SCOOP SOFTWARE

Hochverfügbare Risiko Management Systeme auf Basis von Cassandra, Spark und Solr

Stefan Solich

SCOOP SOFTWARE

Agenda

01 Motivation

02 Cassandra / DataStax Enterprise (DSE)

03 Solr / DSE Search

04 Der Schlussteil

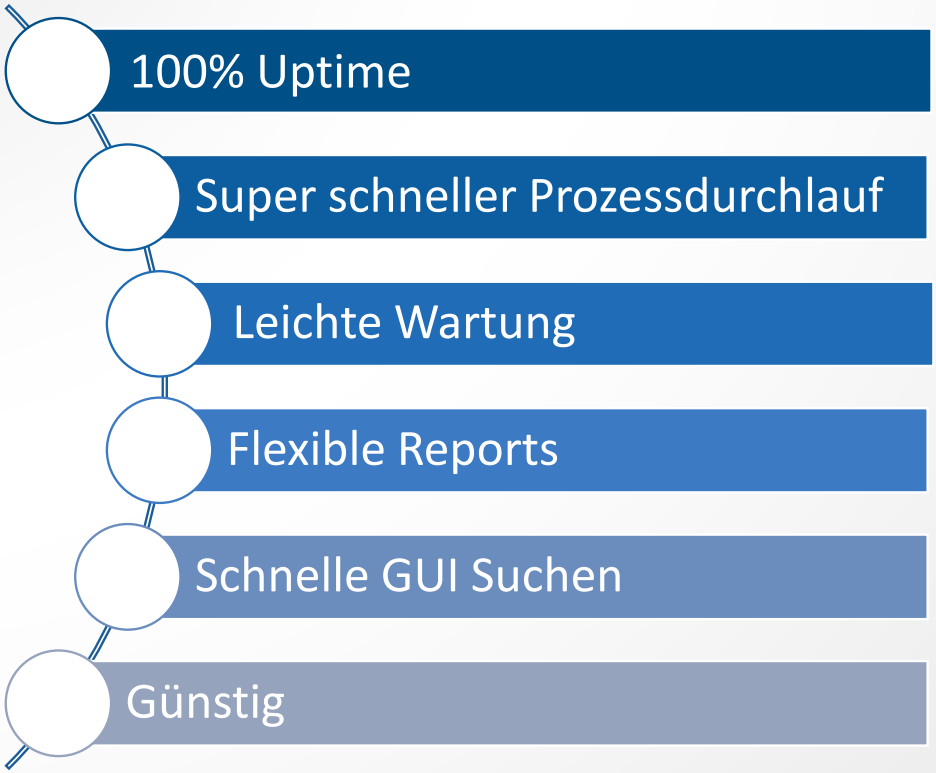
SCOOP SOFTWARE

01 Motivation

Fachliche Anforderungen

- Betrugsprävention
- Prüfung von Bonität
- Verwalten von Limits
- Freigabe von Kauftransaktionen

Anforderung (Simplified)



SCOOP SOFTWARE

Anforderung

- Nichts davon ist Anforderung
- Das ist der Wunsch an **jede** Software

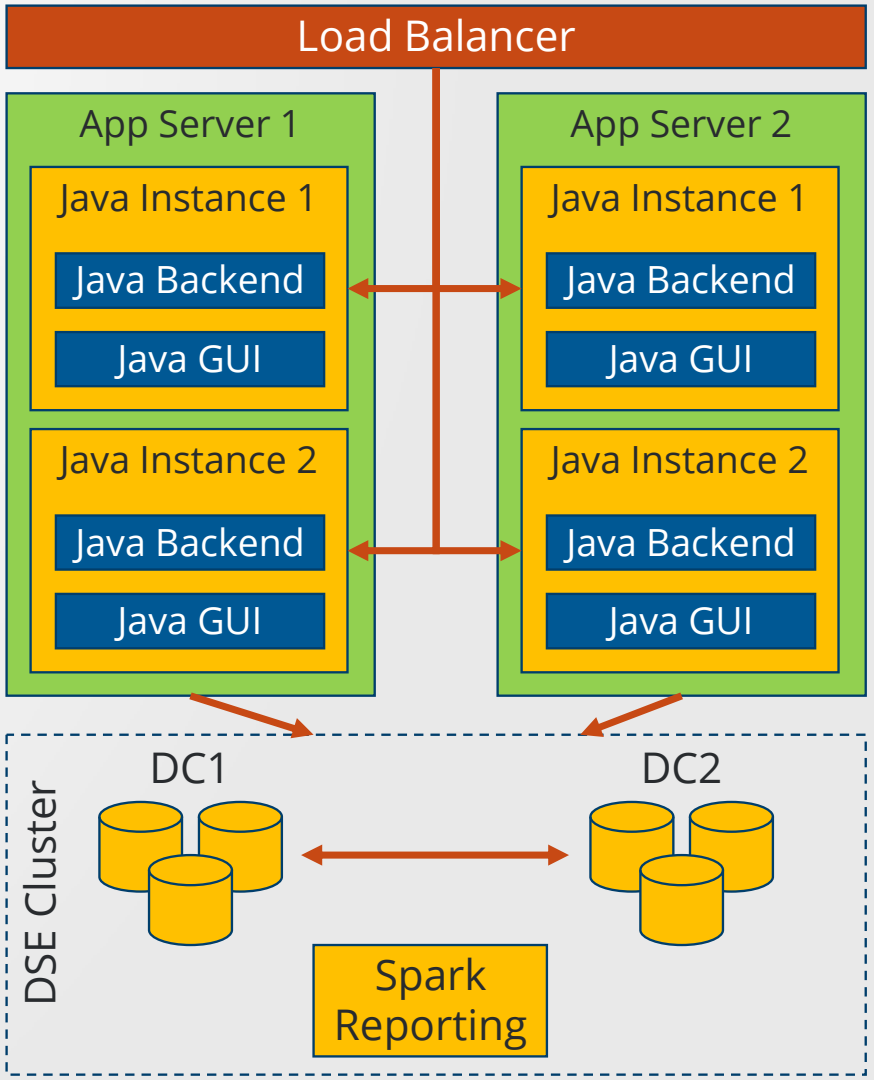
Cassandra hilft dabei

SCOOP SOFTWARE

Architektur

Heterogene Systemlandschaft:

- Cassandra Nodes Bare Metal
- Java Prozesse (je Fachlichkeit)
  - Auf Bare Metal
  - Docker
  - VM
  - OpenShift
  - Alle Multiinstanzfähig / Stateless



SCOOP SOFTWARE

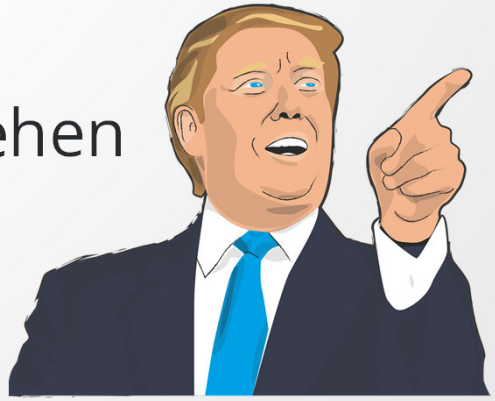
Geschwindigkeit

- Jede Transaktion
  - 6-8 Cassandra Read Requests
  - **20 ms Roundtrip (inklusive Netzwerklatenz)**
  - Write Requests erst nach der eigentlichen Entscheidung
  - Parallel/Asynchron
- 26 Mio Requests
- Peak zum Monatswechsel

SCOOP SOFTWARE

Ausfallsicherheit

- 2 Datacenter (1 darf komplett ausfallen)
- Applikationsknoten sind stateless
  - Jeder Knoten kann die Aufgaben eines anderen übernehmen
- Cassandra verteilt Daten über alle Datacenter
- Katastrophen überstehen



SCOOP SOFTWARE

Leichte Wartung

- OPS Center
- Lifecycle Manager
- Nodetool (repair)
- Keine Wartungsfenster nötig

SCOOP SOFTWARE

Reports und Suche

- Mit Cassandra selbst sehr unbequem
- Reports mit Apache Spark / DSE Analytics
- Suche mit Apache Solr / DSE Search

Günstig

SCOOP SOFTWARE

02
Cassandra / DSE

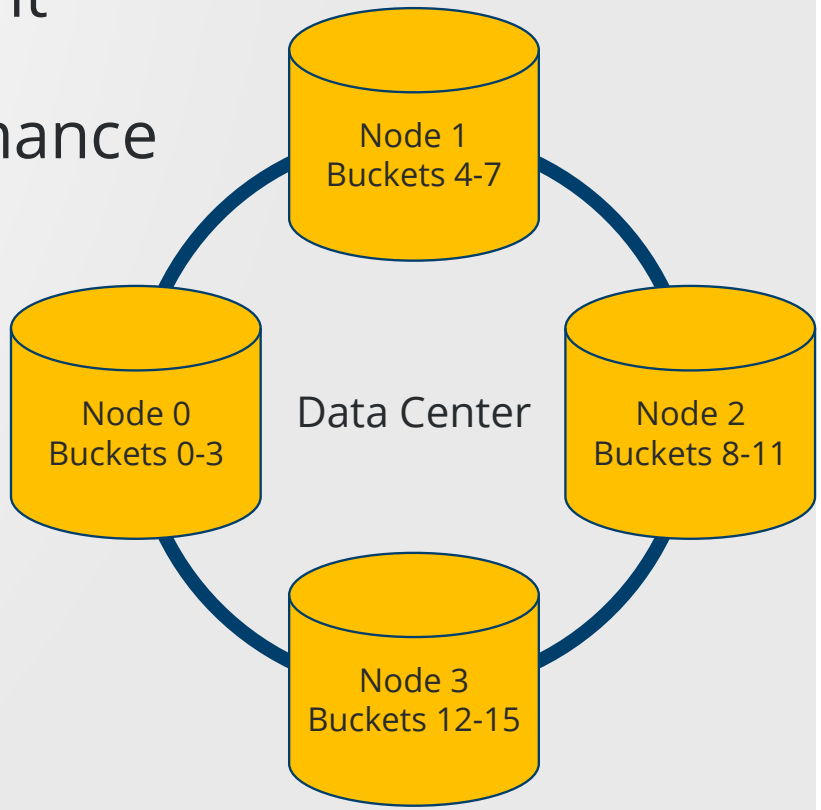
SCOOP SOFTWARE

Cassandra / DSE

cassandra

The Apache Cassandra database is the right choice when you need scalability and high availability without compromising performance

- Lineare Skalierung
- Commodity Hardware
- Datenreplikation
- kein Single Point of Failure
- Open Source



SCOOP SOFTWARE

CQL

- SQL like query language
- Indexes
- **Keine Joins**
- (Java) Driver mit großem Featureset

```
CREATE TABLE cycling.cyclist_category (
    category text,
    points int,
    id UUID,
    lastname text,
    PRIMARY KEY (category, points)
    )
    WITH CLUSTERING ORDER BY (points DESC);

INSERT INTO cycling.cyclist_category (category, points, id, lastname) VALUES ('test1', 25, uuid(), 'Duck');

SELECT * FROM cyclist_category WHERE category='test1';
```

SCOOP SOFTWARE

(Java) Driver: Key Features

- Statements / Prepared Statements
- Query Builder
- Object Mapper
- Synchrone / Asynchrone / Reactive API

- Load Balancing
- Retries
- Konfiguration

SCOOP SOFTWARE

Statements


```
import com.datastax.dse.driver.api.core.DseSession;
import com.datastax.oss.driver.api.core.cql.*;

try (DseSession session = DseSession.builder().build()) {
    ResultSet rs = session.execute("select release_version from system.local");
    Row row = rs.one();
    System.out.println(row.getString("release_version"));
}
```

Query Builder


```
import static com.datastax.oss.driver.api.querybuilder.QueryBuilder.*;

try (CqlSession session = CqlSession.builder().build()) {
    Select query = selectFrom("system", "local").column("release_version");
    SimpleStatement statement = query.build();
    ResultSet rs = session.execute(statement);
    Row row = rs.one();
    System.out.println(row.getString("release_version"));
}
```

```
// SELECT first_name,last_name FROM ks.user WHERE id=?

Select select = selectFrom("ks", "user")
        .column("first_name")
        .column("last_name")
        .whereColumn("id").isEqualTo(bindMarker());
```

Datentyp Counter

- Spezieller Datentyp zum Zählen im Cluster
- (theoretisch) nicht 100% genau
- Nicht idempotent
- Nur eine Counter Spalte pro Tabelle
- Alle anderen Spalten müssen PRIMARY KEY sein
- Initial 0
- Nur Deltas mit + und -

SCOOP SOFTWARE

Lightweight Transactions

- Compare and Set (CAS)
- Benötigt Roundtrips zwischen den CAS coordinators
- Erhöhte Latenz

```
-- LWT
UPDATE cycling.cyclist_name
SET age = 34
WHERE id = 4647f6d3-7bd2-4085-8d6c-1229351b5498
IF age = 33;
```

```
-- LWT
INSERT INTO cycling.race_winners ( race_name, race_position, cyclist_name )
VALUES ( 'National Championships South Africa WJ-ITT (CN)', 7, { firstname: 'Joe', lastname: 'Anderson' } )
IF NOT EXISTS;
```

SCOOP SOFTWARE

Sonderfall: Geld zählen

- Deltas schreiben mit Counter
- Lightweight Transactions
  - Lesen
  - Im Program rechnen
  - Neuen Wert schreiben wenn alter Wert noch in DB steht
- Protokoll
  - Deltas schreiben
  - Beim Lesen Summe bilden
  - Aggregatsfunktion (sum) von Cassandra

SCOOP SOFTWARE

Praxis

SCOOP SOFTWARE

Produkt Dokumentation (DSE)

- Developer Documentation
- Architecture Guide
- Administrator Guide

- Meinung: “Sehr gute Dokumentation”

- Eine Person kann nahezu die komplette Dokumentation abdecken

https://docs.datastax.com/

SCOOP SOFTWARE

04
Apache Solr / DSE Search

SCOOP SOFTWARE

Apache Solr

Solr

- **Volltextsuche**
- Fehlertolerant
- schnell
- Komplexer suchen
  - Umkreissuche
  - Zeit
  - Ähnlichkeitssuchen
  - Sortierung (auch nach Gewichtung)

SCOOP SOFTWARE

DSE Search

- Create search Index on Table
- Suche mittels nativem SELECT möglich
- Suche mittels „solr_query“ in WHERE clause

SCOOP SOFTWARE

Such Index auf Cassandra Tabelle anlegen

```
CREATE SEARCH INDEX IF NOT EXISTS ON event
WITH COLUMNS
        total_score {docValues:true},
        event_status {docValues:true},
        event_type {docValues:true},
        event_link {docValues:true},
        dfp_exact_id {docValues:true},
        dfp_smart_id {docValues:true},
        dfp_fraudscore,
        dfp_fraudscore_rulematches,
        customer_account_number {docValues:true},
        customer_account_payment_account {docValues:true},
        customer_account_email {docValues:true},
        customer_account_phone_number {docValues:true},
        order_line_items,
        order_addresses,
        top_article_number,
        event_creation_time
    AND OPTIONS { lenient: true, reindex: false};
```

SCOOP SOFTWARE

Suche auf Cassandra Tabelle

```
SELECT id FROM nhanes_ks.nhanes WHERE
solr_query='{"q":"ethnicity:Mexi*", "sort":"id asc"}' LIMIT 3;


SELECT id FROM fraud_cases WHERE solr_query= '{'q':'{!tuple v=\'{!geofilt
sfield="order_addresses.geocode" pt="7.08006,50.912817" d="0.3" } \'}';


String solrQuery = "{" +
        "\'q\':" +
        "\'-dfp_token:\"" + fraudScoreRecord.getDfpToken() + "\" AND ( " +
        "event_link:\"" + fraudScoreRecord.getEventLink() + "\"^100000 OR +
        "dfp_exact_id:\"" + fraudScoreRecord.getDfpExactId() + "\"^256 OR " +
        "rmc_person_number:\"" + fraudScoreRecord.getRmcPersonNumber() + "\"^128 OR " +
        "customer_account_number:\"" + fraudScoreRecord.getCustomerAccountNumber() + "\"^64 OR " +
        "dfp_smart_id:\"" + fraudScoreRecord.getDfpSmartId() + "\"^32 OR " +
        "customer_account_payment_account:\"" + fraudScoreRecord.getCustomerPaymentAccount() + "\"^16 OR " +
        "customer_account_email:\"" + fraudScoreRecord.getCustomerAccountEmail() + "\"^8 " +
        "customer_account_phone_number:\"" + fraudScoreRecord.getCustomerAccountPhoneNumber() + "\"^4 OR " +
        "customer_account_email:\"" + fraudScoreRecord.getCustomerAccountEmail() + "\"^2 " +
        ") \'" +
        ",\'sort\':\'score DESC\‘}“;
```

SCOOP SOFTWARE

04
Der Schlussteil

SCOOP SOFTWARE

Fazit

- Cassandra alleine macht einiges gut
- Zusatzfunktionalität nötig
- DSE Integration ist hilfreich
- Testcluster wichtig

Kontakt

SCOOP Software GmbH
Gut Maarhausen
Eiler Straße 3P
D-51107 Köln

Stefan Solich
stefan.solich
@scoop-software.de
www.scoop-software.de