der Realisierung helfen ist es nötig, die Begriffe aus dem Themenbereich Cloud Computing kurz und prägnant zu erklären.

Zum Glück haben Sie die Vorlesung bei Fischi (und vielleicht bei Gio) besucht und können kurz und prägnant die folgenden Begriffe erklären/definieren:

Grid, Multiversion Concurrency Control, Oversubscription, N-Tier Architektur, Bulkhead, Scale Up, Race Condition, WSDL, Hypervisor, Docker

Aufgabe 2: Grundlagen

A) ___/4 B) ___/5 C) ___/5 D) ___/4 E) ___/5 F) ___/4. G) ___/5

___/32 Punkte

- a) Für die Kunden von *Hola!* erarbeiten Sie eine Entscheidungsvorlage: was sind Vorteile von Ihrer Public Cloud gegenüber eigenem Betrieb beim Kunden
- Alle Daten im eigenen Keller Zahlung nach Nutzung
- Einzige Lösung die auch auf Inseln funktioniert
- Energieeffizienz Automatisierbarkeit
- Erreichbarkeit per Auto Leichte Anbieterwechsel
- b) Erklären Sie Ihrem neuen digitalen Nomaden-Freelancer wie ein Loadbalancer prinzipiell funktioniert!
- c) Machen Sie aus dem Themenfeld von *Hola!* jeweils ein Beispiel für Use-Cases die im CAP Theorem C, A und P besonders deutlich erfordern würden.
- d) Vieles an *Hola!* wird ein interessantes verteiltes System. Nicht alles lässt sich verteilen, geben Sie ihren Kunden ein einfaches Beispiel für 7 Zeilen nicht parallelisierbarem Code, in dem eine Control Dependency vorkommt.
- e) Bei den Architekturdiskussionen zum *Hola!*-Setup kommt auch geografisch verteiltes Sharding der Speicherung von Kundendaten auf den Tisch – was könnte dafür dann eine sinnvolle Wahl der Keys sein? Wenn Sie Sharding nutzen, welche Probleme müssen Sie dann langfristig bei *Hola!* beachten?
- f) Welche der folgenden sind Stabilitäts-, also Resilienzpatterns? Bitte klären Sie Ihre neuen Kollegen bei *Hola!* auf, indem Sie diese Patterns ankreuzen.
- Proactor Circuit Breaker Bulkhead Factory
- Singleton Flow Control Caching Timeout
- Streams Doubleton Retry
- Flat JVM
- g) Welche Abstufungen an Konsistenz gibt es? Ordnen Sie die Konstistenzart aus der vorderen Spalte der jeweils passenden Beschreibung zu.

Konsistenzart	Beschreibung
1 Monotonic Read	A Leseoperation liefert immer den neusten Wert
2 Monotonic Write	B Garantiert gleiche Schreib-Reihenfolge für alle Knoten
	Schreiben auf Version X auf anderen Knoten nur erlaubt wenn dort auch schon Version X
3 Read Your Writes	C vorliegt
4 Write Follows Read	D Niemals ältere Version beim Lesen des gleichen Schlüssels liefern
5 Strict	E Client erhält nach dem Schreiben nie einen älteren Wert als zuletzt geschrieben

Aufgabe 3: Algorithmen

A) ___/9 B) ___/9 C) ___/8 D) ___/7

___/33 Punkte

- a) Ihre *Hola!*-User sollen auch per Mobilanwendung die Cloud-Services kontrollieren.
- Warum ist daher der Threadpool kein ideales Pattern für die Verarbeitung der User-Requests in Ihren Frontend-Webservern?
 - Was nehmen Sie stattdessen
 - Wie unterscheidet sich das grundlegende Vorgehen bei Ihrer Alternative
 - Warum ist das in Ihrem Fall besser? (Begründen Sie Ihre alternative Wahl)
- b) Sie überlegen sich, für Cloud-Service-Verträge bei *Hola!* Bitcoin als Zahlungsmittel zu verwenden. Schreiben Sie in Pseudocode einen Algorithmus, der zuverlässig prüft ob eine Zahlung bei **Ihnen** angekommen ist.
- c) Welches war der bahnbrechende algorithmische Fortschritt, der mit der Einführung von Bitcoin realisiert wurde - für welche Problemstellung fand der Erfinder von Bitcoin einen neuen Lösungsansatz, der die bisher bekannten Algorithmen übertraf? Skizzieren und erklären Sie diese Problemstellung.
- d) Auch bei *Hola!* kommen Bloomfilter (in der vereinfachten Form aus der Vorlesung.) zum Einsatz (Details seien der Fantasie der StudentInnen überlassen). Berechnen Sie den resultierenden Counting Bloom Filter in dem die Worte „error“, „warning“ und „info“ enthalten sind.

Wie sähe der Filter aus, wenn „info“ wieder gelöscht würde? (Hashfunktion $n \text{ MOD } 7$, „a“ = 1, Groß/Kleinschreibung ist zu ignorieren!)

Nach

Eintragen:

0	1	2	3	4	5	6

Nach

Löschen:

0	1	2	3	4	5	6

Aufgabe 4: Skalierung / Virtualisierung / Anbieter

A) ___/4 B) ___/4 C) ___/3 D) ___/5 E) ___/5 F) ___/4

___/25 Punkte

- a) Sie bieten auf *Hola!* den Kunden auch öffentliche Kubernetes Cluster.
Wer administriert/verwaltet in diesem Szenario:
- Die Kernelversion:
 - Die Pakete des Betriebssystems:
 - IP Adressen:
 - Die Applikationskonfiguration:
- b) Als skalierbare und auch sonst recht gut Cloud-geeignete Datenbank setzen Sie für die Speicherung der Kundendaten auf Apache Cassandra. Eine besondere Eigenschaft dieser Datenbank ist es, mit „Tuneable Consistency“ für jeden Zugriff die Konsistenzart wählen zu können. Streichen Sie aus folgender Liste die Konsistenzlevel, die es bei Cassandra NICHT gibt:
- | | | |
|------------------|---------|----------------|
| • INTENSE QUORUM | • ANY | • QUORUM |
| • PERFECT QUORUM | • MANY | • BASIC QUORUM |
| • LOCAL QUORUM | • MAYBE | • PARTLY |
| • ALL QUORUM | • ONE | • MONDAY |
| • ANY QUORUM | • NONE | • LOG |
- c) Was ist das Quorum bei Ihrer Cassandra Installation auf Gran Canaria mit 10 Knoten verteilt auf 2 RZs und dem Replikationsfaktor 5?
- d) Als erfahrener Entwickler von verteilten Systemen kennen Sie natürlich auch die besonderen Eigenschaften und Randbedingungen bei Cassandra gut. Worauf müssen Sie achten, wenn Sie die Datenstrukturen auf Cassandra entwerfen, die hoch performante Zugriffe erlauben sollen? Gibt es Use-Cases für die Cassandra nicht sehr gut geeignet ist?
- e) Sie versuchen den Kunden die Kubernetes Cluster aus Aufgabe a) nahezubringen. Vergleichen Sie dazu Vor- bzw. Nachteile der Container Virtualisierung gegenüber einer Lösung, die auf Vollvirtualisierung basiert.
- f) Ihr *Hola!* RZ-Netzwerk soll modern sein, Sie setzen auf SDN. Welche der folgenden Aussagen treffen auf SDN zu, kreuzen Sie an!
- die Netzwerkelemente konzentrieren sich auf Forwarding und Processing,
 - Zusatzfunktionalität kann in Netzwerkkapps ausgelagert werden,
 - Alle Anbieter von SDN Komponenten unterstützen OpenFlow,
 - die Technologien sind schon so alt wie das Internet,
 - Layer 3 wird nicht unterstützt,
 - dynamische Multimandanten-Setups werden einfacher,
 - man kann damit eine verteilte *Hola!* Infrastruktur aufbauen,
 - IDS wird zu einer Art Netzwerk-App