FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Fachbereich 2
Informatik und Ingenieurwissenschaften
Im Studiengang Informatik

Aufbau eines Clusters mit der NoSQL-Datenbank MongoDB auf Basis von Einplatinencomputern

Kolloquium zur Bachelorthesis

Danijel Klaic

Matrikelnummer 996907

1.Betreuer: Prof.Dr. Baun

2.Betreuer: Prof.Dr. Gabel

Frankfurt, 15. September 2016

Agenda

- Einleitung
- Technische Grundlagen
- Hardware- und Softwarekomponenten
- Implementierung
  - Replikation
  - Sharding
- Aggregationsmethoden
- Fazit

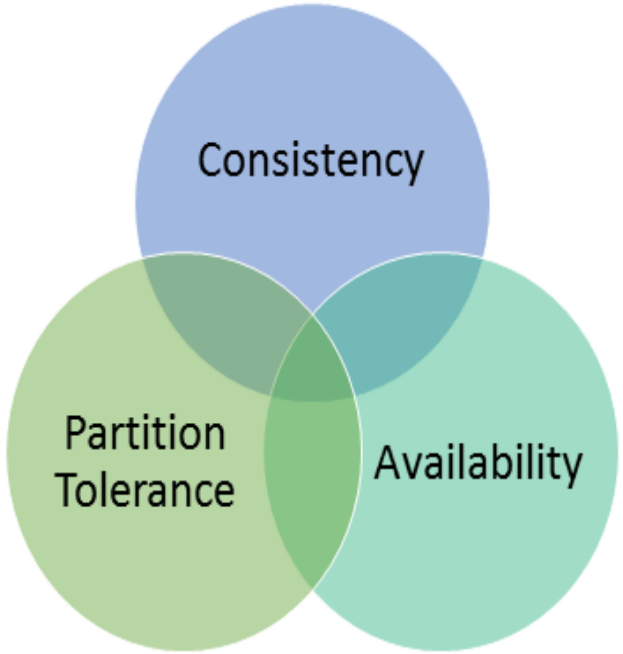
Einleitung

- Relationale Datenbanken
  - Edgar Frank Codd entwickelte die ersten relationale Datenbanken in 1960er und 1970er Jahren.
  - Datenbanksprache SEQUEL entstand in den 1970er Jahren.
- Big Data
  - Volume (Datenvolumen)
  - Variety (Vielfalt an Datenformaten)
  - Velocity (Geschwindigkeit)
- NoSQL-Datenbanken
- MongoDB
  - hu**MONGO**us

NoSQL-Definition

- Kein relationales Datenmodell
- Eignung für Systeme mit verteilter und horizontaler Skalierbarkeit
- Quelloffene Software
- Frei von Schemen oder nur von schwächeren Restriktionen betroffen
- Einfache Datenreplikation zur Unterstützung der verteilten Architektur
- Einfache Programmierschnittstelle
- Kein ACID als Konsistenzmodell

CAP-Theorem



- **Consistency**

Alle Knoten verfügen gleichzeitig über die gleichen Daten.

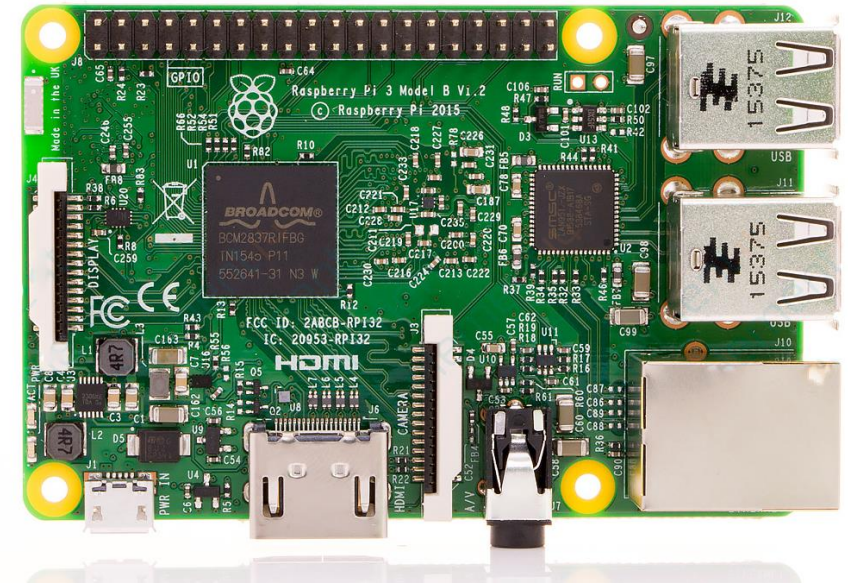
- **Availbility**

Lese- und Schreibzugriffe sind jederzeit verfügbar. Ein Knotenausfall wirkt sich nicht aus auf die Verfügbarkeit.

- **Partition Tolerance**

Das System funktioniert auch bei einem Ausfall einzelner Knoten.

Raspberry Pi 3 Modell B



- Prozessor: Broadcom BCM2837
- CPU: 1.2GHz 64-bit quad-core ARMv8
- GPU: 400 MHz VideoCore IV
- Arbeitsspeicher: 1 GB RAM LPDDR2-900 SDRAM
- Kommunikationsarten: LAN 10/100 Mbps, 802.11n Wireless LAN und Bluetooth 4.1
- Schnittstellen: 4 x USB 2.0, 1 x HDMI, 1 x CSI Camera Port, DSI Display Port, Stereo Out, microSD Port
- Abmessungen: Maße 85,6 x 56,5 Millimeter
- Gewicht: Ca. 45 Gramm

Bildquelle: https://javatutorial.net/wp-content/uploads/2016/03/RaspberryPi-3-Model-B.jpg

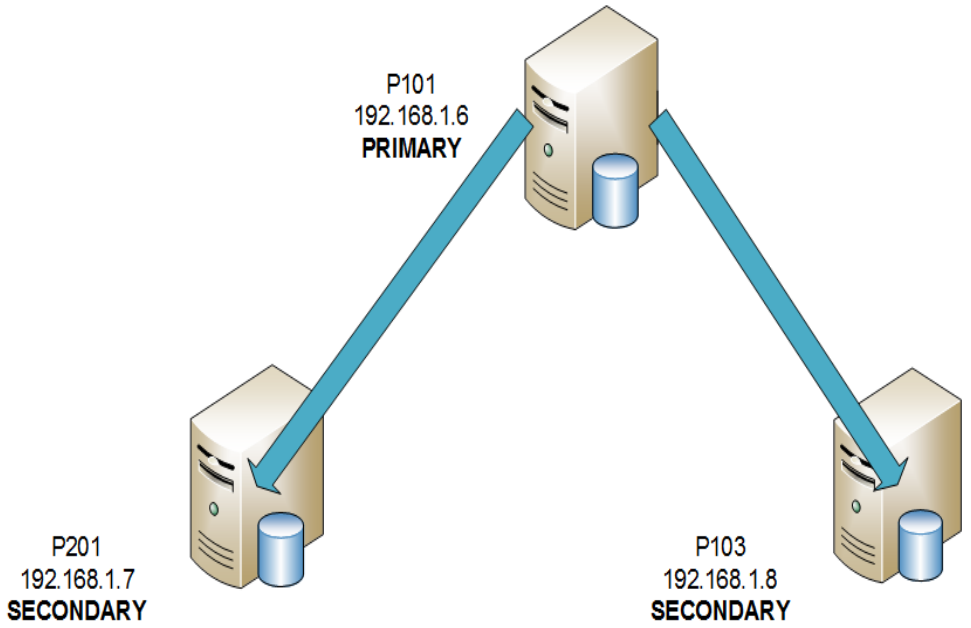
Hardwarekomponenten

- 3 x Raspberry Pi 3 Modell B
  - SD-Karte
- Router
- USB-Hub
- 3 x Netzwerkkabel
- 3 x USB-Stromkabel

Softwarekomponenten

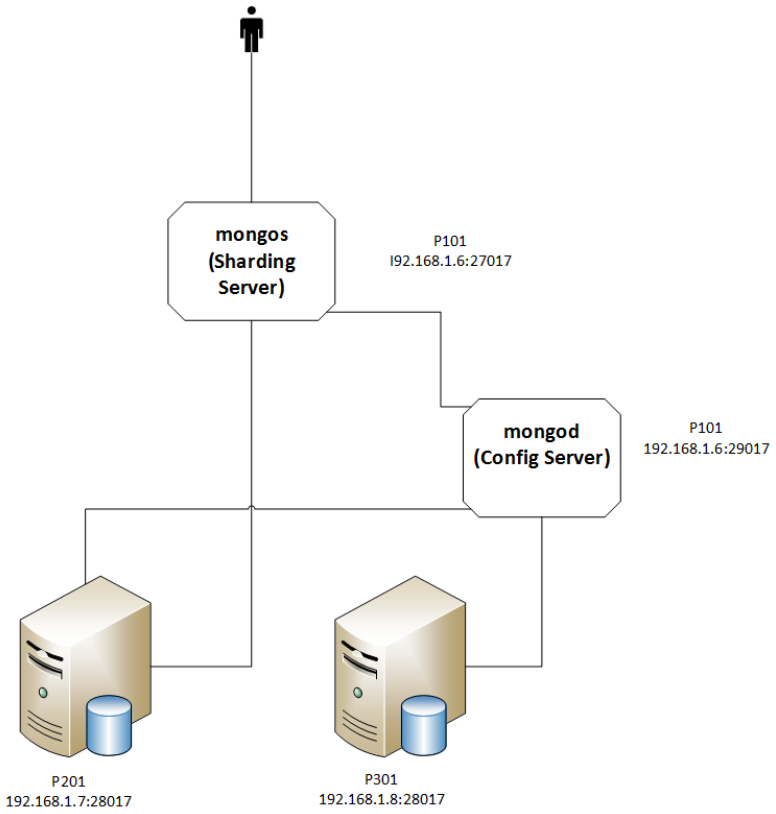
- Win32DiskManager
  - Bootfähige SD-Karte
- Raspbian Jessy
- PuTTY und Remotedesktopverbindung (Windows)
  - Zugriff auf die Eingabekonsole von Raspberry Pi
  - Zugriff auf die Benutzeroberfläche von Raspberry Pi
- MongoDB Version 3.0.9
  - 32-Bit Version
- Robomongo
  - Zugriff auf die Benutzeroberfläche von MongoDB

Replikation



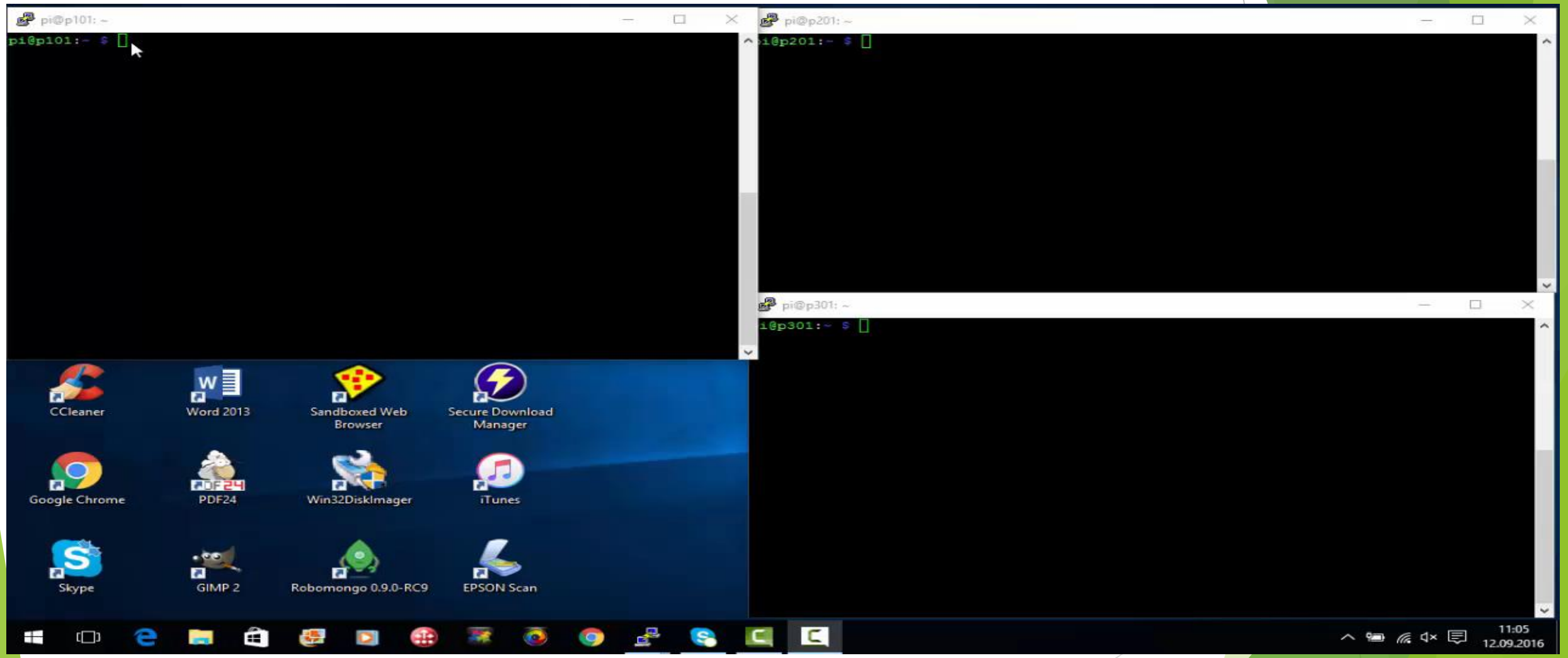
- Steigert die Ausfallsicherheit von Daten.
- Primärserver besitzt sowohl Lese- als auch Schreibrechte.
- Primärserver repliziert die Daten auf den Sekundärserver.
- Beim Ausfall eines Primärservers wird ein Sekundärserver zum neuem Primärserver gewählt.

Sharding

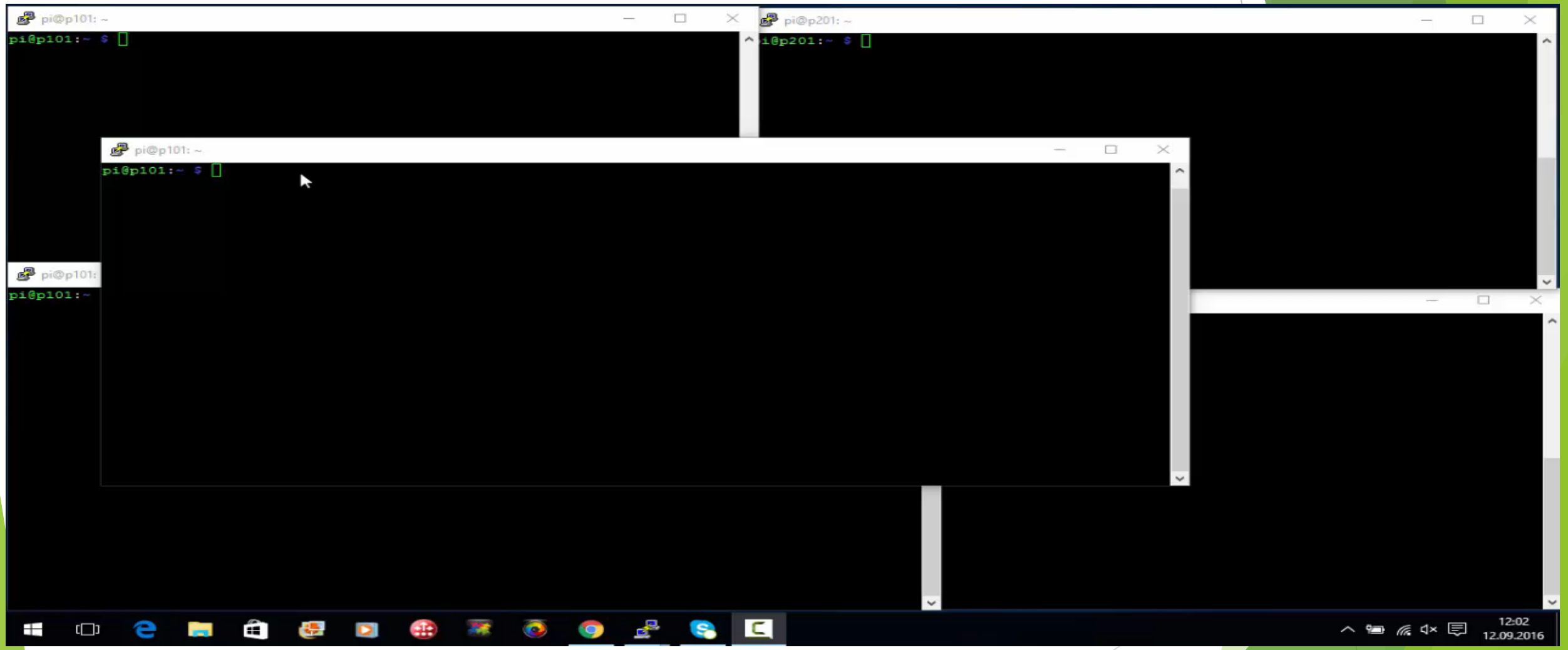


- Partitionierung einer Datenbank
- Große Datenmengen lassen sich in einzelnen Knoten verteilen.
- Chunk= Teilmenge von Daten
- Sharding-Keys bestimmen die Aufteilung von Daten.
- Empfohlen in Verbindung mit Replikation.

Vorführung des MongoDB-Clusters (Replikation)



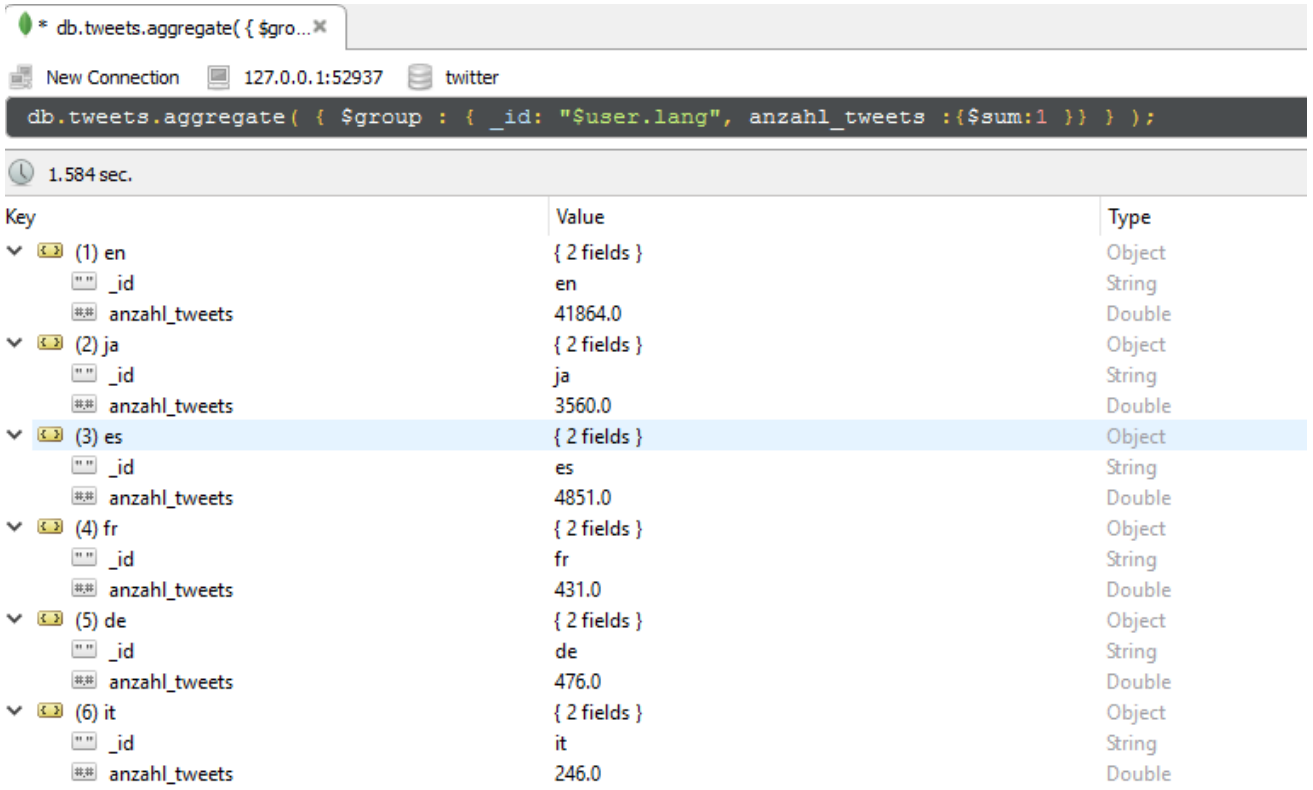
Vorführung des MongoDB-Clusters (Sharding)



Aggregation von Daten

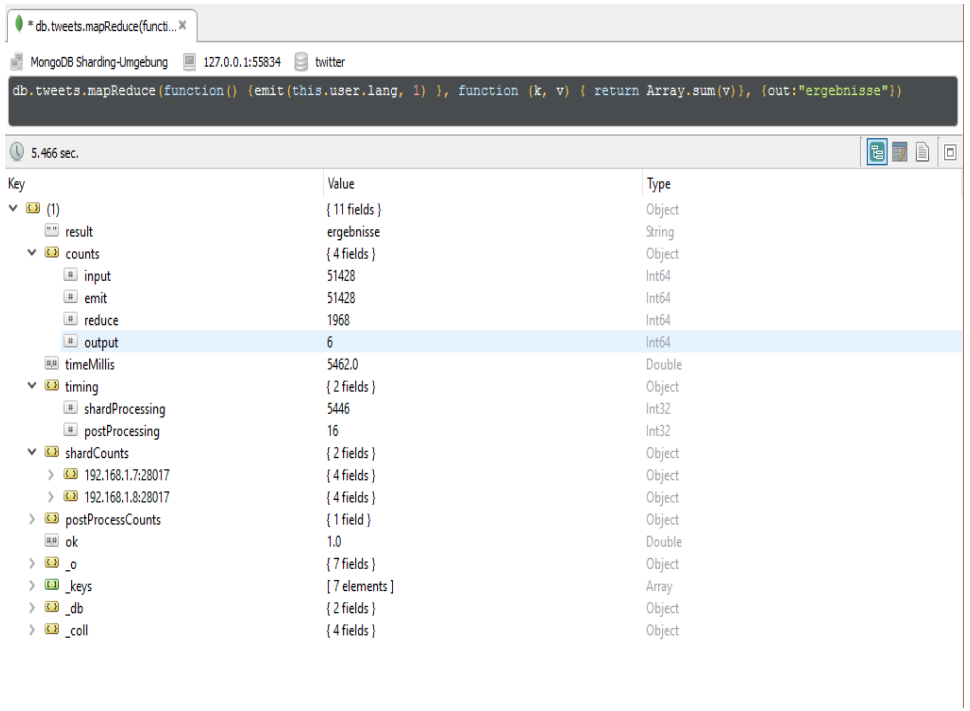
- Einfache Abfragemethoden
  - count()
  - distinct()
  - group()
- Aggregation Pipeline
- MapReduce

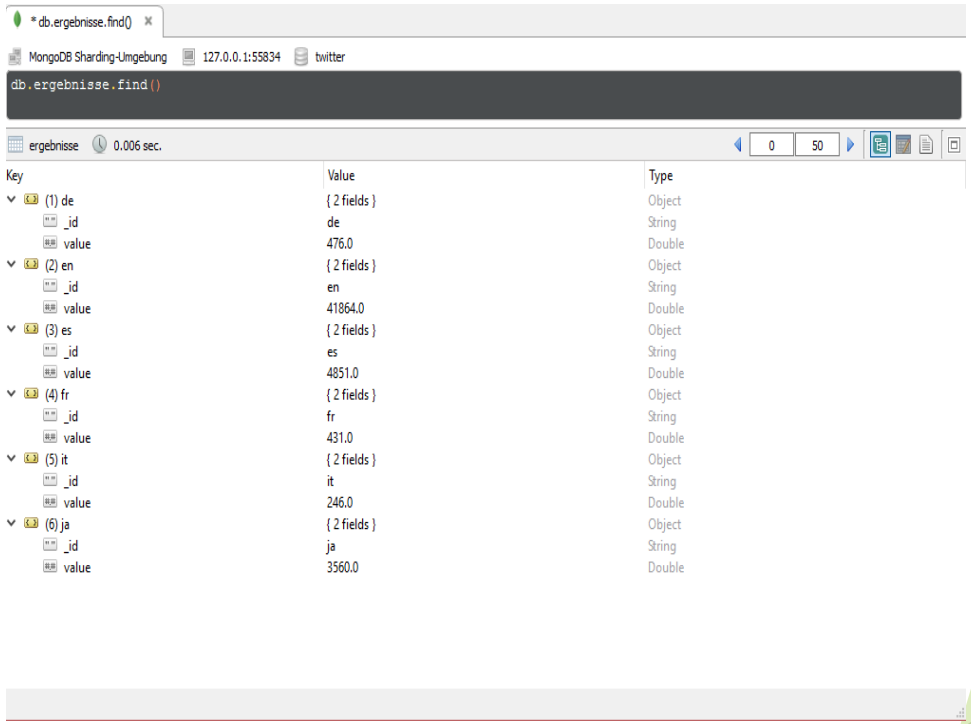
Aggregation Pipeline



Key	Value	Type
(1) en	{ 2 fields }	Object
_id	en	String
anzahl_tweets	41864.0	Double
(2) ja	{ 2 fields }	Object
_id	ja	String
anzahl_tweets	3560.0	Double
(3) es	{ 2 fields }	Object
_id	es	String
anzahl_tweets	4851.0	Double
(4) fr	{ 2 fields }	Object
_id	fr	String
anzahl_tweets	431.0	Double
(5) de	{ 2 fields }	Object
_id	de	String
anzahl_tweets	476.0	Double
(6) it	{ 2 fields }	Object
_id	it	String
anzahl_tweets	246.0	Double

MapReduce


* db.tweets.mapReduce(functi...
MongoDB Sharding-Umgebung
127.0.0.1:55834
twitter
db.tweets.mapReduce(function() {emit(this.user.lang, 1) }, function (k, v) { return Array.sum(v)}, {out:"ergebnisse"})
5.466 sec.
Key
Value
Type
(1)
{ 11 fields }
Object
result
ergebnisse
String
counts
{ 4 fields }
Object
input
51428
Int64
emit
51428
Int64
reduce
1968
Int64
output
6
Int64
timeMillis
5462.0
Double
timing
{ 2 fields }
Object
shardProcessing
5446
Int32
postProcessing
16
Int32
shardCounts
{ 2 fields }
Object
192.168.1.7:28017
{ 4 fields }
Object
192.168.1.8:28017
{ 4 fields }
Object
postProcessCounts
{ 1 field }
Object
ok
1.0
Double
_o
{ 7 fields }
Object
_keys
[7 elements]
Array
_db
{ 2 fields }
Object
_coll
{ 4 fields }
Object


* db.ergebnisse.find()
MongoDB Sharding-Umgebung
127.0.0.1:55834
twitter
db.ergebnisse.find()
ergebnisse
0.006 sec.
0
50
Key
Value
Type
(1) de
{ 2 fields }
Object
_id
de
String
value
476.0
Double
(2) en
{ 2 fields }
Object
_id
en
String
value
41864.0
Double
(3) es
{ 2 fields }
Object
_id
es
String
value
4851.0
Double
(4) fr
{ 2 fields }
Object
_id
fr
String
value
431.0
Double
(5) it
{ 2 fields }
Object
_id
it
String
value
246.0
Double
(6) ja
{ 2 fields }
Object
_id
ja
String
value
3560.0
Double

Fazit

- Analyse von verteilten Funktionalitäten.
- Neue Server lassen sich einfach und schnell in einem vorhandenen Cluster einbinden.
- Die Nachfrage nach NoSQL-Datenbanksystemen steigert sich.
- Besonders im Bereich der Webentwicklung eignen sich unterschiedliche Datenstrukturen.

Vielen Dank für Ihre Aufmerksamkeit!