

Examensarbete
15 högskolepoäng, grundnivå

NoSQL-databaser i ett socialt nätverk.

Ragnvald Persson

Sammanfattning

Syftet med studien är att göra en fördjupning inom NoSQL-databaser och undersöka vilka uppgifter som de olika NoSQL-grupperna passar bäst till i ett socialt nätverk, som t.ex. Facebook och Twitter. Det finns fyra olika typer av NoSQL-databaser: kolumndatabaser, grafdatabaser, nyckelvärde-databaser och dokumentdatabaser. Frågan är vilken NoSQL-databas ska man välja till en viss uppgift i ett givet socialt nätverk. När man ska utveckla ett socialt nätverk, som kräver lagring av data, är det viktigt att känna till vilken typ av databas som bör användas till en viss typ av uppgift.

För att få svar på frågorna har det gjorts en undersökning över vad tidigare forskning har kommit fram till. Det har även gjorts en praktisk studie med alla fyra NoSQL-grupper i ett experiment med lagring av användaruppgifter, meddelanden och vänner.

Abstract

The purpose of the study is to deepen within NoSQL databases and investigate what tasks the different NoSQL groups fit best in a social network, such as Facebook and Twitter. The data is, for example, about the storage of personal data or social networking. There are four different types of NoSQL databases: column databases, graph databases, key value databases and document databases. The question is which NoSQL database should be chosen for a particular task in a given social network. When developing a social network that requires data storage, it is important to know what kind of database should be used for a certain type of task.

In order to answer the questions, an investigation has been made of what previous research has reached. There has also been a practical study of all four NoSQL groups in an experiment with storing user information, messages and friends.

Innehåll

Sammanfattning	2
Abstract	2
1 Inledning.....	1
1.1 Bakgrund	1
1.2 Syfte och forskningsfråga.....	1
1.3 Avgränsningar	1
2 Metod	2
2.1 Mixed methodology	2
2.2 Kvalitativ metod	2
2.2.1 Söka artiklar	2
Sökning 1	3
Sökning 2	3
Sökning 3	4
Sökning 4	4
Sökning 5	5
Sökning 6	6
Sökning 7	7
Sökning 8	7
Sökning 9	8
Sökning 10	8
2.2.2 Övriga artiklar.....	9
2.3 Kvantitativ metod.....	9
2.3.1 Redovisning av metod.....	9
2.3.2 Simulering	10
3. Resultat av litteraturstudie.....	12
3.1 Nyckelvärde-databas.....	12
3.2 Dokumentdatabas	13
3.3 Grafdatabaser	16
3.4 Kolumndatabaser	17
4 Resultatet av den praktiska studien.....	20
4. Analys och diskussion	23
4.1 Analys av den praktiska studien.....	23
4.2 Analys av litteraturstudien	24

4.3 Diskussion	24
4.4 Metoddiskussion	25
5. Svaret på forskningsfrågor.....	26
6. Slutsatser och vidare forskning	26
7. Källor.....	27
Bilaga 1.....	32
MongoDB.....	32
Apache Cassandra.....	39
Neo4j.....	49
Redis.....	59

1 Inledning

1.1 Bakgrund

Databaser utvecklades för att på ett smidigt sätt kunna lagra data till IT-system. Först fanns relaterade databaser, som också kallas för SQL-databaser. "SQL står för Structed Query Language" [2]. "MySQL", Microsoft SQL-server och Oracle är exempel på relationsdatabaser. Problemet med relaterade databaser är att de inte är praktiska när det handlar om skalbarhet på systemet eller lagring av stora datamängder[1]. Idag har det utvecklats flera olika kategorier inom "NoSQL-databaser", NoSQL betyder "not only SQL". Det finns kategorier inom NoSQL-databaser[1]: "grafdatabaser", "kolumndatabaser", "nyckelvärddatabaser" och "dokumentdatabaser". När man ska utveckla ett IT-system, som ska lagra data, så bör man känna till vilken databas som passar bäst för en viss uppgift. Den här uppsatsen beskriver vilka NoSQL-databaser som passar till de olika uppgifterna i ett socialt nätverk.

1.2 Syfte och forskningsfråga

Det finns flera alternativ när det gäller val av NoSQL-databas. De NoSQL-databaser som finns är grafdatabaser, kolumndatabaser, dokumentdatabaser och nyckelvärddatabaser. Idag använder vi sociala nätverk för att kunna kommunicera med varandra. Facebook, LinkedIn och Twitter är exempel på sociala nätverk och måste, bland annat, lagra användaruppgifter och meddelande i en databas.

Alla kategorier inom NoSQL används till sociala nätverk. Syftet med studien är att ge läsaren en förståelse över vilken NoSQL-databas som är bästa valet för en viss uppgift i ett socialt nätverk. Genom att göra en litteraturstudie och en praktisk studie så blir resultatet förhoppningsvis en bättre förståelse över vilken databas som bör användas till en viss uppgift i ett socialt nätverk.

Frågeställningarna är:

- Vad har de olika NoSQL-databaserna för egenskaper, som är relevanta till ett socialt nätverk?
- Vilka uppgifter passar bäst för de olika NoSQL-databaserna i ett socialt nätverk?
- Vilka/Vilken NoSQL-databas passar till ett socialt nätverk, när det handlar om att hantera användaruppgifter, meddelande och vänner?

1.3 Avgränsningar

Till den praktiska studien har det valts ut en databashanterare från varje kategori av NoSQL-databaserna: Redis (Nyckelvärddatabas), Apache Cassandra (Kolumndatabas), MongoDB (Dokumentdatabas) och Neo4j (Grafdatabas). Dessa databashanterare har installerats på en

PC-maskin. Studien har fokuserat på följande kommandon, vanliga i de flesta sociala nätverk, som hanterar användaruppgifter, meddelande och vänner:

- Inläsning av meddelanden
- Inläsning av vänner
- Inloggning av användare
- Registrering av användare

Till den praktiska studien så utvecklades fyra program som kan koppla upp sig mot var sin databashanterare med hjälp av importerade biblioteksfiler. Varje program innehåller ett antal metoder för att kunna skriva/läsa data från databashanteraren. Förutom att kommunikation med databashanteraren så kunde även programmen mäta tiden det tar att utföra kommandon. Hela experimentet har genomförts på en PC-maskin, där programmen och databasservernarna har varit i bruk och där data har lagrats.

2 Metod

2.1 Mixed methodology

För att få fram resultat till forskningen används en "mixed methodology", dvs. en blandad metod [45]. Det är egentligen inte en metod utan en blandning av flera metoder. I det här fallet så används en "kvalitativ metod" och en "kvantitativ metod".

2.2 Kvalitativ metod

Det finns många källor med kunskaper om NoSQL så därför har en "kvalitativ metod" valts ut [47], vilket innebär insamling och analys av information från tidigare undersökningar. Det har redan gjorts undersökningar och experiment med NoSQL-databaser i olika datasystem. Insamling av uppgifter kommer från vetenskapliga artiklar och har samlats in från databasen "IEEE Xplore", som innehåller vetenskapliga artiklar relaterat till datavetenskap. Artiklarna i IEEE Xplore är "peer-reviewed", vilket innebär att de har gått igenom en granskning som gör de trovärdiga och är godkända som vetenskapligt material.

Det söktes artiklar mellan 2010 och 2017. Det skulle hitta artiklar som innehåller den senaste informationen. Genom att skriva AND mellan söktermerna så hittas artiklar som innehåller alla sökord. Sökorden skulle antingen finnas i artiklarnas "keywords" eller "abstrakt".

Artiklarna, som man fick fram i sökresultatet granskades först efter titeln. Sedan granskades artikelns abstrakt, som beskriver kort artikelns innehåll och slutsats. Artiklar som hade relevant information efter granskning av abstraktet samlades in för en mer detaljerad granskning.

2.2.1 Söka artiklar

Här redovisas resultatet av sökningarna som gjordes i IEEE Xplore. Sökresultatet redovisas i en tabell och i en lista, med funna artiklar, under tabellen. I tabellerna finns tre kategorier.

”Resultat” visar antalet artiklar, som matchar sökningen. ”Efter titeln” visar antalet artiklar som har blivit utvalda efter titeln. ”Efter abstrakt” visar antalet artiklar som har blivit utvalda efter artikelns abstrakt. Under tabellen så finns en lista med titlarna på de artiklar som har valts ut efter att deras abstrakt har blivit granskat. Två sökningar gjordes med samma söktermer. Först i artiklarnas abstrakt och sedan i artiklarnas keywords.

Sökning 1

Söktermer: NoSQL AND key-value

Sökning i: Abstrakt

IEEE	Resultat	Efter titel	Efter abstrakt
	43	5	5

- Reduce, You Say: What NoSQL Can Do for Data Aggregation and BI in Large Repositories
- Modeling and querying data in NoSQL databases
- NoSQL Systems for Big Data Management
- Analysis of various NoSQL database
- A performance comparison of SQL and NoSQL databases

Sökning i: Keywords

IEEE	Resultat	Efter titel	Efter abstrakt
	22	8	5

- A Flexible QoS Fortified Distributed Key-Value Storage System for the Cloud
- A multidimensional data storage model for location based application on Hbase
- Implementation of Data Transform Method into NoSQL Database for Healthcare Data
- Implementing Ultra-Low-Latency Datacenter Services with Programmable Logic
- TippyDB: Geographically-aware distributed NoSQL Key-Value store

Sökning 2

Söktermer: NoSQL AND document

Sökning i: Abstract

IEEE	Resultat	Efter titel	Efter abstrakt
	65	5	5

- Voluntary Geographic Information Systems with Document-based NoSQL Databases

- An implementation approach to store GIS spatial data on NoSQL database
- The experience of NoSQL database in telecommunication enterprise
- A Study Into the Capabilities of NoSQL Databases in Handling a Highly Heterogeneous Tree.
- A Study of Genomic Data Provenance in NoSQL Document-Oriented Database Systems

Sökning i: Keywords

IEEE	Resultat	Efter titel	Efter abstrakt
	18	3	0

Sökning 3

Söktermer: NoSQL AND column

Sökning i: Abstract

IEEE	Resultat	Efter titel	Efter abstrakt
	26	3	3

- Query processing over data warehouse using relational databases and NoSQL
- Ontology based data integration of NoSQL datastores
- Review of NoSQL Databases and Performance Testing on HBase

Sökning i: Keywords

IEEE	Resultat	Efter titel	Efter abstrakt
	9	0	0

Sökning 4

Söktermer: NoSQL AND graph

Sökning i: Abstract

IEEE	Resultat	Efter titel	Efter abstrakt
	40	5	5

- Analysis of Data Management and Query Handling in Social Networks using NoSQL Databases
- Representing History in Graph-Oriented NoSQL DataBases: A Versioning System
- Implementation of NOSQL for robotics

- Evaluation of NoSQL Graph Databases for Querying and Versioning of Engineering Data in Multidisciplinary Engineering Environments
- A comparison of NoSQL Graph Databases

Sökning i: Keywords

IEEE	Resultat	Efter titel	Efter abstrakt
	16	4	1

- 2Path: a terpenoid metabolic network modeled as graph database

Sökning 5

Söktermer: NoSQL AND performance

Sökning i: Abstract

IEEE	Resultat	Efter titel	Efter abstrakt
	27	4	3

- Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual
- Review of NoSQL databases and performance testing on HBase
- Performance Characterization of Modern Databases on Out-of-Order CPUs
- A Scalability Comparison Study of Data Management Approaches for Smart Metering Systems

Sökning i: Keywords

IEEE	Resultat	Efter titel	Efter abstrakt
	23	8	5

- EPC Information Services with No-SQL datastore for the Internet of Things
- On the Application and Performance of MongoDB for Climate Satellite Data
- Performance Characterization of Modern Databases on Out-of-Order CPUs
- Performance evaluation of SQL and MongoDB databases for big e-commerce data
- Scientific Computing Doesn't Need NoSQL

Sökning 6

Ett nytt försök gjordes genom att använda ett av sökorden. "NoSQL" lades till i söktermen tillsammans med ordet "Big Data", eftersom NoSQL används för att lagra stora datamängder.

Söktermer: NoSQL AND Big Data

Sökning i: Abstract

IEEE	Resultat	Efter titel	Efter abstrakt
	148	5	0

Sökning i: Keywords

IEEE	Resultat	Efter titel	Efter abstrakt
	82	32	18

- Optimization and application in medical big document-data of Apriori algorithm based on MapReduce
- A Study on Cloud Database
- Using of cloud computing, clustering and document-oriented database for enterprise content management
- Document-oriented data warehouses: Models and extended cuboids, extended cuboids in oriented document
- Big data stream computing in healthcare real-time analytics
- GRAPH/Z: A Key-Value Store Based Scalable Graph Processing System
- Challenges and best practices for enterprise adoption of Big Data technologies
- Big data processing framework of road traffic collision using distributed CEP
- Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications
- Big Data on the Internet of Things
- Learning ontology from Big Data through MongoDB database
- Model-Driven Observability for Big Data Storage
- A Big Data System Design to Predict the Vehicle Slip
- Road Traffic Big Data Collision Analysis Processing Framework
- Big Data architecture for IT incident management
- No SQL in practice: a write-heavy enterprise application
- Document-oriented Database-based Privacy Data Protection Architecture
- Handling Big Data Using NoSQL
- Survey on NoSQL database
- Digital Forensics and NoSQL Databases

Sökning 7

Söktermer: NoSQL AND performance

Sökning i: Abstract

En ny sökning gjordes med samma inställningar förutom att "Big Data byttes ut mot "performance". Även en del av resultaten var artiklar som redan hade hittats.

IEEE	Resultat	Efter titel	Efter abstrakt
	23	12	9

- Towards Performance Evaluation of Cloudant for Customer Representative Workloads
- Realization of secondary indexing to NoSQL database with intelligent adaptation
- A File-Based Approach for Recommender Systems in High-Performance Computing Environments
- Machine Learning Approach for Cloud NoSQL Databases Performance Modeling
- Benchmarking the Performance Impact of Transport Layer Security in Cloud Database Systems
- BigGIS: How big data can shape next-generation GIS
- WMS performance of selected SQL and NoSQL databases
- Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual
- A comparison between several NoSQL databases with comments and notes

Sökning 8

Söktermer: NoSQL AND Compare

Sökning i: Abstract

IEEE	Resultat	Efter titel	Efter abstrakt
	73	7	5

- Genomic Data Persistency on a NoSQL Database System
- Dynamo and BigTable – Review and comparison
- RDF vs. NoSQL databases for the semantic web applications
- Non-structure Data Storage Technology: A Discussion
- Andes: A Highly Scalable Persistent Messaging System

Sökning i: Keywords

IEEE	Resultat	Efter titel	Efter abstract
	0	0	0

Sökning 9

Söktermer: NoSQL AND Comparsion

Sökning i: Abstract

IEEE	Resultat	Efter titel	Efter abstract
	0	0	0

Sökning i: Keywords

IEEE	Resultat	Efter titel	Efter abstract
	0	0	0

Sökning 10

Söktermer: NoSQL AND availability

Sökning i: Abstract

IEEE	Resultat	Efter titel	Efter abstract
	64	9	4

- Medical imaging archiving: a comparison between several NoSQL solutions
- A NoSQL-Based Approach for Real-Time Managing of Embedded Data Bases
- Experimental Assessment of NoSQL Databases Dependability
- Scalable data management: NoSQL data stores in research and practice

Sökning i: Keywords

IEEE	Resultat	Efter titel	Efter abstract
	4	3	2

- Evaluation of the CAP Properties on Amazon SimpleDB and Windows Azure Table Storage
- A Multi-key Transactions Model for NoSQL Cloud Database Systems

2.2.2 Övriga artiklar

På grund av brist i strukturen i sökningen har det inte varit möjligt att definiera vilka sökord som har använts för att kunna hitta artiklarna nedan. Artiklarna har hittats i databaserna ACM och IEEE Xplore och har hittats med söktermer som inte har kunnat dokumenteras.

A Large-scale Online Search System of High-Dimensional Vectors Based on Key-Value Store.

Using Paxos to Build a Lightweight, Highly Available Key-value Data Store.

Cassandra - A Decentralized Structured Storage System.

CBDIR: Fast and Effective Content Based Document Information Retrieval System

Efter sökningen så granskades alla artiklar i en mer noggrann textanalys där själva texten granskades mer detaljerad.

2.3 Kvantitativ metod

2.3.1 Redovisning av metod

Kvantitativ metod innebär insamling av rådata från egna experiment/mätningar. Det finns möjlighet att installera databaserna och göra experiment för att få fram egna resultat [46]. Alla fyra grupper av NoSQL-databaser används till ett socialt nätverk och därför så ingår alla i den kvantitativa metoden. I alla grupper ingår ett antal databashanterare. Nedan redovisas de databashanterare som har använts till ett socialt nätverk:

Redis: Nyckelvärde-databas¹

Apache Cassandra: Kolumndatabas²

MongoDB: Dokumentdatabas³

Neo4j: Grafdatabas⁴

Den praktiska studien har fokuserats på att hantera användaruppgifter, meddelande och vänner i ett socialt nätverk. I ett socialt nätverk finns det användare som måste lagra sina uppgifter som namn och kontaktuppgifter. Sociala nätverk måste också lagra användarnas meddelanden och vänner. Användare ska läggas till och datauppgifter måste läsas in för användarens inloggning. Detta är de grundläggande funktionerna i ett socialt nätverk och det har varit möjligt att prova dessa funktioner. När man loggar in så sker det mest inläsning från databasen då man måste läsa in användarens meddelanden, vänner och användaruppgifter. Ju kortare tid det tar att göra inläsningen desto bättre. Därför så görs en studie över inläsning av ett stort antal meddelande, stort antal vänner och även användarens profil.

¹ <https://redis.io/>

² <http://cassandra.apache.org/>

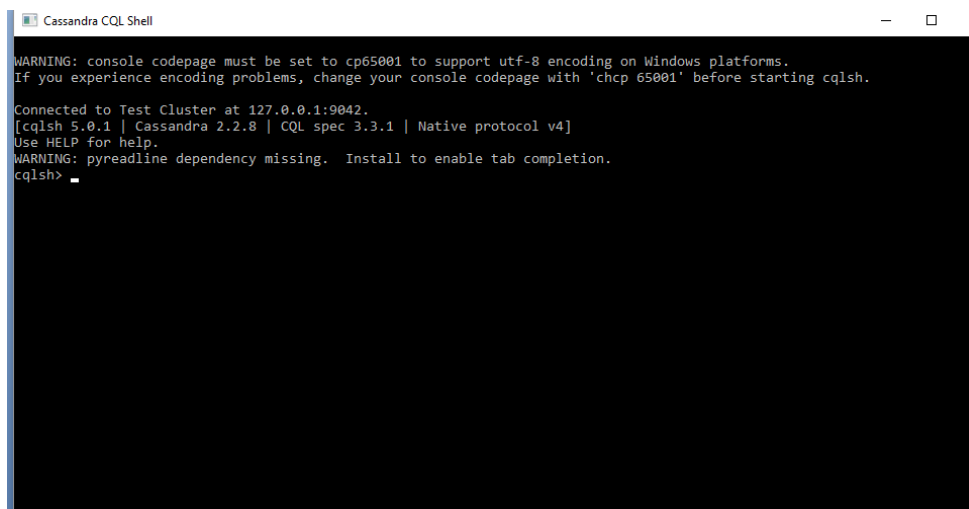
³ <https://www.mongodb.com/>

⁴ <https://neo4j.com/>

2.3.2 Simulering

Installation av simulering

Simuleringen består av fyra program som är uppkopplade mot var sin databashanterare. Alla simuleringar har körts på en PC-maskin med en 64-bitars version av Windows 10. Datorn har 16 GB RAM och en i7 processor med 2,60 GHz. Det första som gjordes var att installera en NoSQL-server mjukvara för varje databashanterare. Man kommunicerade med en server via ett gränssnitt, se Fig. 1.



```
Cassandra CQL Shell
WARNING: console codepage must be set to cp65001 to support utf-8 encoding on Windows platforms.
If you experience encoding problems, change your console codepage with 'chcp 65001' before starting cqlsh.

Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 2.2.8 | CQL spec 3.3.1 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing. Install to enable tab completion.
cqlsh> _
```

Fig. 1: Gränssnittet för kommunikation med servern. Bilden visar gränssnittet till Apache Cassandra.

För att utveckla programmen, som ska kommunicera med serverna så användes programmet "Eclipse Juno". Det behövde laddas ner JAR-filer till programmet. En JAR-fil är en biblioteksfil som innehåller information som behövs för att kommunicerar med serverna. JAR-filerna lades in i projektet i Eclipse. Serverna måste också vara igång för att programmet ska kunna kommunicera. För varje databashanterare så skapades en program med en programklass.

I varje programklass så skrevs det metoder, se Fig. 2, för att kunna kommunicera med databasen. Det behövdes metoder som skulle skriva in data, läsa in data och radera data.

```

public void insertMessage(int id, String mess)
{
    Document document = new Document();
    document.append("id", id);
    document.append("meddelande", mess);
    collection = database.getCollection("meddelanden");
    collection.insertOne(document);
}

```

Fig. 2: Metod som skriver in data till MongoDB-servern

När metoderna var skrivna importerades programklassen "Instant" för kunna mäta tiden det

```

Instant first = Instant.now();
insertUser();
Instant second = Instant.now();
System.out.println(Duration.between(first, second).toMillis());

```

tog att utföra kommandona till servern.

Fig 3: Insättning av en användare mäts med hjälp av Instant-variablarna "first" och "second". Programmet skriver ut tidsskillnaden.

Varje programklass körs separat, så varje klass är ett program. När ett program körs så anropas en konstruktor som kör igång alla metoderna. Alla metoder skriver ut sluttiden med funktionen "println". Runt varje metod så finns en "For-loop" som kör om metoden 100 gånger och gör 100 utskrifter.

Själva designen på hur databaserna ska lagras är lite olika, vilket beror på att databaserna lagrar data på olika sätt. MongoDB, Redis och Apache Cassandra lagrar meddelanden och användare i två separata tabeller och en vän-tabell skapades till en användare för sina vänner. Varje användare fick ett namn och ett id. Varje meddelande får ett id relaterat till användaren och ett innehåll till meddelandet. I vän-tabellen så lagras användarens id och dennes namn. Neo4j lagrar inte data i tabeller utan i noder, som binds samman med hjälp av relationer. Redis kan inte lagra flera attribut i en tabell. Den lagrar bara ett attribut med ett index. Eftersom NoSQL-databaser saknar stöd för relationer mellan tabeller, så går det inte att skapa en relation mellan tabellerna i Redis, MongoDB och Apache Cassandra. Det går alltså inte att ha en relation mellan tabellen för användare, meddelande och vänner. Själva programmet får söka efter data som relaterar till en användare.

Användning av simulering

Den första miljön hade 10000 användare 10000 meddelanden i varje databas. Databasen lagrades också från 100 till 10000 vänner. Den andra miljön hade 10 gånger mer data i alla tabeller och det blev högsta antalet, eftersom alla databaser inte klarade av inläsningen av så många datauppgifter eller så tog inskrivningen lång tid. För att få fram ett rimligt värde på tiden det tar att utföra ett kommando så provades varje kommando 100 gånger. Sedan togs det ut ett medelvärde, eftersom tiden varierade (Se bilaga längst bak i dokumentet).

Det första som programmet gör är att skriva in all data som behövs till experimentet. Sedan läser programmet in databasens innehåll och tar tiden för utförandet av kommandot. Eftersom innehållet i tabellen för användare och meddelanden varierar så måste man radera data och skriva in data på nytt, när man ska uppdatera innehållet i tabellerna.

Först läste programmet in ett antal meddelanden från den miljön med 1000000 användare och 100000 meddelanden. Alla meddelanden, som lästes in, var från samma användare. Först lästes det in 100 meddelanden, sedan 10000 och sist 100000 meddelanden. Inläsningen av vänner gjordes först med 100 vänner, 1000 vänner 10000 vänner och sist 100000 vänner. Det sista kommandot var inläsning av användare och inskrivning av användare. En inläsning av en användare, från databasen, skulle vara en liknelse med en inloggning av en användare. Att skriva in en användare, i databasen, skulle vara en liknelse som att registrera en användare.

Sedan gjorde man om samma steg med den andra lagringen med 10000 användare och 10000 meddelande.

3. Resultat av litteraturstudie

Här redovisas resultatet från litteraturstudien.

3.1 Nyckelvärde-databas

Nyckelvärde-databaser används till e-handelssystem [27][3][5][34]. Rollen, som databasen har i ett e-handelssystem är lagring av varor i kundvagnen [3][5][27] och hantering av kundens preferenser och sessioner [5]. Facebook använder nyckelvärde-databas för lagring av användarprofiler och hämtning av produktnamn [3]. Twitter använder en nyckelvärde-databas för att lagra tweets(inlägg) [13]. Nyckelvärde-databaser har används för att förbättra ett söksystem där data är organiserade i noder som binds samma med vektorer [36]. Det beskrivs i artikeln att fler experiment ska utföras. Fler exempel är snabb insättning och mindre hämtning av data [7], hantering av användare eller sessionsprofiler [20], snabb datahämtning [20][31]. Nyckel-värde-databaser passar bra till enkla applikationer med ett programobjekt med ett attribut [33]. Det tas upp ett exempel på ett användningsfall då man skapar en inloggningssida för en användare. Användardata ändras sällan eller ändras vid en känd tidpunkt.

Redis

Jing Han, Haihong E och Guan Le [9] beskriver att Nyckel-värde-databaser passar bäst till system som inte behöver lagra så mycket data. Redis används som ett exempel. Redis rekommenderas till snabba realtidsanalyser, distribution och hantering av innehåll i cacheminne och lagring av cookies och applikationer där det görs mycket skrivning till databasen [13]. Redis rekommenderas inte till transaktionsoperationer som kräver hög säkerhet. Dessutom så kan tillgängligheten av databasen försämrans om det sker nätverksfel. Det gjordes ett experiment för att se hur väl Redis fungerade i ett sensornätverk [21]. Det vill säga ett nätverk i ett inbyggt system. Tanken var att se om det kunde bli ett bättre resultat än en SQL-databas. Resultatet blev ett nätverk med högre skalbarhet och högre prestanda.

DynamoDB

DynamoDB rekommenderas till system som bara utför CRUD-operationer (Create, Read, Update och Delete) [13]. Exempel på sådana system är onlinespel, webbapplikationer och spårning av "klickströmmar". Klickströmmar är en uppräknings av musklick på länkar på en webbsida. DynamoDB rekommenderas inte till system där man använder relationella JOIN-operationer och normalisering av data från olika tabeller. DynamoDB rekommenderas inte heller till system där antalet läsoperationer eller skrivoperationer kan variera drastiskt.

TrippyDB

Det har utvecklats en nyckelvärdedatabashanterare som heter TrippyDB [22]. Den skulle fungera i ett socialt nätverk, där användarens position är betydande. Det gjordes en jämförelse med dokumentdatabasen MongoDB och resultatet blev att TrippyDB ger högre prestanda, men fungerar inte med alla slags applikationer, utan bara till applikationer där användarens position är betydande.

Voldemort

Voldemort är designat för att användas till enklare lagringar så att man inte behöver använda cachinglager [27].

3.2 Dokumentdatabas

Dokumentdatabaser är ett bra val för system där objekt har flera attribut [33]. Ett exempel är en app som registrerar motorfordon och bilarnas attribut som, till exempel, registreringsnummer, ägare och fordonstyp [33]. Karamjit Kaur och Rinkle Rani [15] beskriver de olika NoSQL-databaserna och deras användningsområden. Dokumentdatabaser är bra till webbapplikationer som lagrar semi-strukturerad data och dynamiska förfrågningar. Man Qi [20] beskriver i sin artikel att dokumentdatabaser är bra till realtidsanalyser, loggning av data och som lager i bloggar. Artikeln "NoSQL Evaluation: A use case oriented survey", beskriver att dokumentdatabaser passar bäst till realtidsanalyser, lagring av heterogen data loggning och lagring av små flexibla webbsidor, som bloggar [49].

MongoDB

Jagdev Bhogal och Imran Choksi [3] beskriver att dokumentdatabaser är bra till lagring av dokument som e-postmeddelande och textdokument. I samma artikel så gjordes ett experiment mellan SQL och NoSQL där man gjorde experiment med en bibliotekslagring. Databaserna i experimentet var dokumentdatabasen MongoDB och SQL-databasen APEX. Anledningen till att MongoDB valdes ut är för sin flexibilitet. Resultatet av experimentet visar att det är enklare att använda NoSQL till bibliotekslagring än SQL.

I en jämförelse mellan SQL och NoSQL så valde man att använda SQL-databasen PostgreSQL och dokumentdatabasen MongoDB [4]. De provades i systemet WMS (Web Mapping Service) som skapar kartor med hjälp av punkter, linjer eller multipolygons. Resultatet visar att det är positivt att använda MongoDB för att skapa kartor med punkter, men prestandan försämrades mycket ju fler användare som använde systemet. Deras resultat visar att storleken på lagringen av data inte påverkar prestandan när man använder MongoDB.

Det har utvecklats ett geografiskt system som heter "ConsultaOpinião", vars syfte är att visa geografisk information om byggnader [18]. Detta system använder MongoDB för lagring av information. Det gjordes också ett experiment mellan MongoDB och en SQL-databas där MongoDB visade bäst resultat. Särskilt när man lagrar mycket data.

MongoDB har varit med i ett experiment med lagring av GIS (Geographic Information System), som är ett geografiskt system [17]. Syftet med detta system är att ta emot, lagra, hantera, visa och analysera geografisk data. Det är effektivare att använda MongoDB än en relationsdatabas.

Satelliter kan lagra data för klimatundersökningar [48]. Det gjordes ett experiment för att se om MongoDB är en lämplig databas för ett system för klimatundersökning. Anledningen till att MongoDB är populär är för sin skalbarhet, förmåga att lagra stora datamängder och användning i distribuerade miljöer

Det har gjorts ett experiment med lagring i form av en trädstruktur. Det vill säga att innehållet delas upp i olika noder [7]. Noderna innehåller loggfiler med stora datamängder och noderna är heterogena. När MongoDB testades, till trädstrukturen, så blev resultatet positivt, men grafdatabaser var bättre.

Dokumentdatabasen "MongoDB" har använts i ett projekt som handlar om att utveckla ett socialt nätverk [43]. Deras slutsats är att det finns fördelar och nackdelar med att använda MongoDB till sociala nätverk. Fördelarna är enkel replikering, hög tillgänglighet, snabb på att uppdatera, lätt att hantera och hög skalbarhet. En nackdel är MongoDB inte har stöd för JOIN-operationer bland dokument, vilket leder till att det blir flera förfrågningar och manuellt sammanfatta data. Det leder till minskad flexibilitet. MongoDB saknar också stöd för hantering av flera operationer samtidigt. Facebook är ett socialt nätverk som inte använder en enda databas utan flera databaser.

Det gjordes ett experiment med MongoDB i ett system med sensorer [19] i en virtuell miljö och en fysisk miljö. Resultatet jämfördes med SQL-databasen PostgreSQL och

kolumndatabasen Apache Cassandra. Det användes två labbmiljöer för att prova prestandan på skrivningen. En med flera klienter och en med bara en klient. Databashanterarna provades både i en virtuell och i en fysisk miljö. I labbarna undersöktes prestandan på läsning och skrivning. Först enstaka skrivningar/läsningar och sedan flera skrivningar/läsningar samtidigt. I slutledningen så beskrivs det att Cassandra är det bästa valet för stora kritiska sensorapplikationer. MongoDB är det bästa valet för små eller medelstora applikationer som inte lagrar kritisk data. Speciellt när prestandan på skrivningen är viktig. PostgreSQL är det bästa valet för system som kräver hög flexibilitet på "databasfrågor" eller när prestandan på läsningen är viktig.

Det har gjorts en studie för att se hur MongoDB fungerar när man ska lagra information om bioinformatik [24]. Systemet kräver en databas med hög feltolerans, hög skalbarhet och som kan lagra mycket data. Anledningen till att MongoDB valdes ut var för sin förmåga att lagra mycket data. Det beskrivs ett användningsfall där man jämför människonjuror och RNA-prover från människolever. Resultatet var positivt för MongoDB, men det finns mer att forska i med denna databas.

Medicinsk data innehåller elektroniska inspelningar, inspektioner, rapporter och prognoser m.m, vilket resulterar till att det blir mycket data i lagret [53]. Ett sådant system är MSPM, vilket kräver skalbarhet och en databas som kan lagra mycket data. Databasen som används i systemet är MongoDB.

Dokumentdatabaser används också i telekommunikationssystem [32]. Systemet hanterar uppgifter som arbetskraft, kunddata, samtalshantering och beslut. Rollen som dokumentdatabasen har är att lagra "Active package", som används mycket i systemet. Ett sådant paket innehåller aktiviteter som personalen gjort. Det gjordes ett experiment för att se vilken databas som gjorde bäst ifrån sig i ett system där det blir mycket skrivning [35]. De databashanterare som var med i experimentet var Cassandra, MongoDB, CouchBase server och MS SQL server. Slutresultatet blev att SQL-lösningen fungerade bäst. Bland NoSQL-databaserna så var Cassandra bäst. I slutsatsen så beskrivs det att dokumentdatabaser var mer lämpade till system där det sker mycket läsning.

MongoDB har använts för att lagra analysdata om fel som har inträffat i IT-system [41], för att sedan använda denna information för återställning. Detta system är under utveckling.

Det gjordes ett experiment med databaserna CouchDB, Cassandra, RavenDB, MongoDB, Hypertable, MS SQL Express och Couchbase [12]. Dessa databaser har jämförts med att skapa lagring för data (buckets), där man har tagit tiden för att utföra operationerna läsning, skrivning och radering. RavenDB och CouchDB var sämst i skrivning, läsning och radering. Cassandra var långsam i läsning, men var bra i skrivning och radering. Couchbase och MongoDB är snabbast på läsning, skrivning och radering.

MongoDB och CouchDB och sökmotorn Lucene experimenterades med ett "PACS-system" [26]. PACS står för "Picture Archiving and Communication Systems". Experimentet gick ut på att lagra bilder inom läkarvården. MongoDB fungerade bra i den här lösningen eftersom den är snabb med att ta fram data, men Lucene var bäst i experimentet, eftersom det har

mer stöd för att sökning med fritext. CouchDB fungerade också bra, men då kan man inte fråga fritt i systemet utan måste ha bestämda frågor.

Pragati Prakash Srivastava, Saumya Goyal och Anil Kumar [13] beskriver i sin analys att MongoDB är ett bra val för webbapplikationer som behöver hög skalbarhet, caching-operationer och hantering av semi-strukturerad data. Samtidigt så beskrivs det i deras analys att MongoDB inte bör användas till system som har krav på JOIN-operationer eller främmande nyckelreferenser. Främmande nycklar och join-operationer används för att skapa relationer mellan flera tabeller i databasen. MongoDB bör inte heller användas till applikationer som kräver hög överensstämmelse med ACID. Exempel där MongoDB rekommenderas är till alternativ för relationsdatabaser till weppapplikationer, system som kräver hög skalbarhet och caching-operationer och för hantering av semistrukturerad data [13].

CouchDB

CouchDB är bra till system där data ofta blir modifierad och uppdaterad [13]. CouchDB är också bra till system där användare befinner sig på långa avstånd från lagringen av data, som de måste ha tillgång till. Dock så kan det bli problem med CouchDB i webbutveckling, när det handlar om att kombinera innehåll i flera dokument.

Clusterpoint

ECM står för "Enterprise Content manangement" och är ett system som användes för att ta emot, hantera, lagra och leverera innehåll i ett företag [28]. Exempel på innehåll är email, dokument, bilder och pdf-filer. SQL används som standard men det har gjorts ett försök med dokumentdatabasen Clusterpoint. Resultatet blir en förbättring, men det finns mer att forska i.

Hypertable

Det sociala nätverket LinkedIn använder Hypertable för lagerproblem där funktionell partitionering inte är lämplig [27]. Det är ett nytt system inte har så bra kvalitet.

3.3 Grafdatabaser

Grafdatabaser

Grafdatabaser är bra till system där relationer mellan data har stor betydelse [16][3][49]. Exempel på ett sådant system är socialt nätverk [16][49] eller lokaliseringssystem [3]. Exempel på system som använder grafdatabaser är Twitter [49]. Det sociala nätverket Facebook använder grafdatabaser för att hitta vänner [3][13] eller få fram bästa vägen för datatrafik [13]. I ett experiment så har man skapat en trädstruktur, där data binds ihop med relationer [7]. Slutsatsen är att grafdatabaser är passande för ett sådant system och att

grafdatabaser bör användas till system som kräver snabb hämtning av data och som samtidigt inte behöver snabb insättning.

Grafdatabaser har använts för utveckling av humanoida robotar [8]. Tanken är att robotarna ska få högre intelligens genom att använda en grafdatabas. Anledningen till att grafdatabasen valdes ut är för att den är mycket användbar för att lagra och ta in information om verkligheten, vilket är en fördel för minneshantering av robotarna.

Med hjälp av databashanteraren Neo4j så har man utvecklat databasen "2Path" [48]. Syftet är att databasen ska lagra biologisk data i ett "metaboliskt nätverk". Resultatet blev en förbättring jämfört med en relaterad databas, men det planeras att utveckla systemet ytterligare.

OrientDB

OrientDB är en grafdatabas som har använts till byggprojekt för stålverksanläggningar [44]. Databasen lagrar teknisk data och hanterar olika versioner av teknisk data. Det förekommer även sökningar av teknisk data. Resultatet blev inte så lyckat med en grafdatabas.

Ne04j

Pragati Prakash Srivastava, Saumya Goyal och Anil Kumar [13] beskriver i sin artikel att grafdatabaser är bra till sökning av information och att länka ihop information. Exempel är sökning av relaterade länkar på det sociala nätverket LinkedIn och sökning av vänner på det sociala nätverket Facebook. I sociala nätverk så ger grafdatabasen Ne04j bäst prestanda i ett experiment med olika grafdatabaser, som används i olika sociala nätverk [11]. Experimentet gick ut på att se vilken databas som ger bäst prestanda i läsning av data och insättning av data. Neo04j var bäst, men det finns mer att undersöka.

3.4 Kolumndatabaser

Kolumndatabaser

Jagdev Bhogal och Imran Choksi [3] beskriver att kolumndatabaser är passande för komplexa system. Exempel på företag som använder kolumndatabaser är internetföretaget Google (Bigtable). Kolumndatabaser är också användbara till "Online Analytical Processing" (OLAP) för beräkning av stora datamängder till ett resultat. Analytiska ändamål är också ett exempel på där kolumndatabaser rekommenderas [30] och till lagring av stora datamängder i inbyggda system och lagring av versionsdata [3] eller till andra system med stora datamängder [23][40][49]. Fler exempel där kolumndatabaser rekommenderas är applikationer som kräver hög tolerans, versionshantering, flexibelt dataschema, snabb skrivning och läsning, stöd för tidsstämplar och partiell lagringsåtkomst [34].

HBase

HBase har provats i ett geografiskt bildhanteringssystem där kolumn databasen HBase lagrar geografiska rutor och koordinatinformation [14]. Resultatet av det experimentet var positivt. I ett annat exempel så provades till ett geografiskt system där man skulle lokalisera enheter på olika platser [44].

HBase experimenterades i ett system för analys för att få fram sannolikheten för att det skulle ske en trafikolycka [42]. Det är ett skalbart och flexibelt system och det blir mycket data att lagra. Systemet är fortfarande under utveckling och det finns problem att lösa.

Hbase är en databas som inte passar till applikationer som kräver skanningar av stora datamängder [13]. Den passar inte heller till transaktionssystemet OLTP (Online Transaction Processing) och relationsanalyser [13]. HBase saknar också stöd för "Server Master-failover", vilket innebär att hela datalagringen kollapsar om servern, som har rollen som "Master", kommer ur bruk [10]. HBase är däremot bra till system som kräver olika varianter av läs och skrivoperationer. Facebook använder HBase för hämtning av post och meddelande. Fler exempel är system som hanterar begränsade SQL-frågor och ytterligare ett exempel är till applikationer som kräver konsistens [13]. Hbase är också lämplig till applikationer som behöver varierande realtidsskrivningar eller realtidsläsningar till stora datamängder [27]. HBase har använts till ett streaming-system, som är under utveckling för att förbättra prestandan [6]. Det gjordes ett experiment för att se hur väl HBase klarade av att lagra uppgifter om patienter [45]. Lagringen var på 1 GB. Detta är ett system som fortfarande är under utveckling.

Apache Cassandra

Dileepa Jayathilake, Charith Sooriaarachchi, Thilok Gunawardena, Buddhika Kulasuriya och Thusitha Dayaratne [7] beskriver i sitt experiment med sin trädstruktur med loggfiler, som innehåller stora datamängder, att kolumn databasen Apache Cassandra är inte passande för en sådan struktur.

Apache Cassandra är en databas som används av det sociala nätverket "Facebook" [25][29]. Facebook tillverkade databasen, som skulle hantera hårdvarufel och för att lösa problemet med sökningen i användarens inbox [29].

Telekommunikation är ett annat experiment med Cassandra [25]. Databasen experimenterades med ett telekommunikationssystem där inkommande samtal skulle hanteras. Det blev en jämförelse med sql-databasen PostgreSQL. Resultatet blev att PostgreSQL är bättre eftersom Cassandra saknar stöd för transaktionsgaranti.

RFID står för "Radio Frequency Identification" och är en teknik för kommunikation mellan en tagg och en läsarenhet [47]. Apache Cassandra passar bra för lagring av RFID-data och beskrivs vara effektiv för skrivning [47][39][35], flexibelt, ha stöd för triggoperationer och för att Cassandra är horisontellt skalbar. Det vill säga att man kan öka prestandan genom att lägga in fler maskiner.

MongoDB, Cassandra och Redis var med i ett experiment där man skulle undersöka databasernas feltolerans [49]. Dessa databashanterare är bland de populäraste databaserna och används i kritiska system. Dock inte för att lagra kritisk data. Resultatet visar att säkerheten för integriteten inte alltid är garanterad. Cassandra var sämst med att skydda dataintegriteten och MongoDB var bäst. Redis var nästan lika bra som MongoDB när det handlar om att skydda data. När det gäller dataåterställning så var MongoDB snabb när fel leder till att databasen är avstängd, men inte om, till exempel, operativsystemet blir ifrånslaget. Redis hanterar replikeringen under alla fel, men fungerar inte så bra när mängden ökar. Cassandra hade hela tiden lång kopieringstid.

Cassandra har även stöd för skalbarhet [37][29][39][27] och hög tillgänglighet [27], är användarvänligt och flexibelt. Exempel på system/applikationer där Apache Cassandra är ett bra val är system där det förekommer mer läsning än skrivning till databasen [13][29][35] och program som inte har stort krav på omedelbar konsistens [13]. Fler exempel är webbapplikationer som måste ge dynamiskt schema och innehåll till användare och program som kräver högt underhåll av programkod [13]. Cassandra är också ett bra val om man behöver bearbeta en stor mängd data [23][37][49][27]. Den största lagringen har över 300 Terabyte som delas upp över 400 maskiner [27]. Exempel på sådana system är online-bokning på hotell och flyg. Företag som Netflix, Twitter och Cisco använder Cassandra för hantering av stora datamängder [23].

Andra exempel där Cassandra är lämpligt är till system som är spritt på flera enheter [47][38]. Cassandra bör inte användas till system som har stark konsistens eller som kräver hög konsekvens på transaktions och relationsoperationer eller till dynamiska frågor som involverar aggregatoperationer eller join-operationer [13]. Det gjordes en lagring av genetisk data i textfiler [26]. Ett problem med system var att datamängden hela tiden ökade. Det krävdes också en databas som har stöd för prestanda i skrivning av data till databas. Cassandra var den databas som valdes ut och det blev ett system med ökad prestanda jämfört med en SQL-databas. Ett sjukvårdssystem kan vara spritt på flera kliniker, sjukhus, forskningsanläggningar och regeringar [38]. Det blir hela tiden mer och mer data som behöver lagras. Apache Cassandra används till ett streaming-system inom vården, som ska lagra meddelanden, som går via streaming. I ett experiment med sökmotorn "Lucence" så beskrivs det att Apache Cassandra är ett bra val för denna sökmotor. Även HBase användes till sökmotorn Solr [40] och det blev en jämförelse med Apache Cassandra. HBase var bäst i experimentet. Twitter är ett socialt nätverk som har börjat använda Apache Cassandra för lagring av tweets och frågor för analys av händelser och historik på det sociala nätverket [39].

Cassandra användes i ett experiment för att undersöka dess egenskaper [37]. Man experimenterade inläsning, skalbarhet och uppdatering. Resultatet visar inläsning inte är den starkaste sidan hos Cassandra. När det gäller skalbarhet så gäller det att se till att ha rätt antal trådar för skalbarheten. Det finns fler artiklar som visar att inläsning inte är den starkaste sidan hos Cassandra [37].

4 Resultatet av den praktiska studien

Här beskrivs den praktiska studien över databashanterare som används i sociala nätverk. Följande hanterare har provats. Redis (Nyckelvärde-databas), MongoDB (Dokumentdatabas), Apache Cassandra (Kolumn-databas) och Neo4j (Graf-databas), eftersom dessa databashanterare används till sociala nätverk. Tabellerna 3.1, 3.2 och 3.3 sammanfattar resultatet av simuleringarna. Tabellernas innehåll är hämtat från Bilaga 1. Värdena i tabellerna beskriver medelvärdet för tiden, som är angiven i millisekunder. På grund av att Redis inte kunde läsa in 100 000 användare eller 100 000 vänner så blev det ingen tid. Experimenten relaterar till sociala nätverk för att i sociala nätverk sker inläsning av vänner användare och meddelanden. Diagrammen 3.1, 3.2 och 3.3 förtydligar information i tabellerna. När man loggar in i det sociala nätverket så måste dessa datauppgifter läsas in till användaren. Det bör kunna göras på så kort tid som möjligt så att användaren kan komma igång med sina aktiviteter på det sociala nätverket. Den praktiska studien ska visa vilken databashanterare som gör de olika inläsningarna på kortast tid.

Kommandon	Cassandra	MongoDB	Neo4j	Redis
inläsning av 100 vänner	4,0	0,3	101,0	0,1
inläsning av 1000 vänner	8,3	1,6	96,8	0,4
Inläsning av 10000 vänner	20,2	9,1	112,0	3,5
Inläsning av 1000000 vänner	31,5	10,0	73,30	–

Tabell 3.1 Inläsning av vänner/andra användare

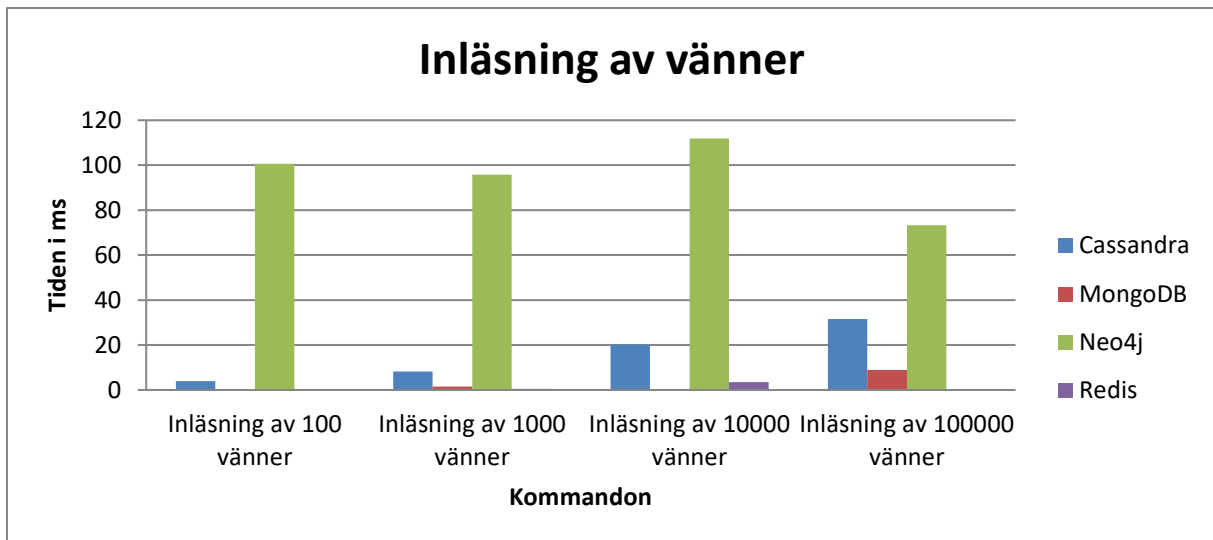


Diagram 3.1 Inläsning av vänner/andra användare

Kommandon	Cassandra	MongoDB	Neo4j	Redis
Hämta en användare	2,59	45,78	0,1	0,63
Registrera användare	3,43	0,18	0,03	0,03

Tabell 3.2 Inloggning/registrering av användare

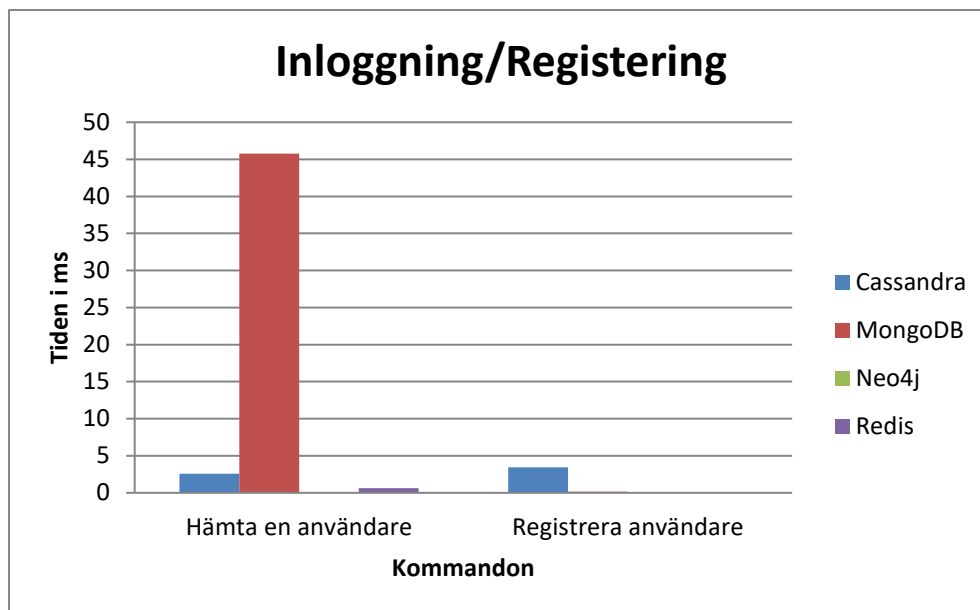


Diagram 3.2 Inloggning/registering av användare

Kommandon	Cassandra	MongoDB	Neo4j	Redis
Hämta 100/100000	4,38	42,33	0,25	0,33
Hämta 1000/ 100000	192,57	43,21	0,66	0,35
Hämta 10000/100000	408,79	48,08	2,81	3,72
Hämta 100000/100000	37,42	140,98	28,73	–
Hämta 100/10000	18,91	7,58	0,25	7,58
Hämta 1000/ 10000	21,92	16,43	0,77	16,43
Hämta 10000/10000	30,92	7,01	0,12	7,01

Tabell 3.3 Inläsning av meddelanden

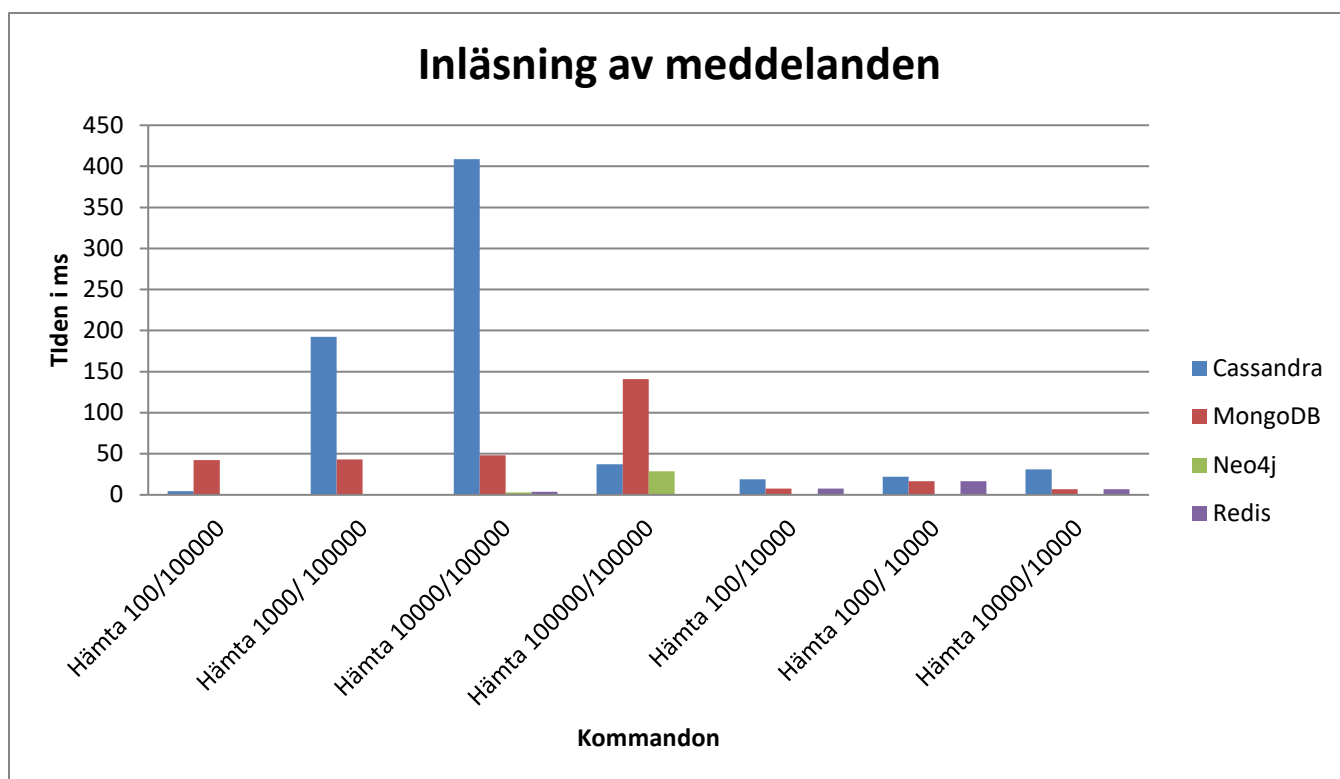


Diagram 3.3 Inläsning av meddelanden

4. Analys och diskussion

4.1 Analys av den praktiska studien

4.1.1 Inloggning och registrering av användare

Tabellen 3.2 sammanfattar resultaten. Redis var den databashanterare som utförde ordern snabbast av alla NoSQL-databaserna i alla tester utom i inläsning och registrering av en användare (Tabell 3.1 - Tabell 3.3). MongoDB var snabbast på registreringen, men Neo4j var snabbast på inläsning av en användare. MongoDB var sämst när programmet skulle hämta en användare.

4.1.2 Inläsning av vänner

Neo4j var sämst i inläsning av vänner i båda miljöerna. Redis var bäst i alla tester, men klarade inte av den inläsningen av högsta antalet vänner. Cassandra var bättre än Neo4j och Redis, men sämre än MongoDB. MongoDB klarade alltså testet bäst av alla databaser.

4.1.3 Inläsning av meddelanden

När meddelanden skulle läsas in så var Neo4j den bästa databasen i både den miljön med 10000 meddelande och 100000 meddelanden. Redis var snabb i den större miljön, men långsammare i den mindre och klarar dessutom inte att läsa in 100000 objekt. Cassandra tog längst tid på sig i inläsning av meddelanden.

4.2 Analys av litteraturstudien

Resultatet beskriver vad de olika kategorierna, inom NoSQL-databaserna, bör användas till, men det beskriver också vad olika databashanterare bör användas till. Ett exempel är Redis och DynamoDB, som är två databashanterare inom kategorin nyckelvärde-databaser. Resultatet beskriver inte någon gemensam rekommendation för dessa databashanterare, vilket innebär att det även beror på själva databashanteraren. Det visar att det inte räcker att bara välja en databashanterare efter kategorin.

Alla NoSQL-databaser är populära inom olika områden, men själva uppgiften, som databaserna har i de olika systemen, beskrivs inte i alla artiklar. Många artiklar beskriver bara till vilket system databasen har använts till.

Artikeln som beskriver ett experiment med databaserna i en virtuell miljö och en fysisk miljö visar att prestandan inte bara beror på val av databashanterare. Utan också på hur man har byggt upp sin miljö [19]. Det som också påverkar är hur många order och klienter som databasen hanterar.

4.3 Diskussion

Frågan var vilken databas som passade bäst till ett socialt nätverk gällande användaruppgifter, meddelande och vänner, vilka uppgifter olika NoSQL-databas bör ha i ett socialt nätverk och vilka egenskaper som kan vara relevanta i ett socialt nätverk. Resultatet från litteraturstudien innehåller information om vilka NoSQL-databaser som används i ett socialt nätverk och vilken roll och vilka egenskaper databaserna har. Alla kategorier kan användas i ett socialt nätverk, men de har lite olika roller. Det finns en konflikt mellan två artiklar. En artikel beskriver att nyckelvärde-databaser används av Twitter för att lagra tweets [13]. En annan artikel beskriver att Apache Cassandra används för att lagra tweets på Twitter [39]. Då är frågan vilken artikel som har rätt. Den artikel som handlar om Apache Cassandra har fått fram sitt resultat från ett experiment med tweets. Den andra artikeln redovisar inget experiment. Artiklarna är också publicerade nästan samtidigt. Men en artikel som beskriver ett experiment är mer trovärdig, eftersom dess innehåll har stöd. En annan möjlighet är att Twitter har utvecklat sitt system och bytt databas som lagrar tweets.

Den praktiska studien var fokuserad på användaruppgifter, meddelanden och vänner i ett socialt nätverk. Detta är det grundläggande kraven i ett socialt nätverk. Det fanns möjlighet att göra experiment med inläsning av meddelanden och inläsning av vänner. Den praktiska studien blir ett tillägg i undersökningen över vilka roller olika NoSQL-databas passar bäst till i ett socialt nätverk. Den ifrågasätter också artiklarna gällande prestanda för inläsning av objekt, som vänner [13][3] och meddelande [3][28].

Redis bör inte användas till system som lagrar mycket användaruppgifter. Som stöd för detta så beskriver Jing Han, Haihong E och Guan Le [9] att Nyckel-värde-databaser passar bäst till

system som inte behöver lagra så mycket data. Dessutom så är denna databas inte så praktiskt, eftersom man bara kan lagra ett attribut till varje objekt. Ändå så används den för hantering av användare och sessionsprofiler [20] och lagring av meddelande [13]. Fördelen är att Redis hela tiden hade lågt värde i alla experiment. De övriga databasernas tid kunde variera kraftigt i de olika experimenten, vilket försvårar vilken databas som man bör finna lämplig till lagring av användaruppgifter, meddelande och vänner.

Neo4j var visserligen bäst på inläsning av meddelanden (Tabell 3.3), bäst efter Redis i registrering och inläsning av användare, men sämst på inläsning av vänner (Tabell 3.1). Trots det dåliga resultatet av sökning av vänner så används Neo4j av Facebook för just sökning av vänner [13][3][13] eller för att få fram bästa vägen för datatrafik [13]. Det som skiljer experimentet mellan inläsning av vänner och inläsning av meddelanden är att programmet läser in allt innehåll från tabellen som lagrar vänner och läser in en del av allt innehåll i tabellen som lagrar meddelanden för flera användare. Neo4j rekommenderas också till system där objekten har relationer mellan varandra [16][3][49].

Den databas som klarar sig bäst i inläsning av vänner, i den praktiska studien, är MongoDB. Enligt en artikel, som redovisar ett experiment med MongoDB, Apache Cassandra och Redis, så var MongoDB den databas som lyckades bäst när det handlar om att skydda dataintegriteten [49]. I den praktiska studien så var MongoDB näst sämst i inläsning av meddelanden, men enligt Jagdev Bhogal och Imran Choksi [3], så används dokumentdatabaser till lagring av e-postmeddelanden [3][28] och textdokument.

Apache Cassandra är enligt några artiklar rekommenderat när det sker mer läsning än skrivning till databasen [13][29][35]. I den praktiska studien så tog det lång tid att skriva in data till databasen. Längre tid än de andra. Detta motsäger några artiklar [12][37], som beskriver att Cassandra är bättre på skrivning än läsning. Cassandra är inte heller lika bra som Redis och MongoDB när det gäller att skydda data [49], vilket är viktigt när det kommer till lagring av användaruppgifter, meddelande och vänner.

4.4 Metoddiskussion

Den praktiska studien har visat tydligt att det finns skillnader mellan de olika NoSQL-databaserna. Hela den praktiska studien genomfördes på en PC-maskin. Man brukar använda servrar för att kunna lagra datauppgifter till ett socialt nätverk, så det vore mer intressant att göra samma experiment med databaserna i en mer avancerad datamiljö. Om man skulle vilja nå det sociala nätverket över Internet så vore det även intressant att kommunicera med databaserna över internet. Facebook och Twitter kommunicerar man med via Internet. Det högsta antalet objekt i den miljö som har använts till den här studien var 100000. Ska man ha ett socialt nätverk som är tillgängligt för vem som helst så bör man egentligen testa med ett större antal. I den praktiska studien så gjordes anrop från bara en klient. En artikel visar datamiljön har betydelse för databasernas prestanda och hur många order och klienter som databasen hanterar [19]. Det skulle vara intressant att experimentera med flera klienter som har kontakt med databasen samtidigt.

5. Svaret på forskningsfrågor

Vilken databas passar till ett socialt nätverk när det handlar om lagring av användaruppgifter, meddelande och vänner? Neo4j var bäst i alla experiment utom i inläsning av vänner. I det experimentet var MongoDB bäst.

Vilka roller har NoSQL-databaser i ett socialt nätverk? NoSQL-databaser har roller som lagring av användarprofiler (Redis [9]), lager för hantering av fel i hårdvara (Cassandra [29]), lagring av inlägg (Cassandra, Redis), lagring av meddelande (MongoDB [3][28] och Redis [13]), sökning av vänner (Neo4j [13][3][13]) och hitta bästa vägen för datatrafik (Neo4j [13]).

Vilka egenskaper har de olika databaserna, som är relevanta för ett socialt nätverk? I ett socialt nätverk så brukar det användas av många användare och det kan ske mycket inläsning och skrivning i det sociala nätverket. Fler saker som är relevanta är dataintegritet och relationer mellan datauppgifter. Några artiklar beskriver att Apache Cassandra är bra till system där det sker mycket skrivning [37][29][35]. Andra artiklar beskriver att Apache Cassandra är bättre på inläsning [13][23][27]. Cassandra rekommenderas också till system som lagrar mycket data [23][37][49][27]. MongoDB har dock bättre skydd mot dataintegritet än vad Cassandra har [49]. Grafdatabaser är rekommenderat för system där det finns relationer mellan dataobjekt [16][49][3][13][52][7], vilket är relevant, eftersom man knyter kontakt med varandra i ett socialt nätverk. Det som också är relevant är hur många läsningar/skrivningar, som görs av varje användare, hur många användare som finns i systemet och om man har en virtuell och fysisk miljö. Databaserna MongoDB och Cassandra testades i ett sådant experiment [19] och båda databasernas prestanda varierade mycket beroende på miljö, antal användare och antal skrivningar och läsningar.

6. Slutsatser och vidare forskning

Den här studien har svarat på vilken databas som är rekommenderad i ett socialt nätverk, vad en NoSQL-databas har för roll i nätverket och vilka egenskaper olika databashanterare har, som är relevanta för ett socialt nätverk. Notera att detta är en initial studie. För att validera våra initiala resultat i denna studie så behövs ett mer avancerat system.

7. Källor

- [1] Margaret Rouse, "NoSQL (Not Only SQL database)", searchdatamanagement, mars 2017[Online] Tillgängligt: <http://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL>
- [2] Mwtoeks, "SQL", Wikipedia, 22 januari 2018 [Online] Tillgängligt: <https://en.wikipedia.org/wiki/SQL>
- [3] Jagdev Bhogal och Imran Choksi, Handling Big Data Using NoSQL, 24-27 mars 2015, Gwangju, Södra Korea, Advanced Information Networking and Applications Workshops, IEEE, 2015. Tillgängligt: IEEE, <http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=709620>
- [4] Stephan Schmid, Eszter Galicz, Wolfgang Reinhardt, WMS Performance of Selected SQL and NoSQL Databases i *International Conference on Military Technologies*, 19-21 maj 2015, Brno, Tjeckien, (ICMT) 2015, IEEE, 13 juli 2015. Tillgängligt: IEEE, <http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=7153736>
- [5] Maria Teresa González-Aparicio och Muhammed Younas och Javier Tuya, A New Model for Testing CRUD Operations in a NoSQL Database i *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 23-25 mars 2016, Crans-Montana Schweiz, IEEE, 23 maj 2016. Tillgängligt: IEEE <http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=7474073>
- [6] Chai-Ping Tsai, Kou-Yu Yang, Jui-Yu Hsieh, Hung-Chang Hsiao, Ching-Hsien Hsu, Publish /Subscribe in NoSQL i , *2014 IEEE 17th International Conference on Computational Science and Engineering*, 19-21 december 2014, Chengdu, Kina, IEEE 29 januari 2015. Tillgängligt IEEE <http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=7023737>
- [7] Dileepa Jayathilake Charith Sooriaarachchi och Thilok Gunawardena, A Study into the Capabilities of NoSQL Databases in Handling a Highly Heterogeneous Tree i *2012 IEEE 6th International Conference on Information and Automation for Sustainability*, 27-29 september 2012, Beijing Kina, IEEE 28 januari 2013. Tillgängligt IEEE <http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=6419890>
- [8] S. Vijaykumar och S.G Saravanakumar, Implementation of NOSQL for robotics i *2010 International Conference on Emerging Trends in Robotics and Communication Technologies (INTERACT)* , 3-5 december 2010, Chennai, Indien, , IEEE 31 januari 2011. Tillgängligt IEEE <http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=5706225>

- [9] Jing Han, Haihong E, Guan Le, Survey on NoSQL Database i 2011 6th International Conference on Pervasive Computing and Applications, 26-28 Oktober 2011, Port Elisabeth, Södra Africa, IEEE 19 december 2011. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=6106531>
- [10] Wumuti Naheman, Jianxin Wei, Review of NoSQL Databases and Performance Testing on HBase i Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), 20-22 December 2013, Shengyang, Kina, IEEE 28 augusti 2014. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=6885425>
- [11] Anita Brigit Mathew och S. D Madhu Kumar, Analysis of Data Management and Query Handling in Social Networks using NoSQL Databases i International Conference on Advances in Computing, Communications and Informatics (ICACCI), 10-13 augusti 2015, Kochi Indien, 2015, IEEE 28 september 2015. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=7275708>
- [12] Yishan Li och Sathiamoorthy Manoharan , A performance comparison of SQL and NoSQL databases i 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), 27-29 augusti, 2013, Victoria, BC, Canada, IEEE 10 oktober 2013. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=6625441>
- [13] Pragati Prakash Srivastava, Saumya Goyal och Anil Kumar, Analysis of various NoSQL database i 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 8-10 oktober 2015, Noida, Indien, IEEE 14 januari 2016. Tillgängligt IEEE <http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=7380523>
- [14] Zhifeng Xiao och Yimin Liu, Remote Sensing Image Database Based on NOSQL Database i 2011 19th International Conference on Geoinformatics, 24-26 juni 2011, Shanghai, Kina , IEEE 12 augusti 2011, Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=5980724>
- [15] Karamjit Kaur och Rinkle Rani, Modeling and Querying Data in NoSQL Databases i 2013 IEEE International Conference on Big Data, 6-9 oktober 2013, Silicon Valley, CA, USA, IEEE 23 december 2013. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=6691765>
- [16] Arnaud Castellort och Anne Laurent, Representing History in Graph-Oriented NoSQL DataBases: A Versioning System i International Conference on Digital Information Management (ICDIM 2013), 10-12 september 2013, Islamabad, Pakistan, Eighth, IEEE 6 januari 2014. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=6694022>
- [17] Xiaomin Zhang, Wei Song och Liming Lu, An Implementation Approach to Store GIS Spatial Data on NoSQL Database i 2014 22nd International Conference on Geoinformatics 25-27 juni, 2014, Kaohsiung, Taiwan, IEEE 10 november 2014. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=6950846>

- [18] Daniel Cosme Mendonça Maia, Breno D. C. Camargos och Maristela Holanda, Voluntary Geographic Information Systems with Document-based NoSQL Databases i *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, 15-18 juni 2016, Las Palmas Spanien, IEEE 28 juli 2016. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=7521439>
- [19] Jan Sipke van der Veen, Bram van der Waaij och Robert J. Meijer, Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual i *2012 IEEE Fifth International Conference on Cloud Computing 24-29 juni 2012*, Honolulu HI, USA, IEEE 2 augusti 2012. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=6253535>
- [20] Man Qi, Digital Forensics and NoSQL Databases i *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 19-21 augusti, 2014, Xiamen, Kina, IEEE 11 december 2014. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=6980927>
- [21] T. A. M. C. Thantriwatte och C. I. Keppetiyagama, NoSQL Query Processing System for Wireless Ad-hoc and Sensor Networks i *2011 International Conference on Advances in ICT for Emerging Regions (ICTer)*, 1-2 september 2011, Colombo, Sri Lanka, IEEE 10 november 2011. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=6075030&tag=1>
- [22] Iskandar Setiadi och Achmad Imam Kistijantoro, TippyDB: Geographically-aware distributed NoSQL Key-Value store i *2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, 19-22 augusti 2015, Chonburi, Thailand, IEEE 23 november 2015. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=7335354>
- [23] Rodrigo Aniceto, Rene Xavier, Maristela Holanda, Maria Emilia Walter och Sérgio Lifschitz, Genomic Data Persistency on a NoSQL Database System i *2014 IEEE, International Conference on Bioinformatics and Biomedicine (BIBM)*, 2-5 november 2014, Belfast, Storbritannien, IEEE 15 januari 2015. Tillgängligt IEEE
<http://ieeexplore.ieee.org.proxy.mah.se/>
- [24] Valeria Guimaraes, Fernanda Hondo, Rodrigo Almeida, Harley Vera, Maristela Holanda, Aleteia Araujo, Maria Emilia Walter och Sergio Lifschitz, A Study of Genomic Data Provenance in NoSQL Document-Oriented Database Systems i *2015 IEEE, International Conference on Bioinformatics and Biomedicine (BIBM)*, 9-12 november 2015, Washington, DC, USA, IEEE 17 december 2015, Tillgängligt:
<http://ieeexplore.ieee.org.proxy.mah.se/document/7359902/>
- [25] Francisco Cruz, Pedro Gomes, Rui Oliveira och Jose Pereira, Assessing NoSQL Databases for Telecom Applications i *2011 IEEE 13th Conference on Commerce and Enterprise Computing*, 5-7 Sept. 2011, Luxemburg, Luxemburg, IEEE 17 oktober 2011, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/6046987>
- [26] Luís A. Bastião Silva, Louis Beroud, Carlos Costa och José Luis Oliveira, Medical imaging archiving: a comparison between several NoSQL solutions i *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, 1-4 Juni 2014, Valencia, Spanien, IEEE 28 Juli, 2014, Tillgängligt:
<http://ieeexplore.ieee.org.proxy.mah.se/document/6864305/>

- [27] Deka Ganesh Chandra, Ravi Prakash och Swati Lamdharia A Study on Cloud Database i *2012 Fourth International Conference on Computational Intelligence and Communication Networks*, 3-5 Nov. 2012, Mathura, Indien, IEEE 6 december 2012, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/6375167/>
- [28] Juris Rats och Gints Ernestsons, Using of cloud computing, clustering and document-oriented database for enterprise content management, 23-25 Sept. 2013, Lodz, Poland, *2013 Second International Conference on Informatics & Applications (ICIA)*, IEEE 31 October 2013, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/6650232/>
- [29] Avinash Lakshman och Prashant Malik, Cassandra - A Decentralized Structured Storage System, *ACM SIGOPS Operating Systems Review*: p.35-40, 14 april 2010
- [30] Moon Soo Cha, So Yeon Kim, Jae Hee Ha, Min-June Lee, Young-June Choi och Kyung-Ah Sohn, CBDIR: Fast and Effective Content Based Document Information Retrieval System i *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, 28 June-1 July 2015, Las Vegas, NV, USA, IEEE 27 Juli 2015, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/7166594/>
- [31] Venkat N. Gudivada, Dhana Rao och Vijay V. Raghavan, NoSQL Systems for Big Data Management i *2014 IEEE World Congress on Services*, 27 Juni-2 Juli 2014, Anchorage, AK, USA, IEEE 22 sep. 2014, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/6903264/>
- [32] Tuncay Yigit, Mehmet Akif Cakar och Asim Sinan Yuksel, The Experience of NoSQL Database in Telecommunication Enterprise i *2013 7th International Conference on Application of Information and Communication Technologies*, 23-25 Oct. 2013, Baku, Azerbajdzjan, IEEE 27 jan. 2014, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/6722702/>
- [33] Philip Moore, Tarik Qassem och Fatos Xhafa, 'NoSQL' and Electronic Patient Record Systems: Opportunities and Challenges i *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 8-10 Nov. 2014, Guangdong, Kina, IEEE 29 jan. 2015, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/7024600/>
- [34] Jiayong Tang, Fei Yang, Yong Zhang och Chunxiao Xing, Using Paxos to Build a Lightweight, Highly Available Key-value Data Store i *Yangzhou, Kina, 2013 10th Web Information System and Application Conference*, 0-15 Nov. 2013, IEEE 27 Mars 2014, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/6778654/>
- [35] João Ricardo Lourenço, Veronika Abramova, Bruno Cabral, Jorge Bernardino, Paulo Carreiro och Marco Vieira, NoSQL in practice: a write-heavy enterprise application i *2015 IEEE International Congress on Big Data*, 27 Juni-2 Juli 2015, New York, NY, USA, IEEE 20 August 2015, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/7207274/>

- [36] Yuezhao Zhang, Li Zha, Jia Liu, A Large-scale Online Search System of High-Dimensional Vectors Based on Key-Value Store i *2012 Eighth International Conference on Semantics, Knowledge and Grids*, 22-24 Oct. 2012, Beijing, Kina, IEEE 24 Dec. 2012, Tillgängligt: <http://ieeexplore.ieee.org/document/6391840/>
- [37] Veronika Abramova, Jorge Bernardino och Pedro Furtado, Testing Cloud Benchmark Scalability with Cassandra i *2014 IEEE World Congress on Services*, 27 Juni-2 Juli 2014, Anchorage, AK, USA , IEEE 22 September 2014, Tillgängligt: <http://ieeexplore.ieee.org/document/6391840/http://ieeexplore.ieee.org.proxy.mah.se/document/6903301/>
- [38] Van-Dai Ta, Chuan-Ming Liu och Goodwill Wandile Nkabinde, Big data stream computing in healthcare real-time analytics i *2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 5-7 Juli 2016, Chengdu, China, IEEE 4 aug. 2016, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/7529531/>
- [39] R. M. Simmonds, P. Watson; J. Halliday, P. Missier, A Platform for Analysing Stream and Historic Data with Efficient and Scalable Design Patterns i *2014 IEEE World Congress on Services*, 27 June-2 July 2014, Anchorage, AK, USA, IEEE 22 sep. 2014, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/6903262/>
- [40] Bao Rong Chang, Hsiu-Fen Tsai, Yo-Ai Wang och Chia-Yen Chen, Realization of secondary indexing to NoSQL database with intelligent adaptation i *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 20-22 Nov. 2015, Tainan, Taiwan, IEEE 15 feb. 2016, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/7407078/>
- [41] Rong Liu, Qicheng Li; Feng Li, Lijun Mei och Juhnyoung Lee, Big Data architecture for IT incident management i *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*, 8-10 Oct. 2014, Qingdao, Kina, IEEE 20 nov. 2014, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/6960762/>
- [42] Duckwon Chung, Xuhua Rui, Dugki Min och Hwasoo Yeo, Road Traffic Big Data Collision Analysis Processing Framework i *2013 7th International Conference on Application of Information and Communication Technologies*, 23-25 Oct. 2013, Baku, Azerbajdzjan, IEEE 27 jan. 2014, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/6722733/>
- [43] Sumitkumar Kanoje, Varsha Powar och Debajyoti Mukhopadhyay, Using MongoDB for Social Networking Website deciphering the pros andcons i *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 19-20 Mars 2015, Coimbatore, Indien , IEEE 13 aug. 2015, Tillgängligt: <http://ieeexplore.ieee.org.proxy.mah.se/document/7192924/>
- [44] Victor Zakhary, Divyakant Agrawal och Amr El Abbadi, Caching at the Web Scale: [Tutorial] i *Proceedings of the 26th International Conference on World Wide Web Companion*, 3-7 april 2017, Perth Australien, ACM Digital Library, 3 April 2017, Tillgängligt: <https://dl.acm.org/citation.cfm?id=3051098>
- [45] J.W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, Thousand Oaks, USA, SAGE Publications Inc, 2003
- [46] Businessdictionary.com, 'quantitative research'. [Online]. Tillgängligt: <http://www.businessdictionary.com/definition/quantitative-research.html>

- [47] Businessdictionary.com, 'qualitative data'. [Online] . Tillgängligt:
<http://www.businessdictionary.com/definition/qualitative-data.html>
- [48] Waldeyr Silva, Danilo Vilart, Daniel Souzat, Maria Emflia Waltert Marcelo Brfgido och Maristela Holandat, 2Path: a terpenoid metabolic network modeled as graph database i 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 15-18 December 2016, Shenzhen, Kina, IEEE, 19 Januari 2017, Tillgängligt:
<http://ieeexplore.ieee.org.proxy.mah.se/stamp/stamp.jsp?arnumber=7822709>
- [49] Robin Hecht och Stefan Jablonski, NoSQL evaluation: A use case Oriented Survey i 2011 International Conference on Cloud and Service Computing, 12-14 Dec. 2011, Hong Kong, Kina , 26 January 2012 IEEE Tillgängligt:
<http://ieeexplore.ieee.org.proxy.mah.se/document/6138544/>

Bilaga 1

MongoDB

Inläsning av meddelande(100000 stycken)

100/100000	1000/100000	10000/100000	100000/100000
45	44	73	260
42	40	54	203
39	39	53	205
39	38	47	254
46	37	50	156
46	39	47	132
43	40	52	155
42	39	48	427
44	39	48	154
43	43	48	124
44	39	50	129
43	39	48	128
44	37	48	137
43	43	46	140
44	49	45	124
45	50	48	125
45	45	46	132
43	43	48	118
43	41	46	123
44	41	49	131

43	39	48	125
41	46	46	134
40	82	47	141
43	82	47	129
45	81	48	136
43	92	48	138
41	98	48	165
43	89	52	191
42	86	45	141
38	42	46	150
39	42	46	117
41	43	49	136
43	42	51	122
43	39	48	152
43	37	47	134
42	36	49	138
43	37	49	121
43	37	47	146
41	37	46	127
42	36	48	141
44	38	46	131
42	39	46	125
45	39	47	128
45	36	48	119
41	38	45	129
42	37	46	127
41	37	46	143
43	37	48	130
43	39	48	118
40	38	45	126
39	38	46	127
39	37	47	146
42	37	48	124
42	39	49	129
40	39	51	168
40	39	47	130
40	39	46	151
40	40	47	120
42	40	47	184
43	37	49	242
43	40	47	205
42	41	45	128
42	43	46	130
41	38	52	120
42	37	46	123

43	39	49	120
41	45	45	121
43	42	50	141
44	40	49	118
43	42	50	124
43	41	49	139
43	40	48	120
43	38	50	121
43	38	47	136
43	38	47	133
43	37	46	142
44	37	47	143
42	37	47	126
41	36	49	127
42	37	48	138
42	37	45	123
42	38	46	123
43	37	49	118
42	39	47	119
42	36	49	124
42	47	49	126
42	41	51	143
42	38	46	130
42	38	45	118
42	39	52	128
42	42	48	117
42	49	48	115
41	42	51	143
43	42	46	119
44	38	46	124
42	38	52	118
43	43	48	127
42	51	48	122
42	45	46	129
41	44	48	139

Inläsning av meddelande (10000 stycken)

100/10000	1000/10000	10000/10000
17	41	22

11	25	10
11	19	11
11	18	7
11	17	8
15	16	6
11	14	10
10	15	6
9	18	7
11	15	9
11	14	11
10	14	7
11	13	7
11	14	7
10	14	6
10	20	6
10	14	5
10	20	7
11	20	8
10	14	6
10	20	5
10	14	6
11	21	5
11	20	6
11	14	6
9	20	10
10	15	11
9	15	10
10	14	9
9	14	6
9	14	10
8	15	7
9	16	10
10	16	7
9	16	5
8	16	5
8	18	6
8	23	6
10	18	6
8	21	6
7	16	10
8	15	6
6	16	7
10	21	6
7	23	5
9	18	6

5	19	6
5	18	7
5	22	5
5	23	5
6	22	6
8	15	6
5	17	6
5	19	6
9	18	6
6	23	6
5	17	10
6	16	10
6	20	6
10	15	6
5	13	6
9	14	6
5	13	8
5	13	6
5	13	6
5	14	10
5	13	10
6	15	7
5	13	9
5	16	10
5	21	10
5	16	7
5	23	5
4	15	6
5	16	8
6	15	6
5	16	6
10	15	6
9	15	6
5	16	5
5	16	5
5	16	6
5	13	5
5	13	5
6	13	5
5	13	5
5	13	9
5	13	5
5	13	6
6	13	6
5	13	7

5	13	6
6	13	6
5	13	6
5	13	6
5	14	7
5	13	6
5	13	6
5	14	6
5	15	6

Hämta vänner

100 vänner	1000 vänner	10000 vänner	100000 vänner
2	1	12	12
1	2	9	9
0	1	9	9
0	2	10	9
1	1	9	8
0	2	9	10
0	2	9	8
0	1	9	9
1	2	9	10
0	1	8	10
0	2	12	9
1	1	8	9
0	2	10	9
0	2	10	9
1	1	9	9
0	1	10	8
0	2	9	9
1	2	10	9
0	1	9	9
0	2	8	8
0	1	8	8
1	2	9	9
0	1	9	9
0	2	9	8
0	2	9	9

1	2	10	8
0	1	9	9
0	2	9	9
1	2	9	8
0	1	8	9
0	2	9	9
1	1	11	9
0	2	8	9
0	1	8	9
1	2	9	9
0	1	9	9
0	2	9	8
1	1	9	9
0	2	8	9
0	2	9	11
1	1	9	9
0	2	8	11
0	1	9	10
0	1	10	9
1	2	9	10
0	2	10	9
0	2	9	8
1	2	9	9
0	2	8	9
1	1	8	9
0	2	8	9
1	1	9	9
0	1	10	8
1	1	9	9
0	2	9	8
0	1	10	10
1	2	9	8
0	2	10	9
0	1	9	9
1	2	9	9
0	2	10	9
0	1	8	9
0	1	12	9
0	2	9	9
1	1	9	9
0	1	10	8
0	2	9	9
0	2	9	8
1	1	9	9
0	2	10	8

0	1	8	8
0	1	9	9
1	1	8	9
0	2	9	9
0	1	8	9
1	2	9	8
0	2	12	10
0	1	8	9
1	2	9	9
0	1	9	10
0	2	9	8
0	2	8	9
0	1	9	8
0	2	8	9
1	1	9	9
0	2	9	9
0	1	9	9
0	2	10	8
0	1	9	9
0	2	9	10
0	1	9	8
0	2	9	10
1	2	8	9
0	2	9	8
0	1	9	10
1	2	9	8
0	1	9	9
0	1	10	9
0	2	9	10
0	2	8	9

Apache Cassandra

Inläsning av meddelande(100000 stycken)

100/100000	1000/100000	10000/100000	100000/100000
4	258	360	47
5	192	329	46
4	176	334	40

6	179	338	41
5	173	363	47
4	186	485	38
4	188	402	39
5	178	341	33
4	186	330	31
4	189	344	31
4	184	360	29
4	169	349	32
6	183	342	31
5	174	431	44
6	172	438	48
5	171	418	39
5	168	397	34
5	170	442	33
5	171	393	34
5	174	432	31
5	168	449	34
5	172	421	32
5	169	359	31
5	173	342	31
4	182	386	34
5	194	339	48
5	174	344	32
5	192	343	32
6	191	387	32
4	178	419	32
4	189	417	37
4	253	492	41
11	189	479	45
5	183	421	35
5	186	360	44
5	181	366	88
4	182	426	32
4	199	349	42
4	184	428	37
4	179	403	32
4	191	454	43
5	187	443	39
4	225	413	36
5	202	448	52
4	190	457	41
4	179	496	38
5	187	515	34
5	185	482	31

5	194	428	32
6	213	432	46
5	186	457	47
4	198	391	35
4	211	345	33
4	221	359	33
4	329	353	45
4	340	353	32
4	215	369	46
5	179	385	31
5	191	426	46
4	225	397	37
4	198	376	34
4	191	344	45
4	188	399	47
5	181	400	64
5	196	411	32
5	200	473	32
6	192	495	47
5	193	497	46
5	190	468	35
6	189	434	39
2	187	415	33
3	191	493	33
4	192	468	32
3	188	421	30
2	194	373	34
3	192	363	34
4	193	506	36
2	181	392	34
2	186	367	31
2	180	413	44
3	197	401	47
3	194	432	32
3	213	444	43
3	193	449	32
3	202	486	30
5	187	354	31
3	183	342	35
3	189	375	43
5	212	382	31
4	192	401	31
2	189	420	32
4	195	450	30
4	183	424	32

4	190	387	32
4	180	450	31
5	182	387	47
5	192	437	31
5	190	490	32
5	180	474	31
5	205	425	31

Inläsning av meddelande (10000 stycken)

100/10000	1000/10000	10000/10000
46	50	69
24	30	29
17	45	32
19	27	32
18	22	34
19	22	32
26	21	33
36	20	35
23	20	33
28	21	33
32	20	35
25	19	35
25	21	33
18	20	33
19	21	32
17	20	31
18	20	33
18	21	31
18	21	36
17	19	34
18	20	33
18	36	32
17	24	33
17	20	32
18	22	34
19	19	27
17	20	30
18	18	28
18	20	27
18	20	24
18	27	26
18	17	24
18	19	28
19	19	26

18	18	26
18	18	25
25	22	32
17	18	35
18	19	25
17	19	33
16	18	46
17	22	38
18	20	28
17	20	35
17	19	27
18	21	30
17	33	27
19	21	33
16	32	27
18	24	29
18	23	27
19	21	24
18	19	26
17	21	25
17	18	26
18	18	26
17	30	26
17	38	28
17	23	28
16	23	27
16	22	25
17	19	27
19	20	27
18	19	26
17	19	26
17	22	25
16	28	32
17	40	28
23	27	30
29	22	30
25	18	28
16	19	27
17	20	27
17	18	27
17	20	28
17	25	26
17	18	40
17	20	39
17	20	32

17	19	58
16	20	31
20	20	25
17	21	29
17	20	28
17	19	27
17	19	30
17	22	30
17	19	25
18	20	27
18	19	25
19	22	27
18	20	27
17	19	27
17	19	30
18	24	39
17	20	33
16	19	34
18	18	45
19	19	39
18	19	48

Hämta vänner

100 vänner	1000 vänner	10000 vänner	100000 vänner
4	24	46	44
4	19	28	30
3	19	32	27
3	14	26	25
3	10	25	28
3	8	27	29
3	8	26	40
2	8	30	37
3	9	32	29
4	8	19	28
3	8	19	31
3	7	18	27
3	6	20	46
3	6	20	35
3	5	21	27
4	6	18	35
3	6	18	44
3	6	20	22
3	6	20	20
3	7	18	21

3	6	21	42
3	7	18	28
2	6	18	21
4	5	17	37
2	6	18	20
4	7	19	40
2	6	22	30
3	7	19	26
3	7	19	33
3	6	20	22
3	5	18	21
3	6	19	21
3	6	18	22
2	6	18	23
4	6	19	24
2	7	18	22
5	5	19	25
3	5	20	35
2	5	25	28
3	6	27	20
3	6	22	22
4	7	22	34
4	6	23	36
3	6	22	32
3	6	22	26
4	7	21	25
2	6	26	24
2	6	22	24
2	8	23	24
2	6	16	20
3	5	20	20
3	9	29	31
3	8	17	41
3	9	18	43
3	10	17	41
2	9	20	47
3	8	20	37
3	11	19	53
3	10	17	32
3	9	20	39
3	10	17	28
2	10	18	26
5	9	18	43
5	9	18	40
5	10	17	28

7	9	18	25
6	8	17	22
5	9	17	20
5	9	20	23
7	9	15	27
5	9	17	25
5	9	16	39
5	9	19	38
5	9	17	40
5	10	18	41
6	9	16	45
5	10	18	37
6	10	16	34
8	9	17	43
5	9	16	54
5	9	24	26
5	9	30	19
5	9	23	40
5	10	17	36
6	9	17	35
5	9	21	49
4	9	30	30
5	8	18	42
5	11	16	44
5	8	16	39
6	9	16	38
6	8	17	41
6	10	16	37
6	9	16	31
7	9	19	21
5	10	17	41
6	10	16	25
6	10	18	21
6	9	17	20
6	6	17	22

Registrering/inloggning

Läsa 1 användare	Registera 1 användare
3	7
1	4
2	5
2	4
3	2
3	2

3	2
3	3
3	4
2	3
3	2
3	3
2	3
3	4
2	3
3	3
1	4
3	4
2	5
3	3
2	3
3	4
3	4
3	4
3	3
3	3
3	4
3	4
2	3
2	3
3	3
2	5
4	2
2	3
4	4
3	3
3	4
3	3
2	3
3	3
2	3
3	3
3	4
2	4
2	2
2	4
2	3
4	4
2	3
2	4
3	3

3	4
2	4
3	4
3	5
2	3
3	2
2	5
2	3
3	4
2	3
2	3
3	3
2	5
2	4
2	2
3	3
3	4
2	5
2	4
3	3
3	3
3	3
3	3
3	3
2	3
3	3
2	3
3	4
2	3
2	3
3	3
2	5
3	4
3	2
2	3
2	3
3	5
3	5
2	3
3	2
2	2
3	4
2	4
4	3
2	3

3	3
3	4
2	4
4	2

Neo4j

Inläsning av meddelande(100000 stycken)

100/100000	1000/100000	10000/100000	100000/100000
6	8	19	62
1	3	7	1820
0	2	7	14
0	1	7	12
1	1	7	11
0	1	7	13
1	1	5	9
0	1	4	11
0	0	5	12
1	1	4	12
0	1	3	12
0	0	3	12
0	1	3	12
1	1	3	12
0	0	3	13
0	1	3	14
0	1	3	15
1	0	3	15
0	1	3	15
0	1	3	14
0	1	3	13
0	1	3	16
1	0	3	15
0	1	3	14
0	1	3	16
0	0	2	16
0	1	3	16
1	1	2	16
0	1	3	15

0	0	2	16
0	1	3	16
0	1	2	13
1	1	2	13
0	0	2	14
0	1	2	14
0	1	2	14
0	0	2	14
0	1	2	13
0	1	2	12
0	0	2	12
0	1	2	12
0	0	1	13
0	1	2	10
0	0	2	12
1	1	2	12
0	1	2	11
0	0	2	9
0	1	2	8
0	1	2	10
0	0	2	8
1	1	3	9
0	0	2	10
0	0	2	9
0	1	1	8
0	0	2	7
0	1	2	8
1	0	2	8
0	0	3	7
0	1	2	7
0	1	2	7
0	0	2	8
1	1	3	7
0	0	2	7
0	1	3	8
0	0	2	7
1	1	2	7
0	1	2	7
0	0	2	8
0	0	2	7
0	1	2	7
0	0	3	7
0	1	2	7
0	0	2	7
1	0	2	7

0	1	2	8
0	0	3	7
0	1	2	8
1	0	2	7
0	0	3	7
0	1	2	7
0	0	2	7
0	0	3	7
0	1	2	7
1	0	2	8
0	0	3	7
0	1	2	7
0	0	3	7
0	0	2	7
0	1	2	7
0	0	3	8
0	0	2	7
0	1	2	7
1	0	3	8
0	0	2	7
0	1	3	8
0	0	2	7
1	0	2	8
0	1	2	7
0	0	2	7
0	0	2	7

Inläsning av meddelande (10000 stycken)

100/10000	1000/10000	10000/10000
7	11	9
1	4	1
1	2	0
0	1	0
1	1	0
0	1	0
1	1	0
0	0	0
1	0	0
0	1	0
1	1	0
0	1	0
0	1	0
0	0	0
0	0	1

1	1	0
0	1	0
0	1	0
0	2	0
1	1	0
0	1	0
0	0	0
0	0	0
1	1	0
0	1	0
0	1	0
0	1	0
0	1	0
1	0	0
0	1	0
0	1	0
0	1	0
0	0	0
0	1	0
1	1	0
0	0	0
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	0	0
0	1	0
0	0	0
0	0	0
0	1	0
0	1	0
0	0	0
0	0	0
1	1	0
0	0	0
0	1	0
0	0	0
0	1	0
0	0	0
0	1	0
1	0	0
0	1	0
0	0	0
0	1	0

0	0	0
0	1	0
0	1	1
0	0	0
1	0	0
0	1	0
0	0	0
0	1	0
0	0	0
0	0	0
0	1	0
1	0	0
0	0	0
0	0	0
0	1	0
0	0	0
0	0	0
0	1	0
1	1	0
0	1	0
0	0	0
0	1	0
0	0	0
0	1	0
1	1	0
0	0	0
0	1	0
0	0	0
0	1	0
0	1	0
1	0	0
0	1	0
0	1	0
0	1	0
0	0	0
0	1	0
0	0	0
0	1	0
0	1	0
0	0	0
0	1	0
0	0	0
0	1	0
0	0	0
0	0	0

Hämta vänner

100 vänner	1000 vänner	10000 vänner	100000 vänner
192	195	158	254
91	96	211	71
89	86	85	65
94	90	91	69
90	99	89	67
169	100	99	69
86	195	87	68
89	87	85	143
86	86	155	73
87	86	82	67
88	87	83	67
87	87	83	66
154	88	84	66
89	88	83	68
88	91	83	68
87	175	83	126
88	91	82	70
88	87	150	64
88	85	83	66
88	85	87	66
147	85	83	65
91	85	84	66
88	85	83	65
88	85	84	65
89	85	82	66
89	87	85	129
93	87	1849	65
223	87	83	68
129	164	88	66
133	86	85	65
112	86	85	66
91	88	84	65
90	88	86	66
92	87	83	65
90	87	84	124
88	85	83	65
149	86	86	65
90	88	84	66
89	87	142	65
89	87	84	65
88	104	87	67
88	87	85	66
89	165	85	66
88	86	84	66

88	86	87	65
153	85	85	124
89	85	86	66
86	86	86	65
87	87	90	65
88	89	111	66
86	84	139	64
86	87	100	66
87	86	92	66
88	85	96	66
88	85	93	65
150	86	90	65
90	85	86	128
89	162	91	66
87	87	90	66
88	86	91	65
87	89	95	65
88	86	88	66
90	101	89	66
98	124	94	65
88	117	109	66
171	126	92	65
88	124	92	66
91	90	89	65
88	86	88	66
90	86	87	122
91	86	90	65
93	92	109	66
94	86	123	69
95	168	147	65
95	85	128	65
109	87	113	65
173	86	93	64
114	87	88	66
107	86	106	65
102	85	88	65
96	88	89	65
92	86	89	123
87	88	90	66
90	89	87	66
96	86	86	66
93	86	89	65
96	86	88	66
96	87	88	68
181	88	89	68

Redis

100/100000	1000/100000	10000/100000	100000/100000
1	0	4	-
0	1	3	-
1	0	4	-
0	0	3	-
0	0	4	-
1	0	4	-
0	1	4	-
0	0	3	-
1	0	3	-
0	1	3	-
1	0	4	-
0	0	3	-
0	1	3	-
1	0	4	-
0	0	3	-
1	1	3	-
0	1	3	-
0	0	3	-
1	1	3	-
0	0	3	-
0	1	3	-
1	0	3	-
0	1	3	-
0	0	3	-
1	0	5	-
0	1	3	-
0	0	3	-
1	0	3	-
0	1	3	-

1	0	3	-
0	1	4	-
0	0	3	-
1	0	3	-
0	1	3	-
0	0	3	-
0	1	3	-
1	0	3	-
0	0	3	-
0	1	3	-
1	0	3	-
0	0	3	-
0	1	4	-
1	0	3	-
0	0	3	-
0	1	3	-
1	0	4	-
0	0	3	-
0	1	3	-
1	0	3	-
0	0	3	-
0	1	3	-
1	0	3	-
0	0	3	-
0	1	3	-
1	0	4	-
0	0	4	-
0	1	3	-
0	0	3	-
1	1	3	-
0	0	3	-
0	0	3	-
1	0	3	-
0	0	3	-
0	1	3	-
1	0	2	-
0	0	2	-
0	1	3	-
1	0	4	-
0	0	6	-
0	1	6	-
1	0	6	-
0	1	6	-
0	0	7	-
0	0	7	-

1	1	7	-
0	0	7	-
0	0	7	-
0	1	9	-
1	0	8	-
0	0	8	-
0	1	6	-
1	0	3	-
0	1	5	-
0	0	3	-
0	0	3	-
1	1	4	-
0	0	3	-
0	0	3	-
1	0	3	-
0	1	5	-
0	0	3	-
0	0	3	-
1	1	5	-
0	0	3	-
0	0	3	-
1	0	3	-
0	1	3	-
0	0	3	-
0	0	4	-
1	1	3	-

Inläsning av meddelande (10000 stycken)

100/10000	1000/10000	10000/10000
17	41	22
11	25	10
11	19	11
11	18	7
11	17	8
15	16	6
11	14	10
10	15	6
9	18	7
11	15	9
11	14	11
10	14	7
11	13	7
11	14	7
10	14	6

10	20	6
10	14	5
10	20	7
11	20	8
10	14	6
10	20	5
10	14	6
11	21	5
11	20	6
11	14	6
9	20	10
10	15	11
9	15	10
10	14	9
9	14	6
9	14	10
8	15	7
9	16	10
10	16	7
9	16	5
8	16	5
8	18	6
8	23	6
10	18	6
8	21	6
7	16	10
8	15	6
6	16	7
10	21	6
7	23	5
9	18	6
5	19	6
5	18	7
5	22	5
5	23	5
6	22	6
8	15	6
5	17	6
5	19	6
9	18	6
6	23	6
5	17	10
6	16	10
6	20	6
10	15	6

5	13	6
9	14	6
5	13	8
5	13	6
5	13	6
5	14	10
5	13	10
6	15	7
5	13	9
5	16	10
5	21	10
5	16	7
5	23	5
4	15	6
5	16	8
6	15	6
5	16	6
10	15	6
9	15	6
5	16	5
5	16	5
5	16	6
5	13	5
5	13	5
6	13	5
5	13	5
5	13	9
5	13	5
5	13	6
6	13	6
5	13	7
5	13	6
6	13	6
5	13	6
5	13	6
5	14	7
5	13	6
5	13	6
5	14	6
5	15	6

**Hämta
vänner**

100 vänner	1000 vänner	10000 vänner
-------------------	--------------------	---------------------

0	1	9
0	1	7
0	1	5
0	1	5
0	0	4
0	1	5
0	1	4
1	0	5
0	1	4
0	0	5
0	1	5
0	0	5
0	1	3
0	1	4
0	0	4
0	1	4
0	1	4
0	0	4
0	1	4
0	0	3
0	1	3
0	0	3
0	0	5
0	1	4
0	0	4
0	1	4
0	0	3
0	1	4
0	0	4
0	1	5
0	0	5
1	0	5
0	0	5
0	1	6
0	1	3
0	0	4
0	1	3
0	0	4
0	1	3
0	0	3
0	1	3
0	0	5
0	1	4
0	0	3
1	0	3

0	1	3
0	0	3
0	1	3
0	0	3
1	0	3
0	1	3
0	0	2
0	0	3
0	1	3
0	0	3

Registrering/inloggning

Läsa 1 användare	Registera 1 användare
8	0
3	0
1	0
2	0
1	0
0	0
1	0
1	0
1	0
0	0
1	1
1	0
1	0
0	0
1	0
0	0
0	0
1	0
0	0
1	0
1	0
1	0
0	0
1	0
1	0
1	0
1	0
1	0
0	0

0	0
0	0
1	0
0	0
0	1
1	0
0	0
0	0
1	0
0	0
0	0
1	0
0	0
1	0
0	0
0	0
1	0
0	0
1	0
0	0
1	0
0	0
0	0
1	0
0	0
0	0
1	0
0	0
0	0
1	0
0	0