

Big Data Management

Thema 14: Cassandra

Jan Kristof Nidzwetzki



Übersicht

1 Grundlagen

- Überblick
- Geschichte
- Datenmodell

2 Architektur

- Der logische Ring
- Persistenz der Daten
- Tunable Consistency
- Read Repair

3 Cassandra im Einsatz

- CQL – Cassandra Query Language
- Performancevergleich

4 Zusammenfassung

Was ist Cassandra?

Cassandra in 50 Worten

„Apache Cassandra is an open source, distributed, decentralized, elastically scalable, highly available, fault tolerant, tuneably consistent, column-oriented database that bases its distribution design on Amazon’s Dynamo and its data model on Google’s Bigtable. Created at Facebook, it is now used at some of the most popular sites on the Web.“¹

¹Quelle: Eben Hewitt, Cassandra: The Definitive Guide, O’Reilly Media, 2010, S. 14

Geschichte von Cassandra

Cassandra ...

- ... wurde 2008 von  entwickelt
- ... ist in Java geschrieben
- ... wird von der *Apache Software Foundation* betreut



- ... steht unter der *Apache License 2.0*

Wer setzt Cassandra ein?

Wo kommt Cassandra zum Einsatz?²

CERN: Legt Messdaten in Cassandra ab

ebay: Betreibt einen hochverfügbaren Cassandra Cluster verteilt über mehrere Rechenzentren

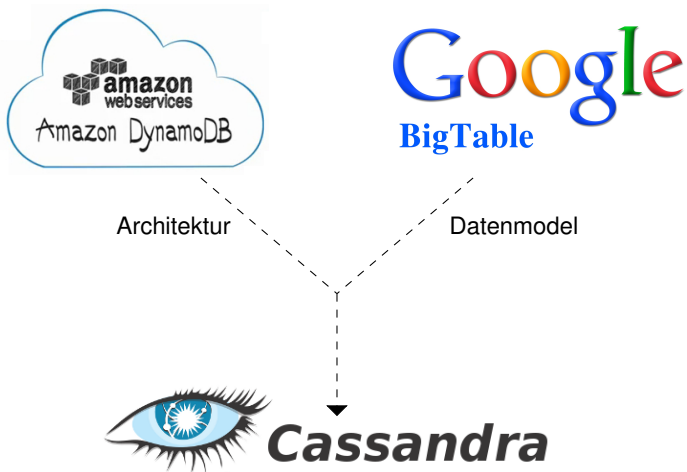
HP: Speichert Metadaten für einen Musik Streaming Dienst in Cassandra

IBM: Nutzt Cassandra als Backend in einem Webmail-Client

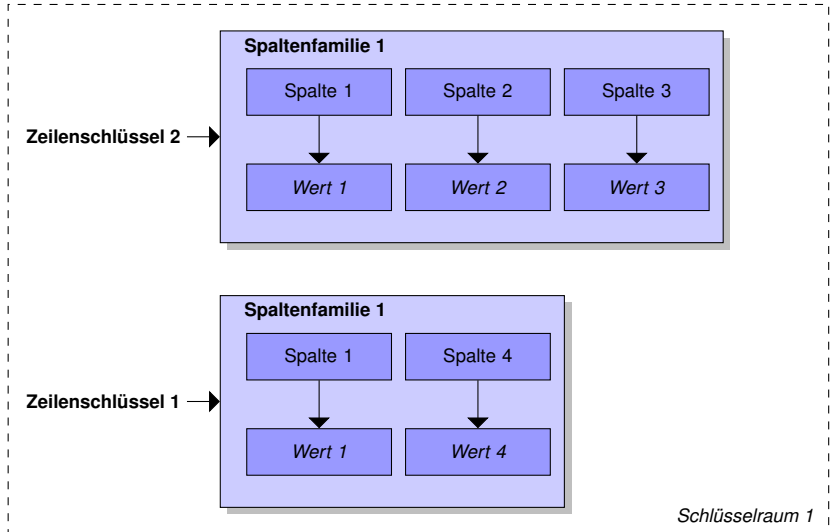
Twitter: Verwendet Cassandra zum speichern von Analysen

²Quelle: <http://planetcassandra.org/Company/ViewCompany>

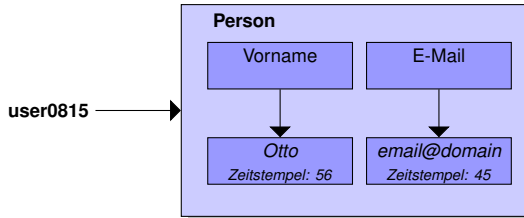
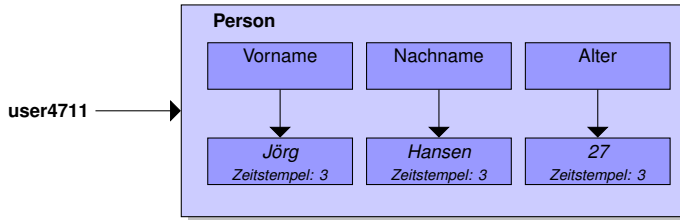
Familie



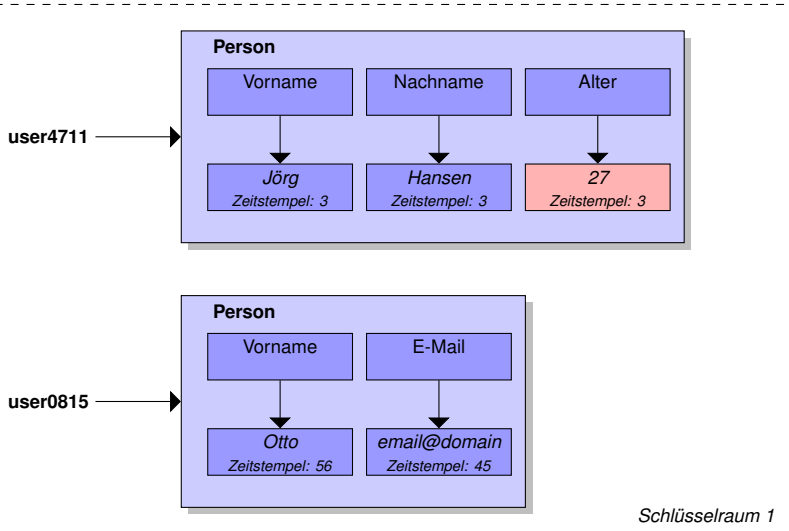
Datenmodell



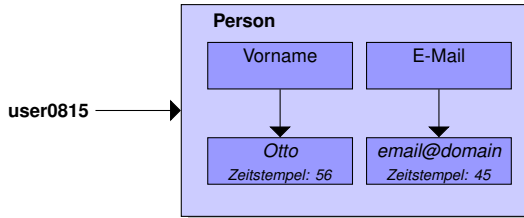
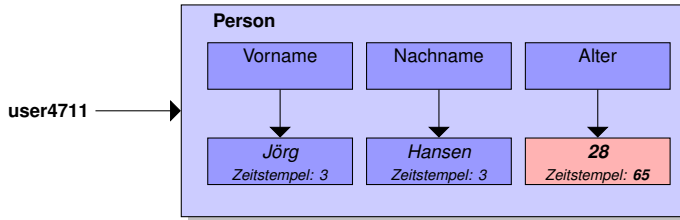
Datenmodell - Beispiel



Datenmodell - Beispiel



Datenmodell - Beispiel

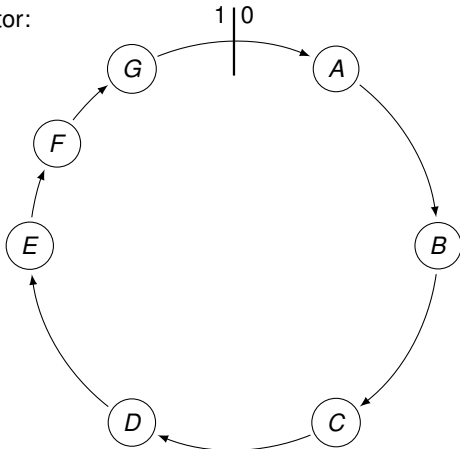


Begriffe

- **Knoten:** Ein Server auf dem Cassandra installiert ist
- **Replikationsfaktor:** Anzahl der Replikate einer Zeile
- **Partitionierer:** Berechnet den Ablageort einer Zeile
- **Platzierungsstrategie für Replikate:** Bestimmt auf welchen Knoten Replikate abgelegt werden

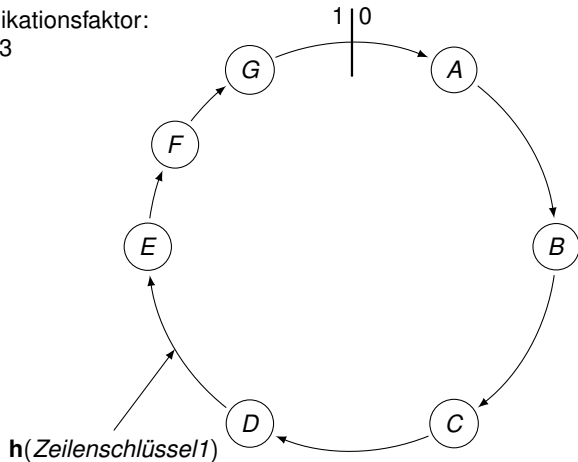
Der logische Ring

Replikationsfaktor:
 $N = 3$



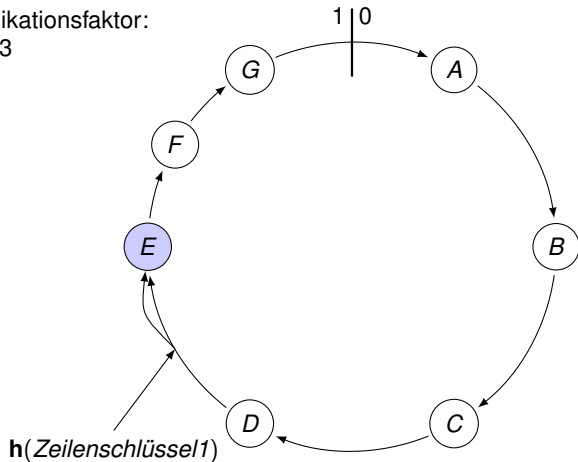
Der logische Ring

Replikationsfaktor:
 $N = 3$



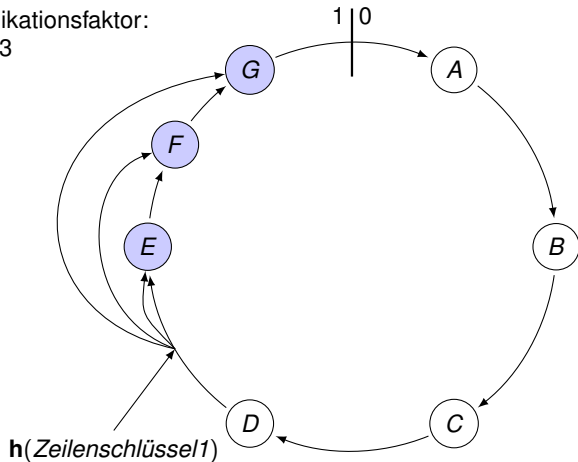
Der logische Ring

Replikationsfaktor:
 $N = 3$



Der logische Ring

Replikationsfaktor:
 $N = 3$



Der logische Ring

Logischer Ring mit einer Node

```
root@node1:~# /root/cassandra/bin/nodetool ring
```

```
Datacenter: datacenter1
```

```
=====
```

```
Address Rack Status State Load Owns Token
```

```
node1 rack1 Up Normal 78.02 KB 100.00% -894499901412982244
```


Der logische Ring

Logischer Ring mit zwei Nodes

```
root@node1:~# /root/cassandra/bin/nodetool ring
```

```
Datacenter: datacenter1
```

```
=====
```

Address	Rack	Status	State	Load	Owns	Token
						395519162814346212
node1	rack1	Up	Normal	82.69 KB	30.07%	-894499901412982244
node2	rack1	Up	Normal	1.55 MB	69.93%	395519162814346212

Der logische Ring

Logischer Ring mit drei Nodes

```
root@node1:~# /root/cassandra/bin/nodetool ring
```

```
Datacenter: datacenter1
```

```
=====
```

Address	Rack	Status	State	Load	Owns	Token
						395519162814346212
node1	rack1	Up	Normal	93.96 KB	30.07%	-894499901412982244
node3	rack1	Up	Normal	42.59 KB	44.34%	-76620706107975918
node2	rack1	Up	Normal	1.51 MB	25.59%	395519162814346212

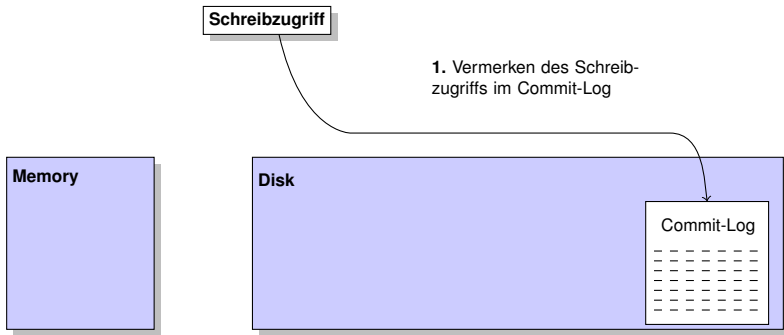
Persistenz der Daten

Schreibzugriff

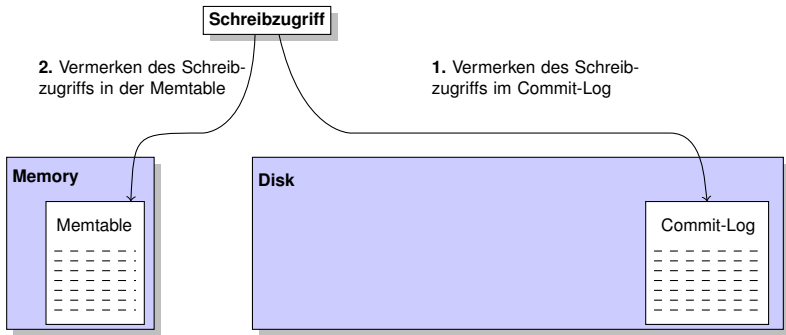
Memory

Disk

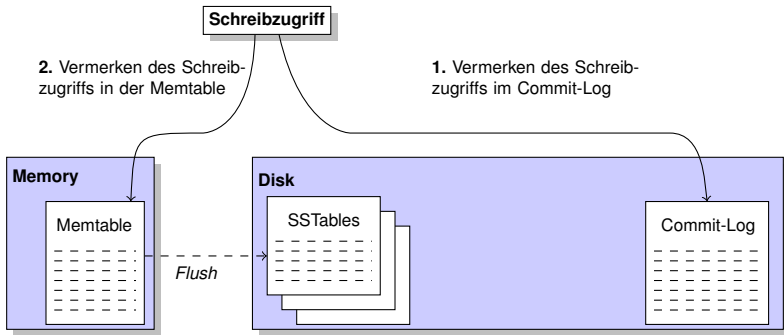
Persistenz der Daten



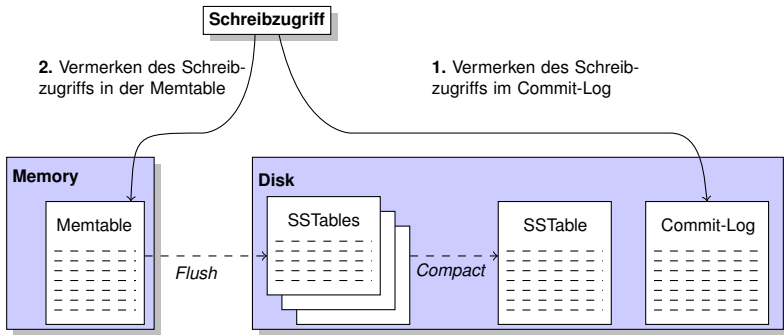
Persistenz der Daten



Persistenz der Daten



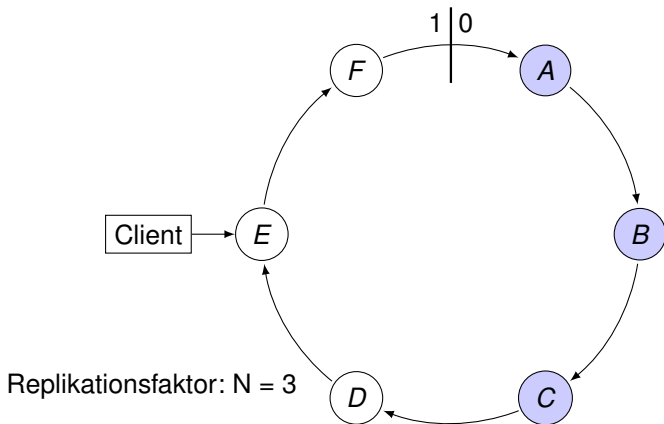
Persistenz der Daten



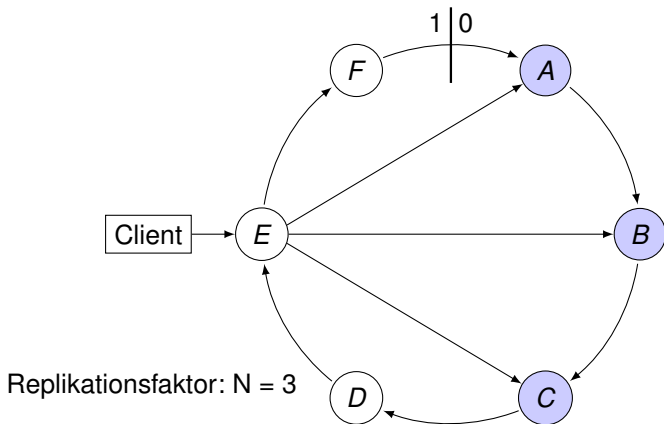
Tunable Consistency – Lesen

Konsistenz-Level	Bedeutung
ONE	Es werden die Zeilen von dem Knoten zurückgeliefert, welcher als erstes antwortet.
QUORUM	Haben $(\frac{\text{Replikationsfaktor}}{2} + 1)$ Knoten geantwortet, werden die Zeilen mit dem neuesten Zeitstempel an den Client ausgeliefert.
ALL	Verhält sich wie QUORUM, jedoch wird mit dem Ausliefern der Zeilen gewartet, bis die Zeilen von allen Knoten vorliegen.

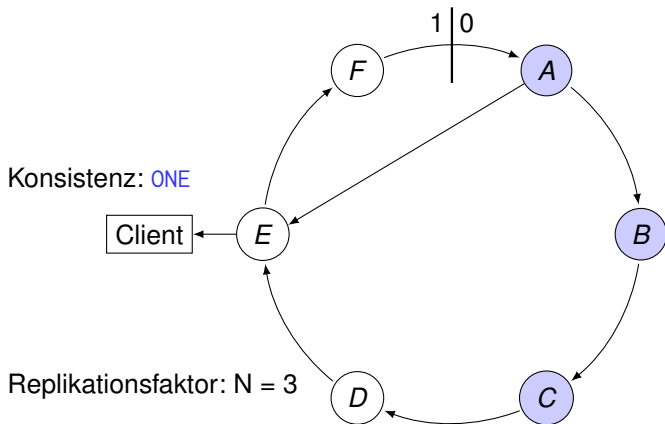
Tunable Consistency – Lesen



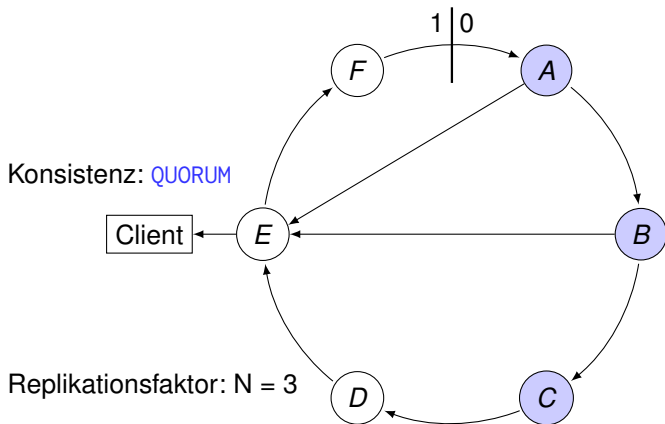
Tunable Consistency – Lesen



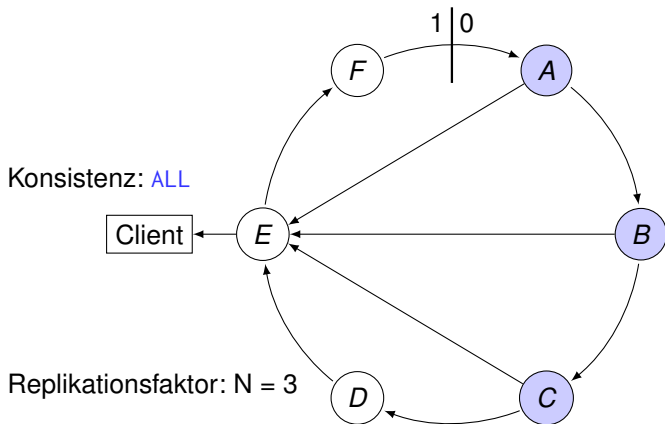
Tunable Consistency – Lesen



Tunable Consistency – Lesen



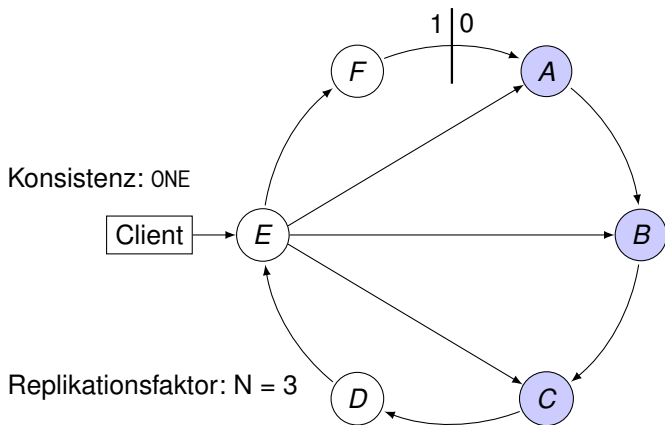
Tunable Consistency – Lesen



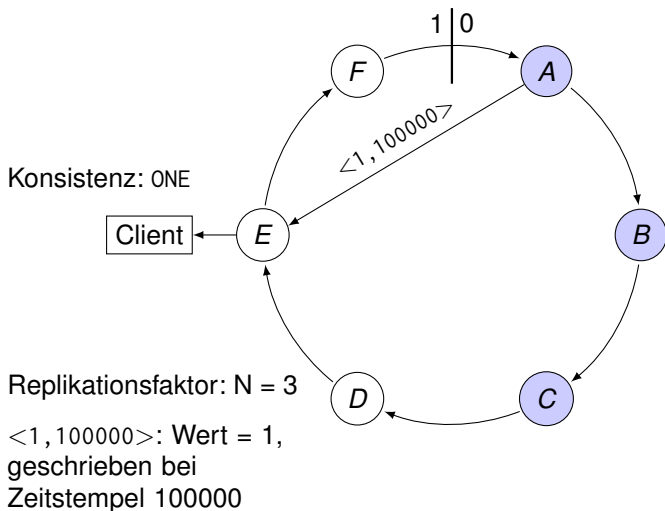
Tunable Consistency – Schreiben

Konsistenz-Level	Bedeutung
ZERO	Die Schreiboperation wird asynchron bearbeitet. Auftretende Fehler werden ignoriert.
ANY	Die Schreiboperation muss auf mindestens einem Knoten durchgeführt worden sein. <i>Hinted Handoffs</i> sind erlaubt.
ONE	Die Schreiboperation muss auf mindestens einem Knoten durchgeführt worden sein.
QUORUM	Es müssen mindestens $(\frac{\text{Replikationsfaktor}}{2} + 1)$ Knoten die Schreiboperation bestätigen.
ALL	Die Schreiboperation muss von allen Knoten bestätigt worden sein, welche für die Daten zuständig sind.

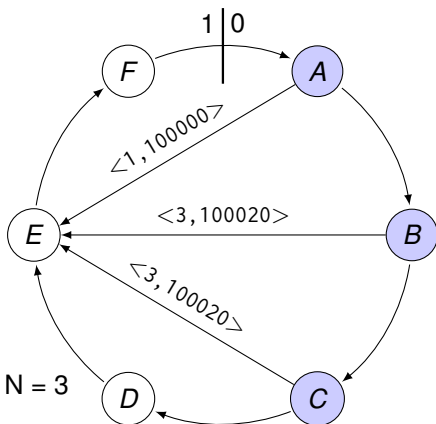
Read Repair



Read Repair



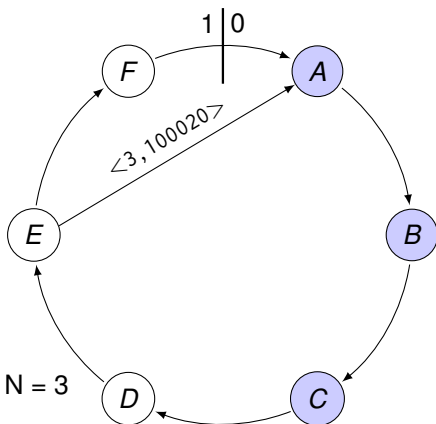
Read Repair



Replikationsfaktor: $N = 3$

$\langle 1, 100000 \rangle$: Wert = 1,
geschrieben bei
Zeitstempel 100000

Read Repair



Replikationsfaktor: $N = 3$

$\langle 1, 100000 \rangle$: Wert = 1,
geschrieben bei
Zeitstempel 100000

CQL – Cassandra Query Language

CQL ...

- ... ist eine Abfragesprache ähnlich *SQL*

CQL – Cassandra Query Language

CQL ...

- ... ist eine Abfragesprache ähnlich *SQL*
- ... ist an die Syntax von *SQL* angelehnt

CQL – Cassandra Query Language

CQL ...

- ... ist eine Abfragesprache ähnlich *SQL*
- ... ist an die Syntax von *SQL* angelehnt
- ... bietet eine stabile Schnittstelle zu Cassandra (JDBC-Treiber verfügbar)

CQL – Cassandra Query Language

CQL ...

- ... ist eine Abfragesprache ähnlich *SQL*
- ... ist an die Syntax von *SQL* angelehnt
- ... bietet eine stabile Schnittstelle zu Cassandra (JDBC-Treiber verfügbar)
- ... enthält **keine** Anweisungen für Joins, etc.

CQL am Beispiel

Abfrage einer Zeile – Casandra CLI und CQL

```
# CLI
```

```
get People['21'];
```

```
# CQL
```

```
SELECT * from People WHERE key = 21;
```

CQL am Beispiel

Anlegen einer Zeile – Casandra CLI und CQL

CLI

```
set users['jsmith'][firstname] = 'John';  
set users['jsmith'][lastname] = 'Smith';  
set users['jsmith'][age] = '22';
```

CQL

```
INSERT INTO users (KEY, firstname, lastname, age) VALUES  
('jsmith', 'John', 'Smith', '22');
```


CQL am Beispiel

Anlegen einer Zeile unter Angabe eines Konsistenz-Niveaus

```
# CLI
```

```
consistencylevel as QUORUM;  
set users['jsmith'][firstname] = 'John';  
set users['jsmith'][lastname] = 'Smith';  
set users['jsmith'][age] = '22';
```

```
# CQL
```

```
INSERT INTO users (KEY, firstname, lastname, age)  
VALUES ('jsmith', 'John', 'Smith', '22')  
USING CONSISTENCY QUORUM;
```

Performance

Paper: *Solving big data challenges for enterprise application performance management, 2012*

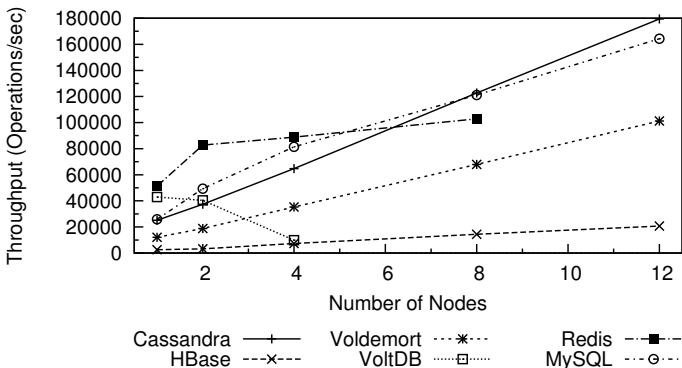


Abbildung: 5% schreibende Zugriffe, 95% lesende Zugriffe.

Performance

Paper: *Solving big data challenges for enterprise application performance management, 2012*

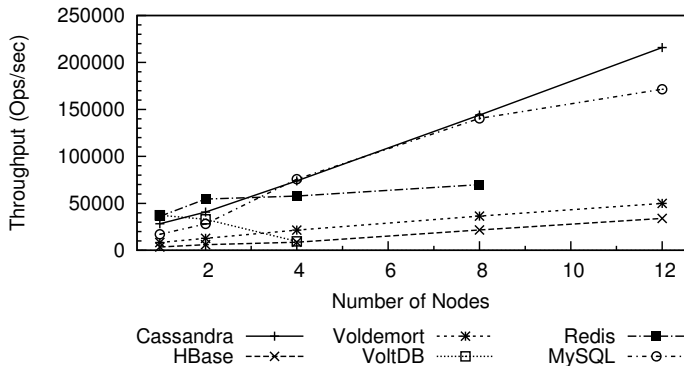


Abbildung: 50% schreibende Zugriffe, 50% lesende Zugriffe.

Zusammenfassung



- Ist ein spaltenorientiertes *Datenbankmanagementsystem*

Zusammenfassung



- Ist ein spaltenorientiertes *Datenbankmanagementsystem*
- Datenmodell von *Google Bigtable*

Zusammenfassung



- Ist ein spaltenorientiertes *Datenbankmanagementsystem*
- Datenmodell von *Google Bigtable*
- Architektur von *Amazon Dynamo*

Zusammenfassung



- Ist ein spaltenorientiertes *Datenbankmanagementsystem*
- Datenmodell von *Google Bigtable*
- Architektur von *Amazon Dynamo*
- Setzt auf *Tunable Consistency*

Zusammenfassung



- Ist ein spaltenorientiertes *Datenbankmanagementsystem*
- Datenmodell von *Google Bigtable*
- Architektur von *Amazon Dynamo*
- Setzt auf *Tunable Consistency*
- Inkonsistenzen werden u. A. mittels *Read Repair* behoben

Zusammenfassung



- Ist ein spaltenorientiertes *Datenbankmanagementsystem*
- Datenmodell von *Google Bigtable*
- Architektur von *Amazon Dynamo*
- Setzt auf *Tunable Consistency*
- Inkonsistenzen werden u. A. mittels *Read Repair* behoben
- Skaliert gut über mehrere *Knoten*

Vielen Dank für Ihre
Aufmerksamkeit