

Willkommen bei Verteilte Systeme!

Von Datenbanken über Webdienste bis zu p2p und Sensornetzen.



Heute: **Replikation, CALM und CRDTs.**
Versprech nur, was du halten kannst.

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Replikation

Replikation

Speichern von Kopien auf mehreren Maschinen, die über Netzwerk verbunden sind.

Gründe für Replikation:

- Geographische Skalierung: Daten eines Nutzers näher am Nutzer -> Verringerung der Latenz
- Anwendung funktioniert trotz ausgefallenen Knoten.
- Größenmäßige Skalierung: Mehr Nutzer können die Anwendung gleichzeitig verwenden.¹

Annahme: Gesamter Datensatz passt auf eine Maschine -> Keine Partitionierung (Sharding)

¹Das machen wir bei Disney: Synchronisierte Caches

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Multi Leader

Multi Leader Replication

Nachteile Single Leader

- Leader nicht erreichbar => keine Änderungen
- Einzelner Leader -> Flaschenhals

Anwendungen

- Progressive Apps: Offline arbeiten
- Kollaborative Apps: Etherpad, Cryptpad, Google Docs etc.

Nachteil Multi Leader

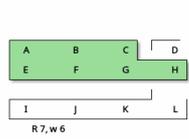
- Lösung von Schreibkonflikten nötig

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Multi Leader

Quorum: Write-Write-Konflikte vermeiden



- Wenn $w \leq \frac{n}{2}$ können 2 Nutzer widersprüchliche Daten schreiben.
- Beim Lesen erkennbar, da $r > n - w$
- write-write Konflikt oder stale data

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Availability

Total Available / High Available

- Antwort erhält, wer **einen** korrekten (nicht versagenden) Server kontaktieren kann
- Auch bei Netzwerkpartitionen zwischen Servern

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Availability

Unavailable

System ist nicht verfügbar bei Netzwerkpartitionen.

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Availability

Ziele

- Ihr kennt verschiedene Arten der Replikation.
- Ihr versteht, dass Replikation zu Inkonsistenzen führen kann.
- Ihr kennt das CALM Theorem.
- Ihr versteht, dass Koordination vermieden werden kann und dies zu einfacheren Systemen führt.

Übersicht Replikation

3 Arten von Replikation werden unterschieden:

- Single Leader
- Multi Leader
- Leaderless

Leaderless Replication

- Verbreitet durch Amazons Dynamo DB
- Auch Riak, Cassandra, Voldemort
- Writes auf jedem Knoten
- Meist „Quorum“ Reads und Writes.

Zusammenfassung Replikation

- Single, Multi, Leaderless
- (a)synchrone Replikation
- Inkonsistenzen möglich
- Quorum Bedingung: $r + w > n$

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Availability

Sticky Available

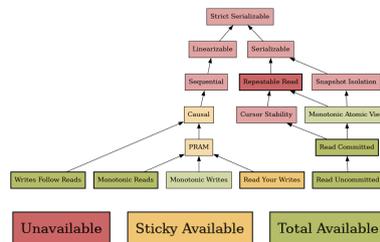
- Antwort erhält, wer einen Server kontaktieren kann, der den **gesamten, dem Nutzer bekannten Zustand** beinhaltet
- Auch bei Netzwerkpartitionen zwischen Servern

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Consistency

Consistency



Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Consistency

Ablauf heute

- Replikation
- Was ist Availability?
- Welche Konsistenzmodelle gibt es?
- Lässt sich Koordination vermeiden?

Single Leader



- Replika: Knoten, der eine Kopie speichert
 - Leader: Eine Replika mit Schreibrecht
 - Schreiben: Anfrage an Leader
 - Leader schreibt lokal
 - Sendet geänderte Daten an alle anderen Replikas (Follower)
 - Follower speichern die Änderungen lokal
 - Lesen auch von Followern
- Hierarchie ähnlich zu NTP.

Quorum

- Sende jeden write und read an n Knoten
- write ist erfolgreich wenn w Knoten ihn bestätigen
- read ist erfolgreich wenn r Knoten ihn bestätigen

Quorum Bedingung: $w + r > n$:

- garantiert Überlapp zwischen w-Knoten und r-Knoten
- $w < n$ kann bei ausgefallenen Knoten schreiben
- $r < n$ kann bei ausgefallenen Knoten lesen
- $w > \frac{n}{2}$ kann write-write Konflikte vermeiden

Availability

- Total Available / High Available
- Sticky Available
- Unavailable

Literatur: Highly Available Transactions: Virtues and Limitations Bailis et al. (2013).

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Availability

Sticky Available - Beispiel

- Daten auf mehrere Server repliziert
- Jede Replika enthält alle Daten
- Nutzer kontaktiert immer denselben Server
- => Sticky Available

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Consistency

Consistency und Availability - Bewertung

- Spektrum möglicher Consistency und Availability.
- Verschiedene Teile eines Systems können verschiedene Anforderungen haben.
- Informierte Entscheidungen treffen!
- Unsere Anwendung muss nicht in jedem Fall 100% konsistent sein.
 - Manchmal reicht eine Entschuldigung.
 - Aber angreifbar! (s. ACIDRain Paper (Warszawski and Bailis, 2017))
- Können wir uns auf Angaben von Herstellern verlassen?

Arne Babenhausen und Carlo Götz
Replikation, CALM und CRDTs

Einstieg Replication Availability Consistency CALM Theorem CRDTs Quellen Abschluss

Consistency

