Cloud Computing

Klausur an der Hochschule Karlsruhe – University of Applied Sciences
Sommersemester 2023, Montag 10.07.2023, 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

Sie finden dass Twitter, Thread, Mastodon und wie die Nachrichtenplattformen auch alle heißen mögen nicht gut genug performen und möchten Ihr eigenes Ding machen!

Quitter soll es heißen! Lauter Quits und Quitten sollen die Nachrichtentypen der Zukunft sein!

Um das realisieren zu können stellen Sie ein Team zusammen und müssen zuallererst die Begrifflichkeiten für das gemeinsame Know-How klären, zum Glück haben Sie die Vorlesung bei Fischi und Gio besucht und können kurz und prägnant die folgenden Begriffe definieren:

Scale Out, Consistent Hashing, PaaS, LAMP, Amdahls Law, Eventual Consistency, BASE, Bulkhead, Row Key, SDN

Aufgabe 2: Grundlagen

A)__/5 B) __/5 C)__/5 D)__/6 E)__/6 F)__/5 __/32 Punkte

A) Beim Aufbau von Quitter stehen Sie vor der Frage ob sie Teile des Projektes in einer public Cloud realisieren sollten (statt alles in eigenen Rechenzentren…).
Welche Vorteile hätte das?
Welche Nachteile?

B) Quits und Quitten sind User Generated Content.
Sie überlegen das bei Amazon auf S3 abzulegen, welche Business-Überlegungen müssen Sie sich dabei machen?

C) Egal ob in der private oder public Cloud – Quitter wird ein interessantes verteiltes System.
Nicht alles lässt sich verteilen, geben Sie ihrem Team ein einfaches Beispiel für 7 Zeilen nicht parallelisierbaren Code in dem eine Flow Dependency vorkommt.

D) Quitter soll auch Quitten unterstützen die von anderen Usern als den ursprünglichen Autoren nachträglich angepasst werden können.
Das ist im verteilten System gar nicht so einfach – Sie überlegen sich, Multiversion Concurrency Control einzusetzen.
Welches Problem löst das?
Wie funktioniert es prinzipiell?
Welche Voraussetzungen müssen für die Funktion erfüllt sein?
Nennen Sie noch andere Beispiele für mögliche Funktionen bei Quitter bei denen die Methode von von Vorteil wäre.

E) Im verteilten Softwaresystem von Quitter spielt sich natürlich viel in den Ecken und an den Kanten vom CAP Theorem ab.
Was besagt es?
Denken Sie sich für jeden Extremfall passend zur Story ein Beispiel aus.

F) Bei den Architekturdiskussionen zum Quitter-Setup kommt auch geografisch verteiltes Sharding der Speicherung der Quitten auf den Tisch – was könnte dafür dann eine sinnvolle Wahl der Keys sein?
Wenn Sie Sharding nutzen, welche Probleme müssen Sie dann langfristig bei Quitter beachten?

Aufgabe 3: Algorithmen

A)__/7 B) __/7 C)__/7 D)__/7 E)__/6 __/34 Punkte

A) Quitten und Quits werden von Ihren Usern wahrscheinlich hauptsächlich per Mobilanwendung bearbeitet und requitted.
- 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 bezahlte Transaktionen auf Quitter Bitcoin als Zahlungsmittel zu verwenden.
Leider ist Bitcoin wegen seines Stromverbrauchs in Verruf geraden – warum braucht das eigentlich so viel Strom?
Erklären Sie Ihrem Team das Grundprinzip des Proof-of-Work Algorithmus bei Bitcoin.

C) Welches war der bahnbrechende algorithmische Fortschritt, der mit der Einführung von Bitcoin realisiert wurde - für welches Problemstellung fand der Erfinder von Bitcoin einen neuen Lösungsansatz der die bisher bekannten Algorithmen übertraf? Skizzieren und erklären Sie diese Problemstellung.

D) Die Quitten von Ihrem Quitter werden häufig requitted, dazu ist es nötig, festzustellen ob ein Quit überhaupt noch existiert.
Damit dafür nicht so viele Service-Calls nötig sind, verwalten Sie Quits in einem Bloom Filter, so kann sehr schnell herausgefunden werden ob ein Quit schon existiert.
Schreiben Sie in Pseudo Code eine Funktion, die die Quit-ID in Ihrem Bloom Filter einträgt (unter Verwendung der gegebenen Hash-Funktionen `HashQ1(…)`, `HashQ2(…)`, `HashQ3(…)`).
Schreiben Sie in Pseudo Code eine weitere Funktion, welche im Bloom Filter prüft, ob die Quit-ID schon vorhanden ist.
Wenn nun häufig Quits gelöscht werden, gibt es ein Problem mit der Anwendung von Bloom-Filtern.
Haben Sie das Problem in Ihrer Implementierung berücksichtigt?
Wenn ja: wie?
Wenn nein: was müssten Sie anpassen?

E) Der Bloom-Filter hilft schon bei der Verarbeitung von Quits, aber nun soll noch eine Suche über die Worte die in Quits enthalten sind über alle existierenden gespeicherten Quits und Quitten realisiert werden – dafür wäre doch Map Reduce eine gute Lösung!
Was könnte ein Mapper in diesem Vorgang machen und was ein Reducer?

Aufgabe 4: Skalierung / Virtualisierung / Anbieter

A)__/4 B)__/6 C)__/4 D)__/5 E)__/5 __/24 Punkte

A) Sie skalieren Ihr Quitter auf einem öffentlichen Kubernetes Cluster.
Wer administriert/verwaltet in diesem Szenario:
- Die Kernelversion:
- Die Pakete des Betriebssystems:
- IP Adressen:
- Die Applikationskonfiguration:

B) Weshalb könnte sich Ihr Architekt für die Lösung aus Aufgabe A) entschieden haben?
Vergleichen Sie dazu Vor- bzw. Nachteile der Container Virtualisierung gegenüber einer Lösung die auf Vollvirtualisierung basiert.

C) Welche Probleme löst Ihnen dabei Kubernetes?

D) Natürlich ist Cassandra die Datenbank der Wahl für das Speichern der Quitten und Quits mit bei Quitter.
Was ist das Quorum bei einer Cassandra Installation mit 10 Knoten verteilt auf 2 RZs und dem Replikationsfaktor 5, wie Sie sie verwenden?

E) Welche der folgenden Aussagen treffen auf SDN zu, kreuzen Sie an!
[ ] die Netzwerkelemente konzentrieren sich auf Forwarding und Processing,
[ ] Zusatzfunktionalität kann in Netzwerkapps 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 Quitter Infrastruktur aufbauen,
[ ] IDS wird zu einer Art Netzwerk-App