

Analysis of No-SQL data-models

Major Project Report

Submitted in fulfillment of the requirements

for the degree of

Master of Technology

in

Computer Science & Engineering

By

Patel Karan H.

(19MCEC10)



Computer Science & Engineering Department

Institute of Technology

Nirma University

Ahmedabad-382 481

Analysis of No-SQL data-models

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Computer Science & Engineering

By

Karan Patel
(19MCEC10)

Under the guidance of

Internal Project Guide:

Prof. Monika shah

Assistant Prof. CSE Department,

Institute of Technology,

Nirma University, Ahmedabad.



Computer Science & Engineering Department

Institute of Technology-Nirma University

Ahmedabad-382 481

Declaration

This is to certify that

- a. The thesis comprises my original work towards the degree of Master of Technology in Computer Science and Engineering at Nirma University and has not been submitted elsewhere for a degree.
- b. Due acknowledgment has been made in the text to all other material used.

- Patel Karan

19MCEC10

Disclaimer

“The content of this paper does not represent the technology, opinions, beliefs, or positions of Intel Corporation, or its employees, vendors, customers, or associates.”



Certificate

This is to certify that the Major Project entitled “**Analyse of No-SQL data-models**” submitted by **Patel Karan H. (19MCEC10)**, towards the fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering, Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date:

Place: Ahmedabad

Prof. Monika shah

Internal Guide

Dr. Priyanka Sharma

Program Coordinator

Dr. Madhuri Bhavsar

HoD - CSE

Dr. Rajesh Patel

Director, IT

Acknowledgements

I would like to express my gratitude and sincere thanks to **Dr.Priyanka Sharma**, PG Coordinator of M.Tech Computer Science and Engineering and **Dr.Madhuri Bhavsar** for guidelines during the review process.

I take this opportunity to express my profound gratitude and deep regards to **Prof. Monika shah**, guide of my inhouse project for his exemplary guidance, monitoring and constant encouragement.

- **Patel Karan**
19MCEC10

Contents

Declaration	III
Disclaimer	IV
Certificate	V
Acknowledgements	VI
Abstract	XI
1 Introduction	1
1.1 Research statement	1
1.2 Research objectives	2
1.3 Thesis Organization	3
2 Background	4
2.1 Databases	4
2.1.1 Relational databases	4
2.1.2 NoSQL Databases	5
2.2 Power-API	8
3 LITERATURE REVIEW	9
3.1 Energy Measurement Approaches and Tools	9
3.2 power estimation programs	10

3.3	Energy Consumption and applications design	11
3.4	Power utilization and reliability	15
3.5	Potential of SQL and No-SQL databases	16
4	methodology	19
4.1	dataset information	19
4.1.1	Modelling techniques	20
4.2	Data-models	20
4.3	architecture of finding energy consumption by different data-models .	28
5	Results	29
5.1	Result table	29
5.2	Observation strictly based on above data-model's result	29
5.3	Improvised model number 4	30
6	Future Tasks	32
7	Conclusion	33
	References	34

List of Figures

2.1	MongoDB architecture	8
4.1	Parent referencing	22
4.2	child embedding	23
4.3	two way referencing	24
4.4	parent embedding	25
4.5	child referencing	26
4.6	denormalising	27
4.7	Proposed Solution	28
5.1	improvised data-model	31

List of Tables

Result	29
Improved Result	30

Abstract

Nowadays, IT industry is growing within all other sectors. However as per report, the decade before , the total Energy consumption of ICT(Internet and communication technology) devices were consuming was less than a 3% of total consumption of all devices but with the revolution of industry 4.0 , data becomes a big and really important aspect so that not only with the hardware but it is essential too to develop a software which consumes less energy. By making this project, we have tried to contribute in that area.

Chapter 1

Introduction

With the growth of the Internet, the majority of companies have begun to use databases to store and analyse vast data sets. applications with large databases are applicable in each and every industry nowadays likewise financial, retail, education, and communications, manufacturing, utilities, and transportation. However, with these much popularity and ample amount of usage ,still very little information about power usage of programs and, especially ,of their databases and data-models. Therefor, the experiments have been carried out to monitor the power usage footprint and the execution time of the data-models. Two different well-known databases are recently adopted in majority of applications. 1) relational 2)non-relational databases. Majority demanding part of the data-models is to get potential result in reading and writing data. nonetheless, compressing value of the energy consumption for the applications is a highly prioritised objective for any organisation and it will grow more and more.

1.1 Research statement

Green IT has come to be associated with systems that use less resources than prior iterations of related devices or materials. In green information technology, software has started to play a significant role. As a result, the programme that instructs the

system to begin processing is the primary source of ICT power consumption. We will stop the pattern of machines consuming more electricity per year if software can be programmed so that hardware can perform its job with less power. The project demonstrates that making more and more efficient use of software will result in significant cost savings for greener software. The opportunity for savings ranges from 35 percent to 87 percent, depending on the situation.

While, as far as we know, the database handles the majority of the work, the task of assessing the energy usage of the database's different data models has yet to be accomplished. The results of different data models are often intuitive and cannot be confirmed.

We will calculate the power usage of various data models of a single NoSQL database, mongodb, in this article. We used more than five data models to arrive at this estimate, and we tried to access the same data when controlling the power usage of each data model..

1.2 Research objectives

Our specific research objectives are as follows:

- a. To find a suitable tool to measure a energy consumption is one of the important objective of this project.
- b. The another objective is to find a optimistic data-model to the two factors called energy consumption and response time.

1.3 Thesis Organization

The rest of this thesis is organised as follows:

- Chapter 2 illustrate the background knowledge of relational database and No-SQL databases.
- Chapter 3 express the literature review for energy consumption tools and technology.
- Chapter 4 gives data regarding implementations of data-models and information about data-set .
- Chapter 5 in which, all the results were collected through tabular form and draw some points related to analysis.
- Chapter 6 includes future work
- Chapter 7 Conclusion

Chapter 2

Background

In this part, the various information bases have been covered, just as the contrasts among SQL and No-SQL data sets. This part additionally contains essential insights regarding Force Programming interface just as an outline of how to utilize it.

2.1 Databases

The information base administration framework (DBMS) [2] is an efficient framework for effectively recuperating and backing up records. In this segment, we present the two sorts of dbs (data sets) that we utilized in our examination: Since they are the most widely recognized data sets open today, one is a social data set, MySQL, and the other is a No-SQL data set, MongoDB.

2.1.1 Relational databases

A relational database is a collection of data structured by constructs such as record, column, and tabular format. A relational database [3] provides the programming interface. Scaling a relational database architecture is also a difficult task. However, relational databases support features such as indexing, which aids in the sorting of records. :

```
id=7 — name=messi — lastName=zepa — country=British—birthYear=1987 —
club=Gold Madril — location=brazil
```

Unlike No-sql databases, relational databases follows the ACID properties and ACID transactions too. The Four ACID properties are discussed as follow. [4]:

- Atomicity: "each exchange ought to be executed completely, else it isn't executed. As needs be, it is called a nuclear exchange".
- Consistency: "each exchange takes the data set starting with one substantial state then onto the next".
- Isolation: "every exchange is detached from some other exchange".
- Durability: "every exchange is saved by the framework even in case of a framework disappointment or restart. At the end of the day, an exchange should persevere and consequently never be lost".

2.1.2 NoSQL Databases

Non-relational and appropriated information bases [6] are two highlights of NoSQL data sets [5.] NoSQL data sets, in contrast to social data sets, follow the CAP hypothesis [7], which means "Consistency, Accessibility, and Segment Resilience." NoSQL data sets are disseminated information stores, for example, record stores (e.g., MongoDB [8]), and chart data sets BigTable [10], Dynamo [9] are instances of putting away and key-esteem stockrooms. MongoDB information base [8] has been researched in our technique as the clarification for the plenitude of recognizable proof in NoSQL data sets accessible today; this data set is utilized by a portion of the world's most notable organizations, including many big companies.

MongoDB

MongoDB contrasts from social data sets by they way it works. The "definition and language" are found in Table 2.1. A MongoDB execution is an assortment of

information bases that can't be changed. As opposed to social information bases, which store information as tables for every data set, MongoDB stores information as sets. Every arrangement has a bunch of decides that should be followed. A record with the BSON format is an assortment of hard and simple fields. A control is a key-esteem pair in which the keys are consistently strings and the qualities are consistently strings, objects, booleans, or whole numbers. Any record, in the same way as other information bases, has a naturally created discipline called `id`, which appoints the record a totally novel `id`.

an information of hockey player is in below portion (JSON Fromat):

```
{
  _number: 6000e717b700k23,
  FN: messi,
  LN: fleko,9
  representator:brazil,
  BD: 1987,
  Ads: {
    c=jhls,
  }
}
```

Atomic worth are considered. The instance underneath indicates the identical instance cited formerly however which include a few complicated worth:

```
{ _number: 6000e717b700k23,
  FN: messi,
  LN: fleko,9
  representator:brazil,
  BD: 1987,
  Ads: {
    c=jhls,
    street: shamino lintroa - 78750 kapdegebqs. },
  club: Gold Madril,
```

location: brazil,

hobbies: [reading, playing football]

} MongoDB has an element called `embedding` that permits you to store a record inside a record. In spite of the fact that MongoDB doesn't have any outlines, the names of the data sets and the names of the assortments in which they are put away are protected as metadata. Moreover, the recorded files are saved as metadata. MongoDB upholds lists. They are additionally pointers. Indeed, even installed fields, for example, `with.town` from the second model above, might be recorded. Every irregular get right of section to recovery and assortment questions is supported by the fields referenced. Inside the existence of a set, lists might be produced whenever.

The documents' identification discipline is mechanically recorded. MongoDB's setup is partitioned into three classifications, as found in Figure 2.3.4: "MongoDB Query Language," "MongoDB Information Model," and "MongoDB Stockpiling Motors." Clients may utilize the MongoDB Query Language to access and control their records in forefront ways. This insightful and useful frameworks may profit by the system. The MongoDB Information Model simplifies it to search for and join information of any kind. Beside that, the MongoDB Information Model permits you to alter the data set diagram without influencing the general effectiveness of the question execution. MongoDB is in its third level. The MongoDB Stockpiling Motor is addressed by the system. It upholds WiredTiger, MMAPv1, In-memory, Encoded, and the third birthday celebration birthday festivity motor, among others. We should make reference to that MongoDB gives WiredTiger as a default carport motor, which gives the best in general carport effectiveness of all the carport motors recorded.

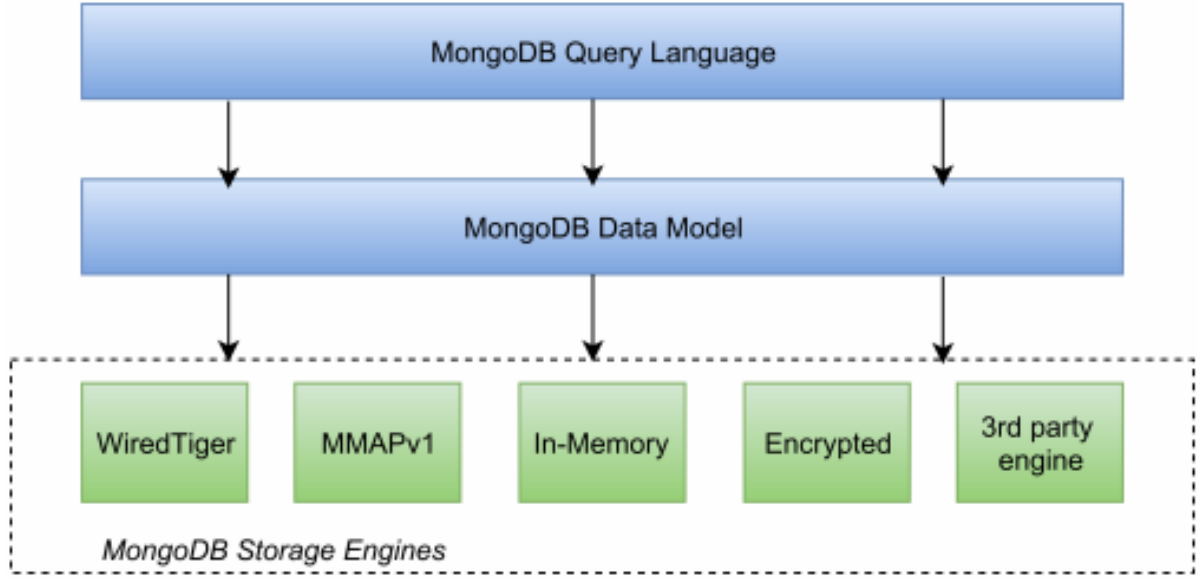


Figure 2.1: MongoDB architecture

2.2 Power-API

Force Programming interface is a profiler that offers power data (later the parameter's are transferred to milli watt) with regards to information rendition for each machine segment (e.g., CPU, memory, and so on) [11]. Force Programming interface utilizes sensors and scientific designs for its force estimation. Nouredine et al.

This Programming interface permits you to decide how much force the focal handling unit consumes the force. Nouredine et al. [12] directed an examination to evaluate the precision of the Force Programming interface profiler utilizing Powertaker [13]. This present trial's discoveries uncovered that There are just minor contrasts between the power utilization assessed by Powertaker and the power gauges given by the Force Programming interface profiler [11]. We tend to utilize Force Programming interface profiler for our investigation along these lines (i.e., high exactness). Besides, the Force Programming interface profiler doesn't present commotion in the information, as exhibited by a trial directed by Abtahizadeh et al. [14].

Chapter 3

LITERATURE REVIEW

In this part, we'll take a gander at research on programming program power computations and the effect of data sets and styles on in general bundle results. The force size cycles and apparatuses are talked about in Section 3.1. The utility format can affect the force consumption, as talked about in area three.2. Area 3.3.3 addresses troublesome conditions including power admission and security. Segment 3.4.4 features related works that address the effect of cloud styles on bundle results. The effect of information bases on the general yield of bundles is the focal point of Section 3.5.

3.1 Energy Measurement Approaches and Tools

According to researcher [33] proposed "an assessment principally based absolutely strategy that ascertains the power admission of a product program thing the utilization of the ensuing detailing from Equation" [15].

total $E = E(n) + E(c) + E(s)$;

$E(n)$ = energy consumed by network,

$E(c)$ = energy consumed by different components like cpu,memory

$E(S)$ = energy consumed by system like Os,process

According to Kansal et al report[16] conjointly arranged partner degree way to deal with compute the energy utilization of an program. the vital qualification between their methodology and furthermore the methodology is that it licenses for the assessment of energy utilization even once the machine is on pause and inactive states. program burn-through energy when anticipating demands from the plate. the general power utilization of the program .

According to researchers [17] report presented partner approach upheld sensors. data collector are utilized between the apparatus on that estimations are made and consequently the force supply (i.e., the wellspring of electrical flow). The data collector gather right information from the office source. The gathered information, keep in a very information variety worker, will be, for instance, a voltage estimation. The activity of the sensors isn't perpetually persistent anyway occasional throughout time frames. when running the deliberate programming bundle, the data filter a piece of the sensors and the product are connected.

3.2 power estimation programs

here the detailed description of power consumer profilers has been given in the section.

G-Tracker

researchers [19] acquainted an apparatus alluded with as G-Tracker. G-Tracker will appraise the capacity utilization of the focal processor all through the running of a PC code. G-Tracker was wont to show programming from totally various classifications: information handling programming, sound and web search machine. experiments depicted that web Voyager is that the best web program among every one of the tested programs.

Powercope

researcher [20] presented a profiler of energy utilization known as scopeEX. scopEX was planned predominantly for estimation versatile programs. Powerscope is incredibly equivalent to PowerAPI inside the feeling that it gauges the energy utilization of a code (i.e., versatile applications) at the strategy level. Notwithstanding, Powerscope follows an extraordinary assessment approach (than PowerAPI) to calculate the energy devoured by various techniques among a chose interaction. This profiler utilizes 3 diverse programming parts for its activity [15]:.

- Power Screen that capture power results from a computerized multimeter.
- Power instrument that ascertains the Power utilization of a code abuse the power tests examined by the Energy Screen and furthermore data tested by the Framework Screen.

joulemeter

Joulemeter [21], [22] could be a bundle instrument created by Microsoft which will assess the energy devoured by personal computer, a worker, or a setup system. This instrument moreover permit displaying the effect of various kind of parts, (for example, focal preparing unit use, memory usage, or screen brilliance) on absolute energy utilization. Curiously, Joulemeter might be acclimated gauge the energy utilization in an extremely data focus or perhaps all through programming advancement.

3.3 Energy Consumption and applications design

According to horse et al finding[23] "utilized a favored net site, StackOverflow, as essential information supply to realize the product bundle specialists' view on complexities with respect to the power utilization of their program. it has been showed that the amount of queries on power utilization duplicated by many numbers exclusively on special years (from 2014 to 2015). it has been suggested that programmers

are effectively looking for pointers concerning the energy power of their applications”.

According to Abtahizadeh et al report [14]. ”same kind of work, they estimated the power utilization of uses exploitation Force Programming interface profiler. Notwithstanding, they exclusively considered MySQL data set. They showed that cloud examples will successfully downsize the energy utilization of any application. They depicts furthermore that the execution of the local information Intermediary Example can surprisingly optimised the power effectiveness of a any program which this example is a ton of material for cloud applications taking care of enormous solicitations of peruse loads. They moreover tracked down that a blend of the confined information Intermediary design with the distributed-Based Switch doesn’t significantly affect energy utilization”

according to Manotas et al report[24] ”led AN experimental examination inside which they explored the effect of 4 web workers (i.e., crossbreed, jaguar, meage) on the power utilization of a web program. it depicts that the power utilization of a web program relies upon the net worker wont to deal with demands. They conjointly showed that the effect of the net worker relies upon the choices that are utilized. each web worker will increment or reduction the energy utilization of the net application, looking on the highlights that it’s executed”.

According to Hammes et al [25]report ”given a relative investigation of SQL and NoSQL data sets inside the cloud, any place they featured the presentation of each PostgreSQL informationbase and MongoDB data set, upheld on a cloud worker. during this paper, they analyzed the exhibition of the data sets abuse two particular true circumstances: the essential situation addresses a very organized information and furthermore the subsequent situation addresses unstructured information. By testing Create,Read, Update, and Annihilate tasks for the 2 data sets with the 2 unmistakable situations, they found that mongodb has higher performance than any other. ”.

according to Bunse et al report[26] ”looked at the energy utilization of shifted

arranging calculations, running on AN inserted framework. The arranging calculations examined are some of the algorithm. The examination show that value addition sort is that the additional power-productive calculation. They likewise discovered no relationship of the energy utilization and also the time nature of the calculations. implying that calculations that devour a ton of energy are not basically individuals who set aside longer effort to be executed”.

According to Arunagiri et al report[27] ”looked at all totally changed executions of recipes used to take care of the overall sound system coordinating with issue. The measurements utilized for examinations are: the quantity of energy burned-through, the ordinary stretch and also the overall least cost accomplished. Their outcomes show that the chart cut algorithmic guideline performs more than the mimicked hardening calculation”.

According to Sahin et al report[28] ”explored the energy productivity of fifteen primary, action related creational style designs, authorized in partner program. for every example, it analyzed power utilization of the adaptations of the apparatus previously and once applying configuration blueprint. Their result depict that vogue designs essentially affect energy utilization. Nonetheless, the effect on energy utilization of changed kinds of configuration designs isnt the equivalent. certain plan designs like Decorator can build the energy use of an application by up to 700%. Comparative examination was led by Bunse et al. [48]. Nonetheless, they researched the energy intensity of style designs implemented in cell phones applications. The designs considered in their investigation are: Veneer, Unique Production line, On-looker, Decorator, Model, manage Technique. They estimated the energy utilization of uses running on by and large very surprising cell phone models (e.g., Samsung System S2, Nexus One) abuse the PowerTutorApp instrument from Michigan College. Their outcomes show that now and again, style examples can altogether build the energy utilization of versatile applications. Specifically, similarly to Sahin et al. [28], they found that Decorator design effectively will build the energy utilization of portable applications”.

according to Sahin et al report[29] "led partner observational examination at stretches that they explored the impact of logic on the energy utilization of uses. The examination was directed misuse 200 program carrying out six usually utilized refactorings. Their outcomes show that code refactorings have an influence on the energy utilization of uses. huge loads of explicitly, they tracked down that every one the tried code refactorings can possibly every expansion related cut back the energy utilization of an application, except for Concentrate confined value, which reliably debilitated the power utilization of the examined program".

According to Procaccianti et al report[30] "led an exact examination that researched the effect of 3 very surprising ORM approaches on the energy utilization of uses. ORM means "Article Social Planning", that allows a connection between a PC data set and item arranged programming. this strategy is wide utilized for improving a table hang on during a relative data set in a basically consistent article by means of its credits. In Java, we will make reference to "Sleep" among the structures ordinarily wont to play out this undertaking. However, this method works with the control of social information bases exploitation object-arranged programming, anyway during this paper, they showed that this procedure envelops an adverse outcome on the power consumption of partner program. in extra subtleties, the creators directed their investigations explicitly on basic applications, testing 3 very exciting ways that control a Structured Query Language informationbase—electronic database—on-line database—computer database—electronic data administration: plain SQL inquiries inside the stock code, and Move and TinyQueries. Moreover, they target-chasing their examinations mulling over three factors: similar three methodologies, the components of the relative data set, and furthermore the type of the inquiry wont to separate the predetermined information from the social data set (SQL). The pre-owned solicitations control the supported Muck tasks (i.e., Make, Read, Update, Erase). Their outcomes are introduced essentially on 2 tomahawks: the energy power and furthermore the execution season of the basic web-application. Curiously, results are showing the tried methodologies gives pros and compromise as

far as running time and power utilization: the Impel composition is that the highly power exceptional and is time consuming among the 3 tried methodologies. the principal proper methodology is the utilization of plain SQL inquiries inside the ASCII text document because of it's least energy burning-through and the fastest by and large the experiments. The third methodology, that of TinyQueries, performed far superior than Impel, anyway marginally more terrible compare to the unadulterated tructured query langauge inquiries".

3.4 Power utilization and reliability

As of late, specialists have begun examination the energy worth of safety apparatuses. mik previously mentioned [31] "made partner degree impact screen that gathers power tests and characterizes a past filled with the energy utilization from these examples. They moreover projected partner degree information instrument that creates an impact signature, when commotion sifting. to keep with exploratory outcomes on a force unit iPAQ running a Windows versatile operating system, the device accomplishes a genuine positive pace of 99% inside the grouping of malware focusing on cells".

one of the researcher [32] extended a path for processing the energy utilization examples of 2 cell phones part of the sustem i.e., central processor and remote well managed card. The arranged procedure has been carried out and tried all through a ping-flood assault led all through the execution of an authentic application. Test results show that the arranged technique is solid, precise, and dependable for location the frequency of such assault.

Hoffmann et al [33] examined the quality force utilization of shifted parts of normal robot cell phones, and arranged that power utilization fluctuates enormously in follow. They tried to estimate the power consumption devoured by virus and ended that in follow, the extra force devoured by virus is basically inadequate to be distinguishable exploitation rate of estimations.

As per data, [34] presented a creative origination expressed as "Energy-mindful Interruption Recognition Frameworks", that recognizes malevolent conduct in cell phones as per their energy impression. To execute this thought, the energy utilization of the numerous equipment segments of a cell phone should be estimated with an extra degree of exactness. They researched a couple of estimations draws near: elevated level and shallow level. hectolitre scale quantify can give significant data not only as far as energy utilization, but rather set up the flexibleness to detach the office utilization of each part. in qualification to the LL approach, the proportions of the hectolitre approach don't have all the earmarks of being quick. The LL approach gives speedy information on the utilization of the entire framework at a harshness of 260 ms. in this scenario such count does not separate the capacity utilization of every part, the creators selected a mix of the 2 methodologies, that they expressed as optimistic level mensuration. cubic centimeter utilizes an undeniable level Programming interface to require activity straightforwardly from the battery driver to have each quick estimations and data on the utilization of each part. With this methodology, they had the option to only check ping-flood assaults.

Palmieri et al [35] report concerning partner assault focusing on cloud administrations providers, where assailants played out a forswearing of administration that represent considerable authority in (energy-situated disavowal of administration). all through this assault, the objective is to utilize offered assets to increment impressively the energy utilization while not setting off insurance instruments in screens. this kind of assault can monetarily affect cloud administrations providers.

3.5 Potential of SQL and No-SQL databases

Specialists examined the SQL and MongoDB data bases over time [57]. According to our knowledge, no previous research has looked at the impact of NoSQL data sets on cloud-based application use in the workplace.

According to researcher report.. [25]" gave a near investigation of Databases

information bases inside the cloud, inside which they featured the presentation of each the PostgreSQL data and consequently the MongoDB information authorized on a worker inside the cloud. 2 totally unique genuine circumstances: the essential circumstance addresses very organized information and in this manner the subsequent situation addresses unstructured information. when testing the tasks "Make", "Read", "Update" and "Obliterate" (approx. 1,400,000 tasks) for the 2—the 2 data sets with "In two totally various circumstances they found that PostgreSQL information bases perform higher than data sets. MongoDB information in cloud conditions".

According to jarat el. report. [36] "have assembled distributed a relative report on this in vogue information, id est MySQL and MongoDB. They analyzed absolutely totally various activities (Muck tasks) to imagine the presentation of the 2 data sets. They utilized 3 classes of datasets (I., Little, Medium, and Huge). Like our outcomes, the creators have shown that the MongoDB information beats MySQL for cutting edge inquiries, especially those with numerous joins. They also showed that MySQL data sets performed higher than MongoDB data sets for little measures of data. These outcomes worked out certain tips for code designers. it's proposed to utilize MySQL information bases instead of MongoDB for transitional information if there aren't any exceptional inquiries, and MongoDB as opposed to MySQL knowledgebases for medium information that include confounded questions and joins".

according to Loanis report [37] "arranged a cloud-empowered structure for perceptive NoSQL information bases. His arranged system was applied to a couple of popular NoSQL information bases: HBase, vaticinator and Riak. The outcomes showed that the prophetess information bases ar speedy writable. savvy while adding hubs, however not a part of correction reason. HBase is that the speediest of the 3 data sets inspected. HBase scales well for hub increments. Notwithstanding, the Riak information bases precisely rebalance the hubs. The creators gave a model execution of their programmed bunch that produces it easy to perform programmed

flexible activities from any NoSQL motor. The arranged model offers engineers and architects the opportunity to picture and confirm the level of quantifiability of a casualty application in these noSQL data sets”.

According to vory la. report. [38]” presented Partner in Nursing Partner in Nursing way to deal with learning the actual properties of NoSQL information upheld during a high cloud setting. The projected methodology was assessed in MongoDB, HBase and prophetess data sets utilizing a practical burden. Rackspace1 utilizes since the cloud foundation, a favored Wikipedia information wont to produce the investigation dataset, it should consistently be referenced that this examination gives measurements just to frameworks that ar deceiving, as NoSQL data sets, that scale. As a limit to the current work, these snap measures don’t work with a decreasing framework. The creators have shown that the specialized decision and furthermore the vibe of everything about NoSQL information inspected affect the expansion of most recent hubs (i.e on the scaling)”.

Chapter 4

methodology

First of all, the benchmark data-set and six types of data-model have been created with the help of python script. Then, we have tried to access data through queries. however, queries are designed in a way that we can access the data through all the relationships like first two queries are focusing on extracting data from employee relationship, another two queries are made to collect the data from department with reference of company whereas one query is made up to just interact with company and department irrespective of employee. so, this is how unbiased fusion of queries has been taken to perform proper analysis.

4.1 dataset information

this is the generated benchmark dataset of multinational 10 companies where each company includes 50 departments and 1000 employees in each departments so overall, it is a 5,00,000 entries of employees. However, we have just trying to replicate the arrangement of employees of Amazon company as it has around 500 to 600 thousands employees.

4.1.1 Modelling techniques

- **embedding:** In a simple language embedding means appending a document inside a document. Suppose, one person has multiple addresses so rather than making multiple documents, all the addresses documents can be embedded easily inside the "person's" document. The embedded operation can be performed two ways likewise child embedding and parent embedding. Above, explained scenario is known as "child embedding" and on the other way around, if we append "person's document" inside "addresses document" then it can be referred as parent embedding.
- **referencing:** In this concept, rather than embedding whole document, only reference can be added inside a document. Basically, there are three types of referencing known as parent reference, child reference and two way referencing. Let's understand this scenario with the same example above explained. Suppose, without adding all the information of person, if we add just person's Id inside the addresses document then it can be referred as "Parent referencing". On the other way around if we add the addresses document's Ids inside person's document then it can be known as child referencing. Well, for double referencing, it's required to make a separate document in which just references (IDs) are added without extra information, can be known as double side referencing.
- **denormalising:** It is a very basic concept as all the information of separated documents are embedded inside one document. As an example, if all the information of person and addresses are embedded inside a single document then it is known as "denormalising".

4.2 Data-models

" company has departments and all departments have employees".

there are three document collections. Company, department and Employees. so company collection is referred as parent document collection and the first level child of company is department and the second level child is employees. So, basically for the employee document, second level parents are department and first level parents are company.

- **parent referencing (1)**= In this model, The reference of company (company id) is added inside the child(department,employee) collection.
- **child embedding (2)**= in this model, Child documents(department and employees) are added inside the parent document(company document).
- **Two way referencing (3)**= in this model, basically references of child and parent documents (employee_ID,company_ID and department_ID) is embedded inside the single document.
- **parent embedding (4)**= in this model, the parent documents(company, department) is embedded inside the child document(employee).
- **child referencing (5)**= in this model, the reference of child document(employee_ID) is added inside parent document(company,department) .
- **denormalization (6)**= in this model, all the information related to company,department and employee is embedded inside a single document.


```
> db.dept.findOne()
{
  "_id" : 1,
  "dept_name" : "debate",
  "dept_projs" : 9674,
  "dept_city" : "Johnsonside",
  "emps" : [
    {
      "salary" : 144224,
      "city" : "Hallville",
      "_id" : 1,
      "emp_name" : "Leah Moody"
    },
    {
      "salary" : 83884,
      "city" : "Port Robinchester",
      "_id" : 2,
      "emp_name" : "Megan Burnett"
    },
    {
      ...
    },
    {
      "salary" : 247718,
      "city" : "Joshuaberg",
      "_id" : 999,
      "emp_name" : "Brandy Garcia"
    },
    {
      "salary" : 175376,
      "city" : "Tracyfurt",
      "_id" : 1000,
      "emp_name" : "Eileen Hill"
    }
  ],
  "cmp_id" : 1
}
```

Figure 4.1: Parent referencing

```

db.cmp.findOne()
{
  "_id" : 1,
  "cmp_profit" : 804256,
  "depts" : [
    {
      ...
      "dept_name" : "security",
      "dept_projs" : 9972,
      "_id" : 10,
      "dept_city" : "Cartertown",
      "emps" : [
        {
          "salary" : 197940,
          "city" : "Cherryborough",
          "_id" : 9001,
          "emp_name" : "Jenna Cooke"
        },
        ...
        {
          "salary" : 194631,
          "city" : "Lake Davidmouth",
          "_id" : 50000,
          "emp_name" : "Tanner Johnston"
        }
      ]
    }
  ],
  "cmp_country" : "Luxembourg",
  "cmp_name" : "Willis LLC"
}

```

Figure 4.2: child embedding

```
> use ss4
switched to db ss4
> show collections
cdes
cmp
dept
emp
> db.cdes.findOne()
{
  "_id" : ObjectId("5ebd11d2b29e5084752d07d4"),
  "emp_id" : 1,
  "dept_id" : 1,
  "cmp_id" : 1
}
```

Figure 4.3: two way referencing

```
> use ss5
switched to db ss5
> show collections
emp
> db.emp.findOne()
{
  "_id" : 1,
  "salary" : 144224,
  "city" : "Hallville",
  "emp_name" : "Leah Moody",
  "dept" : {
    "dept_name" : "debate",
    "dept_projs" : 9674,
    "_id" : 1,
    "dept_city" : "Johnsonside",
    "cmp" : {
      "cmp_profit" : 804256,
      "cmp_country" : "Luxembourg",
      "cmp_name" : "Willis LLC",
      "_id" : 1
    }
  }
}
```

Figure 4.4: parent embedding

```

> db.cmp.findOne()
{
  "_id" : 1,
  "cmp_profit" : 804256,
  "depts" : [
    {
      "dept_name" : "debate",
      "dept_projs" : 9674,
      "_id" : 1,
      "dept_city" : "Johnsionside",
      "emps" : [
        1,
        ...
        999,
        1000
      ]
    },
    {
      "dept_name" : "method",
      "dept_projs" : 9385,
      "_id" : 2,
      "dept_city" : "Michaelshire",
      "emps" : [
        1001
        ...
        49999,
        50000
      ]
    }
  ],
  "cmp_country" : "Luxembourg",
  "cmp_name" : "Willis LLC"
}

```

Figure 4.5: child referencing

```
> use ss10
switched to db ss10
> show collections
emp
> db.emp.findOne()
{
  "_id" : 1,
  "emp_name" : "Leah Moody",
  "cmp_name" : "Willis LLC",
  "dept_id" : 1,
  "cmp_id" : 1,
  "cmp_country" : "Luxembourg",
  "salary" : 144224,
  "city" : "Hallville",
  "dept_name" : "debate",
  "dept_projs" : 9674,
  "dept_city" : "Johnsonsides",
  "cmp_profit" : 804256
}
```

Figure 4.6: denormalising

4.3 architecture of finding energy consumption by different

here, The plan of finding energy foot-prints is mentioned by below architecture.

- Architecture to consume energy consumption and response time.

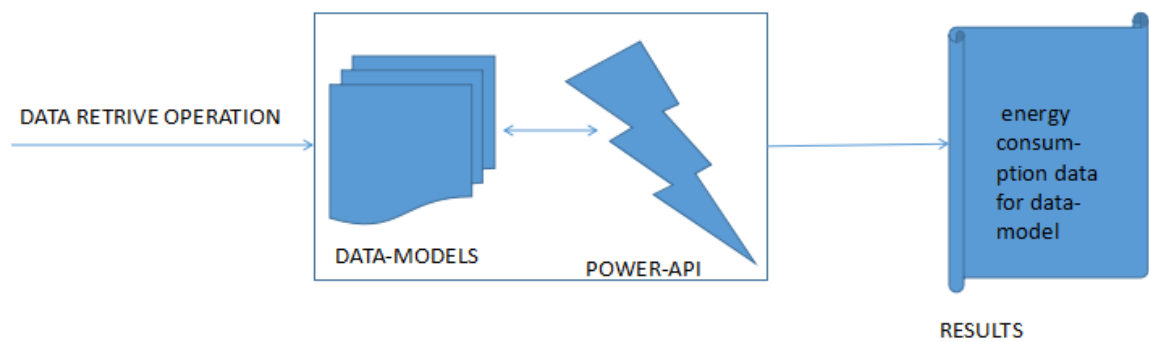


Figure 4.7: Proposed Solution

Chapter 5

Results

5.1 Result table

model name and no.	energy consumptions(in millie-watts)	Responce time(secs)
parent referencing(1)	5770	1740
child embedding(2)	5960	1920
two way referencing(3)	5044	666
parent embedding(4)	6030	43
child referencing(5)	4940	477
denomalisation(6)	6270	32

5.2 Observation strictly based on above data-model's result

- The decade before model number(6) was the best choice as we used to compare only one parameter and that was response time but in this decade, software developers are more aware about energy consumption too.
- if we analyse others models than number (6), we can see that model number 1,2,3 and 5 consumes better less energy than model number 6 but the prob-

lem is response time as it is very hard parameter to sacrifice so they are not providing optimistic result.

- However, we focus on model number 4 then it has less energy consumption than model number 6 but response time is little bit more so we can focus on that model and we tried to optimise that model to get some better response time and got success too.

5.3 Improvised model number 4

- **Improvised model no. (4)** = in this model, we have tried to add another document collection. In which, Applied parent embedding concept on just company and department documents so that, the queries only related with department and company can be execute faster and use some less resources.

model name and no.	energy consumptions(in millie-watts)	Response time(secs)
parent embedding(4)	6020	33

Observation

- In the improved version of data-model, we can see that the energy-consumption and response time both are much optimised. So Likewise we can consider this kind of data-model to store and retrieve information.

```

> db.cmp.findOne()
{
  "_id" : 1,
  "cmp_profit" : 804256,
  "cmp_country" : "Luxembourg",
  "cmp_name" : "Willis LLC"
}
> db.dept.findOne()
{
  "_id" : 1,
  "dept_name" : "debate",
  "dept_projs" : 9674,
  "dept_city" : "Johnsonside",
  "cmp" : {
    "cmp_profit" : 804256,
    "_id" : 1,
    "cmp_country" : "Luxembourg",
    "cmp_name" : "Willis LLC"
  }
}
> db.emp.findOne()
{
  "_id" : 1,
  "salary" : 144224,
  "city" : "Hallville",
  "emp_name" : "Leah Moody",
  "dept" : {
    "dept_name" : "debate",
    "dept_projs" : 9674,
    "_id" : 1,
    "dept_city" : "Johnsonside",
    "cmp" : {
      "cmp_profit" : 804256,
      "cmp_country" : "Luxembourg",
      "cmp_name" : "Willis LLC",
      "_id" : 1
    }
  }
}

```

Figure 5.1: improvised data-model

Chapter 6

Future Tasks

We can create different data-models in other No-SQL databases also likewise Hbase,cassandra and HANA.Additionally, we can increase the size of database to check the energy-foot prints for more in-depth study.

Chapter 7

Conclusion

I have learned new things about No-SQL database,why mongodb is best among all and how to design a data-models.Apart from this, we can conclude that nowadays, it is required to design a data-model which consume less energy along with less response time.Moreover, One of the most exciting feature about data-model is , we can perform changes as per our requirement which makes easy for us to get desire results.

References

- [1] A. Bourdon, A. Nouredine, R. Rouvoy, and L. Seinturier, “Powerapi: A software library to monitor the energy consumed at the processlevel,” *ERCIM News*, vol. 2013, no. 92, 2013.
- [2] C. Data, *An introduction to database systems*. Addison-Wesley publ., 1975.
- [3] S. W. Dietrich, D. Goelman, C. M. Borrer, and S. M. Crook, “An animated introduction to relational databases for many majors,” *Education, IEEE Transactions on*, vol. 58, no. 2, pp. 81–89, 2015.
- [4] H. Garcia-Molina, *Database systems: the complete book*. Pearson Education India, 2008.
- [5] J. Han, M. Song, and J. Song, “A novel solution of distributed memory nosql database for cloud computing,” in *Computer and Information Science (ICIS), 2011 IEEE/ACIS 10th International Conference on*. IEEE, 2011, pp. 351–355.
- [6] A. Moniruzzaman and S. A. Hossain, “Nosql database: New era of databases for big data analytics-classification, characteristics and comparison,” *arXiv preprint arXiv:1307.0191*, 2013.
- [7] S. Gilbert and N. A. Lynch, “Perspectives on the cap theorem.” *Institute of Electrical and Electronics Engineers*, 2012.
- [8] P. Membrey, E. Plugge, and D. Hawkins, *The definitive guide to MongoDB: the noSQL database for cloud and desktop computing*. Apress, 2011.

- [9] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: amazon’s highly available key-value store,” in *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6. ACM, 2007, pp. 205–220.
- [10] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, “Bigtable: A distributed storage system for structured data,” *ACM Transactions on Computer Systems (TOCS)*, vol. 26, no. 2, p. 4, 2008.
- [11] A. Nouredine, A. Bourdon, R. Rouvoy, and L. Seinturier, “A preliminary study of the impact of software engineering on greenit,” in *Green and Sustainable Software (GREENS), 2012 First International Workshop on*. IEEE, 2012, pp. 21–27.
- [12] “Runtime monitoring of software energy hotspots,” in *Automated Software Engineering (ASE), 2012 Proceedings of the 27th IEEE/ACM International Conference on*. IEEE, 2012, pp. 160–169.
- [13] W. Vereecken, W. Van Heddeghem, D. Colle, M. Pickavet, and P. Demeester, “Overall ict footprint and green communication technologies,” in *4th International Symposium on Communications, Control and Signal Processing (ISCCSP 2010)*. IEEE, 2010.
- [14] S. A. Abtahizadeh, F. Khomh et al., “How green are cloud patterns?” in *Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance*. IEEE, 2015, pp. 1–8.
- [15] A. Nouredine, R. Rouvoy, and L. Seinturier, “A review of energy measurement approaches,” *ACM SIGOPS Operating Systems Review*, vol. 47, no. 3, pp. 42–49, 2013

- [16] A. Kansal and F. Zhao, “Fine-grained energy profiling for power-aware application design,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 2, pp. 26–31, 2008.
- [17] A. E. Trefethen and J. Thiagalingam, “Energy-aware software: Challenges, opportunities and strategies,” *Journal of Computational Science*, vol. 4, no. 6, pp. 444–449, 2013.
- [18] A. Pathak, Y. C. Hu, and M. Zhang, “Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof,” in *Proceedings of the 7th ACM european conference on Computer Systems*. ACM, 2012, pp. 29–42.
- [19] N. Amsel, Z. Ibrahim, A. Malik, and B. Tomlinson, “Toward sustainable software engineering (nler track),” in *Proceedings of the 33rd International Conference on Software Engineering*. ACM, 2011, pp. 976–979.
- [20] J. Flinn and M. Satyanarayanan, “Powerscope: A tool for profiling the energy usage of mobile applications,” in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA’99. Second IEEE Workshop on*. IEEE, 1999, pp. 2–10.
- [21] M. Goraczko, A. Kansal, J. Liu, and F. Zhao, “JouleMeter: Computational energy measurement and optimization,” 2011.
- [22] J. Reich, M. Goraczko, A. Kansal, and J. Padhye, “Sleepless in seattle no longer.” in *USENIX Annual Technical Conference*, 2010.
- [23] G. Pinto, F. Castor, and Y. D. Liu, “Mining questions about software energy consumption,” in *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 2014, pp. 22–31.
- [24] I. Manotas, C. Sahin, J. Clause, L. Pollock, and K. Winbladh, “Investigating the impacts of web servers on web application energy usage,” in *Green and*

- Sustainable Software (GREENS), 2013 2nd International Workshop on. IEEE, 2013, pp. 16–23.
- [25] D. Hammes, H. Medero, and H. Mitchell, “Comparison of nosql and sql databases in the cloud,” Proceedings of the Southern Association for Information Systems (SAIS), Macon, GA, pp. 21–22, 2014.
- [26] C. Bunse, H. Höpfner, S. Roychoudhury, and E. Mansour, “Choosing the” best” sorting algorithm for optimal energy consumption.” in ICSoft (2), 2009, pp. 199–206.
- [27] S. Arunagiri, V. J. Jordan, P. J. Teller, J. C. Deroba, D. R. Shires, S. J. Park, and L. H. Nguyen, “Stereo matching: Performance study of two global algorithms,” in SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, 2011, pp. 80 211Z–80 211Z.
- [28] C. Sahin, F. Cayci, I. L. M. Gutiérrez, J. Clause, F. Kiamilev, L. Pollock, and K. Winbladh, “Initial explorations on design pattern energy usage,” in Green and Sustainable Software (GREENS), 2012 First International Workshop on. IEEE, 2012, pp. 55–61.
- [29] C. Sahin, L. Pollock, and J. Clause, “How do code refactorings affect energy usage?” in Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, 2014, p. 36.
- [30] G. Procaccianti, P. Lago, W. Diesveld et al., “Energy efficiency of orm approaches: an empirical evaluation,” 2016.
- [31] H. Kim, J. Smith, and K. G. Shin, “Detecting energy-greedy anomalies and mobile malware variants,” in Proceedings of the 6th international conference on Mobile systems, applications, and services. ACM, 2008, pp. 239–252.

- [32] A. Merlo, M. Migliardi, and P. Fontanelli, “Measuring and estimating power consumption in android to support energy-based intrusion detection,” *Journal of Computer Security*, vol. 23, no. 5, pp. 611–637, 2015.
- [33] J. Hoffmann, S. Neumann, and T. Holz, “Mobile malware detection based on energy fingerprints—a dead end?” in *Research in Attacks, Intrusions, and Defenses*. Springer, 2013, pp. 348–368.
- [34] A. Merlo, M. Migliardi, and P. Fontanelli, “On energy-based profiling of malware in android,” in *High Performance Computing Simulation (HPCS), 2014 International Conference on*. IEEE, 2014, pp. 535–542.
- [35] F. Palmieri, S. Ricciardi, U. Fiore, M. Ficco, and A. Castiglione, “Energy-oriented denial of service attacks: an emerging menace for large cloud infrastructures,” *The Journal of Supercomputing*, vol. 71, no. 5, pp. 1620–1641, 2015.
- [36] R. Aghi, S. Mehta, R. Chauhan, S. Chaudhary, and N. Bohra, “A comprehensive comparison of sql and mongodb databases,” *International Journal of Scientific and Research Publications*, vol. 5, no. 2, 2015.
- [37] I. Konstantinou, E. Angelou, C. Boumpouka, D. Tsoumakos, and N. Koziris, “On the elasticity of nosql databases over cloud management platforms,” in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 2385–2388.
- [38] T. Dory, B. Mejías, P. Roy, and N.-L. Tran, “Measuring elasticity for cloud databases,” in *Proceedings of the The Second International Conference on Cloud Computing, GRIDs, and Virtualization*. Citeseer, 2011.