

Kapitel 1

NoSQL- Datenbanken

NoSQL- Datenbanken machen heutzutage ein Revolution.

1.1 MongoDB

siehe[Hol12], siehe [Llo13] Von Geodaten bis NoSQL: Leistungsstarke PHP-Anwendungen
<http://ezproxy.bib.fh-muenchen.de:2125/doi/book/10.3139/9783446431225>

Dokumentdatenbanken gruppieren die Daten in einem strukturierten Dokument, typischerweise in einer JSON-Datenstruktur. Auch MongoDB verfolgt diesen Ansatz und bietet darauf aufbauend eine reichhaltige Abfragesprache und Indexe auf einzelne Datenfelder. Die Möglichkeiten der Replikation und des Shardings zur stufenlosen und unkomplizierten Skalierung der Daten und Zugriffe macht MongoDB auch für stark frequentierte Websites äußerst interessant.

MongoDB bietet zudem auch Map/Reduce zur Datenverarbeitung an. Map/Reduce ist ein Kernbaustein zur Verarbeitung großer Datenmengen (auch in anderen NoSQL-Datenbanken), da mit diesem Ansatz die Berechnungen auf einfachem Wege auf viele Server zur parallelen Berechnung verteilt werden können. Mehr dazu erfahren Sie in Abschnitt 14.6. Siehe auch weitere Kapitel wie 14.2 Datenbanken und Datenschema

Die nicht-relationale Datenbank 'MongoDB' macht mit einem effizienten Dokumentorientierten Ansatz, einfacher Skalierbarkeit und hoher Flexibilität dem bewährten MySQL-System zunehmend Konkurrenz. Der Begriff NoSQL steht, anders als zunächst von den meisten erwartet, nicht für "kein SQL", sondern für "nicht nur SQL"(Not only SQL). Ziel des neuen Ansatzes ist es vielmehr, relationale Datenbanken sinnvoll zu ergänzen wo sie

Defizite aufzeigen. Entstanden ist dieses Konzept in erster Linie als Antwort zur Unflexibilität, sowie zur relativ schwierigen Skalierbarkeit von klassischen Datenbanksystemen, bei denen die Daten nach einem stark strukturierten Modell gespeichert werden müssen.¹

In Zeiten von Web 2.0, Cloud Computing und sozialen Netzen wachsen die Datenmengen bei Web-Anwendungen dramatisch. Damit steigen die Anforderungen an Datendurchsatz und Skalierbarkeit der darunter liegenden Datenbanken immer weiter an. Ein Paradebeispiel für diese Problematik ist Twitter. Wie der US-Blog "TechCrunch" im Juni 2011 berichtete, muss die IT-Landschaft des populären Informationsnetzes mittlerweile rund 200 Millionen Tweets am Tag bewältigen. Ein Jahr zuvor waren es "nur" 65 Millionen. Bei solchen datenintensiven Applikationen leiden relationale Datenbanken üblicherweise unter Leistungsproblemen und stoßen oft an ihre Grenzen. Aus diesem Grund setzt Twitter neben MySQL mittlerweile auch auf NoSQL-Lösungen, um die Verfügbarkeit und Performance seines Dienstes zu verbessern.

Das System 'MongoDB' wurde 2009 vom amerikanischen Startup 10gen nach rund zwei Jahren Entwicklung der Öffentlichkeit als Open-Source-Lösung vorgestellt. Der etwas gewöhnungsbedürftige Name stammt von dem englischen Begriff "humongous", der sich ins Deutsche mit 'gigantisch' beziehungsweise "riesig" übersetzen lässt. Die Lösung basiert auf der Programmiersprache C++ und ist für die Betriebssysteme Windows, Mac OS X und Linux erhältlich. Sowohl 32-Bit- als auch 64-Bit-Systeme werden unterstützt.

Wie der Hersteller erklärt, ist die Lösung auf starke Leistung, große Datenmengen, hohe Flexibilität sowie einfache Skalierbarkeit ausgelegt.

1.2 Tabellen haben ausgedient - Dokumente als Datensätze

Die nichtrelationale Datenbank MongoDB speichert Datensätze in Form von Dokumenten im JSON-Format. Diese sind Javascript-Objekten sehr ähnlich. Der entscheidende Unterschied zu MySQL besteht darin, dass MongoDB dokumentenorientiert arbeitet. Dokumentenbasierende Datenbanken sind auf eine schemafreie Struktur ausgelegt. Bei

¹NoSQL: <http://www.computerwoche.de/a/datenbanksysteme-fuer-web-anwendungen-im-vergleich,2496589>

MongoDB gibt es also kein festes Tabellenschema und dadurch beispielsweise auch keine zwingenden Relationstabellen und "Joins", die mit der Weiterentwicklung und dem Ausbau der Datenbank immer komplexer werden. Stattdessen lassen sich Relationen entweder direkt im Datensatz speichern oder bei Bedarf individuell bei der Datenabfrage erstellen. Dadurch ist die Datenstruktur wesentlich flexibler als bei MySQL und lässt sich einfach horizontal skalieren.

Statt von Tabellen spricht man bei MongoDB von Kollektionen (Collections). Jede Kollektion kann Dokumente beinhalten analog zu Zeilen beziehungsweise Datensätzen in einer MySQL-Tabelle. Dokumente werden im so genannten BSON-Format gespeichert und ausgegeben. Sie sind dann Javascript-Objekten sehr ähnlich. Das Format stammt von dem kompakten JSON-Format (Javascript Object Notation) ab und ist, wie das Präfix "Binary" (Binary JSON = BSON) andeutet, für eine Overhead-arme Darstellung von binären Datenobjekten ausgelegt. Jedes Dokument kann dabei eine beliebige Anzahl an Feldern besitzen, während eine verschachtelte Array-Struktur ebenfalls möglich ist. Zudem dürfen Dokumente auch innerhalb eines Dokuments gespeichert werden.

1.3 Introduction

Siehe in Deutsch: <https://www.iks-gmbh.com/assets/downloads/Einfuehrung-in-MongoDB-iks.pdf>

1.4 CRUD

1.5 Schema Design

1.6 Performance

Siehe in Deutsch über Indexes: https://books.google.de/books?id=kRUbDAAAQBAJ&pg=PA53&lpg=PA53&dq=index+in+mongodb+was+ist+da&source=bl&ots=80Kgw664kZ&sig=rEhHo3g4JRVAVXwUr_In5xzWB8c&hl=en&sa=X&ved=0ahUKEwiBvd_VsbfQAhVDtBQKH4_ASAQ6AEINjAC#

v=onepage&q=index%20in%20mongodb%20was%20ist%20da&f=false

1.7 Aggregation Framework

1.8 Application Engineering

1.9 Sharding

MongoDB bietet mit AutoSharding ein Feature, das es ermöglicht, einen Datenbankserver automatisch auf verschiedene physikalische Maschinen aufzuteilen und somit die Datenbank horizontal zu skalieren. Um das Sharding bei MongoDB zu konfigurieren, werden drei Komponenten benötigt...., siehe Sharding <https://www.iks-gmbh.com/assets/downloads/Einfuehrung-in-MongoDB-iks.pdf>

1.10 ODMs für MongoDB

ODM ist Object-Document Mapper für nichtrelationale Datenbanken wie MongoDB, Apache Cassandra etc.

1.10.1 Morphia

Morphia is the Java Object Document Mapper for MongoDB <http://mongodb.github.io/morphia/>

1.10.2 Doctrine

The Doctrine MongoDB ODM project is a library that provides a PHP object mapping functionality for MongoDB. <https://github.com/doctrine/mongodb-odm>

1.11 Beziehungen

Folgenden Relationen wie One-to-One **Teilabschnitt 1.11.1**, One-to-Many **Teilabschnitt 1.11.2** und Many-to-Many **Teilabschnitt 1.11.3** blabla

1.11.1 One-to-One

bla

1.11.2 One-to-Many

bla

1.11.3 Many-to-Many

bla

1.12 Storage Engines

1.12.1 MMAPv1

MMAPv1 is default a storage engine. MMAPv1 automatically allocates power-of-two-sized documents when new documents are inserted This is handled by the storage engine. MMAPv1 is built on top of the mmap system call that maps files into memory This is the basic idea behind why we call it MMAPv1.

1.12.2 WiredTiger

To invoke a WiredTiger storage engine, you would use:

Listing 1.12.1: Something else

```
> killall mongod
```

Listing 1.12.2: Something else

```
> mkdir WT
```

Listing 1.12.3: Something else

```
> mongod -dbpath WT --storageEngine wiredTiger
```

Dann mongo starten mit 'mongo', dann:

Listing 1.12.4: Something else

```
> db.foo.insert({'name':'andrew'})
```

Listing 1.12.5: Something else

```
> db.foo.stats()
```

-> man bekommt die Information über die Collection. Which of the following are features of the WiredTiger storage engine?

- Document-level concurrency
- Compression

1.13 Indizes

Index hilft, Datenbanken zu optimieren. Die nachfolgende Abbildung 1.1 demonstriert

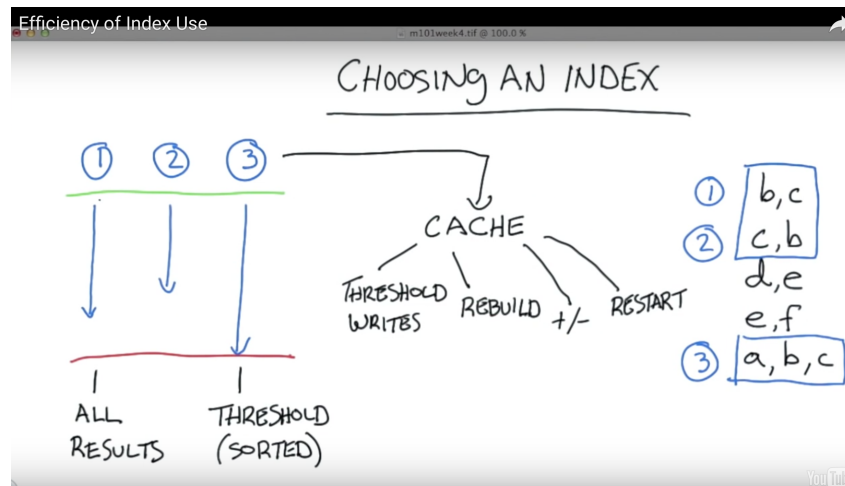


Abbildung 1.1: Efficiency of Index Use

<https://docs.mongodb.com/manual/indexes/> Neben dem obligatorischen Primär-Index auf dem Feld `_id`, das in jedem Dokument existieren und pro Collection eindeutig sein muss, können Sie in MongoDB bis zu 63 weitere Sekundär-Indizes pro Collection anlegen, um Suchanfragen zu beschleunigen. Ein Sekundär-Index kann auf einem einzelnen Feld oder einer Gruppe von Feldern angelegt werden.

Which optimization will typically have the greatest impact on the performance of a database?

Adding appropriate indexes on large collections so that only a small percentage of queries need to scan the collection.

1.13.1 Creating Indexes

blabla

Listing 1.13.1: Something else

```
> db.students.createIndex();
```

blabla

Listing 1.13.2: Something else

```
> db.students.explain().find();
```

Quiz: Please provide the mongo shell command to add an index to a collection named students, having the index key be class, student_name. Neither will go in the 1"direction..

Listing 1.13.3: Something else

```
> db.students.createIndex({student_name:1, class:1});
```

Listing 1.13.4: Something else

```
> db.students.dropIndex({student_name:1});
```

Multikey Indexes

blabla

Index Creation Option, Unique

für jedes attribut kann man Unique definieren, d.h. doppelte Werte dürfen nicht vorkommen

Listing 1.13.5: Something else

```
> db.students.createIndex({student_id : test}, {unique:true});
```

Please provide the mongo shell command to create a unique index on student_id, class_id, ascending for the collection students.

Listing 1.13.6: Something else

```
> db.students.createIndex({student_id:1, class_id:1}, {unique:true});
```


Index Creation, Sparse

Im Fall, wenn ein Attribut nicht in allen Dokumenten vorkommt, aber für dieses ein Unique Index definiert werden soll, muss Folgendes verwendet werden:

Listing 1.13.7: Something else

```
> db.students.createIndex({cell:1}, {unique:true, sparse:true});
```

blabla, siehe den Shellbefehl, blabla

Listing 1.13.8: Something else

```
> db.students.createIndex({student_id:1, class_id:1}, {unique:true});
```

siehe Codeauszug

1.14 Prototyp

Kommentare zu den Fotos hinzufügen, Eingebettete Kommentare, siehe <http://ezproxy.bib.fh-muenchen.de:2125/doi/pdf/10.3139/9783446431225.014>

1.15 Replikation und Verfügbarkeit

MongoDB kann Server in Replikationsgruppen anordnen, damit bei Ausfall eines Servers die Verfügbarkeit der Datenbank trotzdem gewährleistet ist. blabla

1.16 Aggregation Framework

How good is it? Mapping between SQL and Aggregation

1.17 Fazit

Siehe Listing 1.13.8

Doch wie der Begriff Not only SQL (NoSQL) andeutet, stehen beide Datenbanksysteme nicht unbedingt in direkter Konkurrenz zueinander, sondern können sich gegenseitig ergänzen. Dennoch, wenn es um die persistente Datenspeicherung bei Web-Anwendungen geht, stellen relationale Datenbanken nicht mehr die einzige Alternative dar. Bei eigenen Projekten wären Entwickler heute also gut beraten, die Vor- und Nachteile der beiden Systeme gegenüberzustellen und entsprechend den eigenen Anforderungen und Prioritäten zu bewerten. Muss das System mit großen Datenmengen effizient umgehen können? Werden hohe Anforderungen an Skalierbarkeit und Flexibilität der Datenbank gestellt? Sollen sich die Daten über mehrere Server verteilen lassen? Sind häufige Änderungen an der Datenstruktur in Zukunft zu erwarten? Wenn Sie die meisten dieser Fragen mit "Ja" beantworten, dann sollten Sie sich MongoDB zumindest näher anschauen.

Daten in MongoDB verfügen über ein flexibles Schema. Kollektionen (=Collections) erzwingt keine Struktur.

Listing 1.17.1: Verbindungsaufbau

```
1 public static void main(String[] args) {
2
3     MongoClient mongoClient = new MongoClient("localhost", 27017);
4     MongoDBDatabase db = mongoClient.getDatabase("test");
5     MongoCollection<Document> collectionOfZips = db.getCollection("zips");
6
7     // weitere CRUD-Operationen mit der ausgewählten Kollektion
8 }
```

Test 1.17.1

1.18 Apache Cassandra

Kapitel 2

Kap2

asdasdasdasd[:ae] ölköklök[Abubakar:2016aa]

2.1 Main

löasdlöaskd

2.1.1 Test

test

Literaturverzeichnis

- [Hol12] Arno Hollosi. *Von Geodaten bis NoSQL: leistungsstarke PHP-Anwendungen: Aktuelle Techniken und Methoden für Fortgeschrittene*. München: Hanser, 2012. ISBN: 9783446431225. DOI: 10 . 3139 / 9783446431225. URL: [http : / / www . hanser-elibrary.com/isbn/9783446429956](http://www.hanser-elibrary.com/isbn/9783446429956).
- [Llo13] Eva María Lloret Ivorra. *Con gusto: Spanisch*. 1. Aufl., [5. Nachdr.] Stuttgart: Klett Sprachen, 2013. ISBN: 3125149800.

Abbildungsverzeichnis

1.1	Efficiency of Index Use	7
-----	-----------------------------------	---

Quelltextverzeichnis

1.12.1 Something else	6
1.12.2 Something else	6
1.12.3 Something else	6
1.12.4 Something else	6
1.12.5 Something else	6
1.13.1 Something else	7
1.13.2 Something else	7
1.13.3 Something else	8
1.13.4 Something else	8
1.13.5 Something else	8
1.13.6 Something else	8
1.13.7 Something else	9
1.13.8 Something else	9
1.17.1 Verbindungsaufbau	10