An Intelligent Energy Aware Approach for Big Data Storage in Cloud Data Centers

A Thesis

submitted in partial fulfillment of the requirements for the award of the degree of

Doctor of Philosophy

by

Sumedha Arora
(Reg no: 901603020)

under the guidance of

Dr. Anju Bala
Associate Professor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala - 147004, INDIA

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala - 147004, INDIA

December 2020

.........dedicated to all young generation researchers

Candidate Declaration

I hereby certify that the work, which is being presented in the thesis, entitled **An Intelligent Energy Aware Approach for Big Data Storage in Cloud Data Centers**, in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** and submitted to the institution is an authentic record of my own work carried out during the period **June 2016** to **December 2020** under the supervision of **Dr. Anju Bala**. I have also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

The matter presented in this thesis has not been submitted elsewhere for the award of any other degree or diploma from any institution.

Sumedha Arora

(Sumedha Arora)
Regn. No. 901603020

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Anju Bala

(Dr.Anju Bala)
Associate Professor
Department of Computer Science & Engineering
Thapar Institute of Engineering and Technology, Patiala, 147004
Punjab, INDIA

Acknowledgements

The thesis represents not only the work at the keyboard, rather it is almost four years of my research at Thapar Institute of Engineering and Technology at computer science department. It has provided me a platform to groom my skills. The presented work is only possible with a close association of many people. I extend my sincere gratitude and appreciation to all of them.

Firstly, I would like to thanks my God, my lord Krishna, I have experienced his presence and support through out my journey. The blessings of almighty has made my journey smooth and easy.

I would express my heartiest gratitude to my PhD supervisor, Dr Anju Bala for her constant support during Ph.D study. I am very thankful for her motivation and patience. At every point I have gained immense knowledge from her. Besides being my supervisor, at a times she has been my mother by helping me in taking daily life decisions. Her morning calls to get the update of the work had became my morning alarms. Her availability and dedication towards my work has made my journey easy and smooth. I could not have imagined a better advisor and mentor for my Ph.D study.

I would also thanks to the rest of my thesis committee: Dr Rinkle Rani, Dr Ashutosh Mishra, Dr MD Singh for their insightful comments and encouragement. Also, I would like to thanks Dr PS Rana for imparting me the knowledge on data science that has helped me to shape my research. I would like to extend my special thanks to Dr Mukesh singh for providing the knowledge and issuing me the electrical equipments required for my experiments.

I am highly thankful to Head, Computer Science and Engineering Department, Dr. Maninder Singh, for his encouragement. He has always been their to listen our requirements and provided the access to the laboratory and research facilities. I would like to express my sincere thanks to Dr. Inderveer Chana, Associate Head, Computer Science and Engineering Department for her constant support in carrying out this research. I would also like to thanks Dr. Rafat Siddique, Dean of Research and Sponsored Projects for academic support. My sincere thanks to Dr. Parkash Gopalan, Director, Thapar Institute of Engineering and Technology, for providing all the facilities and resources that have been instrumental in the creation robust research environment in the University.

I would like to thank my family: my late Grand father, Yashpal Arora who has encouraged me to start this journey of PHD. I owe my deepest gratitude towards my grandmother, Sushma Arora for her blessings and eternal support. Her infallible love has always been my greatest strength. I am greatfull to my parents and to my brothers for spiritually supporting me during my PhD. A special thanks to my sister Geeta Duggal and her family who supported me through out this journey.

I also wish to thanks my fellow labmates Ajay, Sanjhu, Sandeep, Savinder, Nitin, Bhavya, Priya, Gurlin, Shweta for the stimulating discussions and for the sleepless nights while working together. All the fun we had in the last four years has been recorded as an important part of my life. I am also very grateful to Nikhil, Shirish who all helped me in various ways during my PhD journey. Of course a big thank you to Gaurav, Aman, Sachin and Simar for helping me enormously, especially with the latex formatting. Finally to Vivek Dabra, my best friend who has always been by my side throughout this PhD, living every single minute of it, and without whom, I would not have the courage to embark on this.

I must thank the office staff including lab technicians and security guards, who were always available whenever required. I acknowledge Rakesh sir and Satinder sir who were always there to to help whenever required.

I further extend my thanks to all the hostel staff that provided the homely environment through my PhD. My sincere thanks to Warden Dr Anju Bala(also my superviosr), Mess manager Mrs Renu, and caretaker sir Mr Magher Singh. I will always cherish the warmth shown by Mrs Poonam, Anita, Dilpreet and Ritu mam. My special thanks to fellow hostelmate, Nisha for being a compassionate friend.

I whole heartily acknowledge Council of Scientific and Industrial Research, for their financial support to allow me to carry out my research.

Sumedha Arora

Abstract

The advancement in current technology has lead to the rapid rise in big data applications like E-commerce, scientific computing, healthcare etc. These applications require enormous computing capabilities such as high end infrastructure, platforms and softwares. Cloud data centers provide these facilities based on pay as you go model, yet raise several challenges which include energy efficiency, scalability, privacy, and storage etc. Among these issues, energy efficiency has turned into an upcoming challenge for executing the big data applications in cloud environment.

Energy has become a critical resource in modern computing systems, which presents challenges to the traditional storage systems. The energy consumed by the storage subsystem surpasses all other sub-components present in the server. The disks in high-end servers are responsible for the high power consumption. Hence, the prediction based energy-aware approach is required for an efficient data placement among the disks to power it down for the long duration. Prediction also helps in identifying which data objects need to be replicated. Therefore, an integration of data prediction with placement along with the disk scheduling provides an optimal solution to reduce energy and time consumption.

To achieve the set objectives, an extensive literature survey of existing data prediction models and energy efficient storage techniques has been done. But the previous research does not cover all the aspects such as data prediction, data placement including replica management and disk scheduling for big data storage. Therefore, in this work, an intelligent energy aware approach is proposed to reduce storage energy consumption in the cloud environment.

Firstly, the storage prediction model has been proposed that generates and customizes the SQL traces to find the frequency of each query fired on the real data streams obtained

from the SCATS sensors of Dublin city. Based on the calculated frequency, the future frequency of each query has been predicted using ensemble approach. The predicted results have been tagged and classified as popular and unpopular data based on threshold frequency. The experimental results are validated in terms of accuracy, recall, precision, error rate, F-score. It yields 87.5% accuracy and successfully reduces the error rate to 11%. The highest measure of precision possible with the proposed model is 89% with 87% recall. The ROC value of 0.93 reveals the best capability of the proposed storage prediction model.

Next, an intelligent energy aware approach has been proposed that optimally utilizes the prediction results to place the predicted popular data in hot disks using replication. Hot disks are the set of disks that remains active for most of the time. While, unpopular data is allocated to the cold disks that usually remain in standby state. When the user inputs the request, an intelligent disk scheduling technique has been applied for searching and selecting the most available disk that would execute the request. The replication allows the scheduler to select the disk that would execute the request with minimum energy and time. The disk selection is based on the maximum remaining time to move to idle state and minimum waiting time for the disk in active state. Likewise, the identified disk would save maximum disk spins. The standby disk would not be given any request until it can be satisfied by idle and active disk. The reduced energy and seek time has been measured in real world environment using multimeter and clampmeter which shows 9.7% decrease in the total execution time.

Finally, an entire intelligent energy-aware approach has also been categorically validated in a cloud environment. The performance has been evaluated on OLTP (e.g., e-commerce) applications benchmarked with financial and websearch input-output traces. Based on the performed experiments, the optimized replica, best disk ratio for each application is selected that consumes least energy and time to execute the request. The experimental results are compared with the reference benchmarks and existing literature. The proposed approach outperforms the existing approach with the 6.8% improvement in accuracy yielded by storage prediction model. Also, the 6% reduction in the energy consumption is seen along with the 18.26% improvement in the total execution time using intelligent energy aware approach.

List of Publications

International Journal

1. **Sumedha Arora**, and Anju Bala, "A survey: ICT enabled energy efficiency techniques for big data applications.", *Cluster Computing*, Springer, 23:775-796, 2019. [**SCI Indexed, Impact Factor - 3.458**]

2. **Sumedha Arora**, Anju Bala "An Ensembled Data Frequency Prediction Based Framework For Fast Processing Using Hybrid Cache Optimization.", *Journal of Ambient Intelligence and Humanized Computing*, Springer Publications, 2020. [**SCI Indexed, Impact Factor - 4.594**]

3. **Sumedha Arora**, and Anju Bala, "PAP: power aware prediction based framework to reduce disk energy consumption.", *Cluster Computing*, Springer Publications, 1-18, 2020. [**SCI Indexed, Impact Factor - 3.458**]

International Conference

1. **Sumedha Arora** and Anju Bala, "Energy-Aware Disk Storage System for Cloud Data Centers", *International Conference on Innovations in Information and Communication Technologies (ICI2CT 2020)*, 2020[Status-Accepted].

Communicated

1. **Sumedha Arora**, and Anju Bala, "An Intelligent Energy Efficient Storage System for Cloud Based Big Data Applications", *simulation modelling practice and theory journal* , elsevier, 2020 [**SCI Indexed, Impact Factor - 3.58**][**Status-Under Revision**]

Table of Contests

Title **Page No.**

Abstract . . . . **v**

List of Publications . . . . **viii**

Table of Contests . . . . **ix**

List of Figures . . . . **xiii**

List of Tables . . . . **xv**

List of Notations . . . . **xvii**

List of Abbreviations . . . . **xx**

Chapter 1 Introduction **1**

1.1 Big Data: An Overview 2

1.1.1 Big Data Applications: Categories 3

1.1.2 Big Data Applications: Emerging Cloud Technology 4

1.1.3 Big Data Applications: Cloud Data Centers 5

1.1.4 Big Data Applications: Challenges to Cloud Data Centers 7

1.2 Energy Efficiency Challenges for Cloud Data Centers 9

1.2.1 Energy-Efficiency Techniques 11

1.2.2 Storage Energy Efficiency Techniques 14

1.3 Research Motivation 16

1.4 Thesis Contributions 17

1.5 Thesis Oraganisation 18

Chapter 2 Literature Survey **22**
2.1 Cloud Data Center: Big Data Challenges 23
2.2 Energy Efficiency Techniques 24
2.2.1 Energy Efficiency Techniques: Research Challenges 30
2.3 Storage Energy Efficiency Techniques 32
2.3.1 Data Prediction Approaches 32
2.3.2 Data Placement Approaches 36
2.3.3 Disk Scheduling Approaches 37
2.4 Gap Analysis 43
2.5 Problem Statement 44
2.6 Conclusion 45

Chapter 3 Data Frequency Based Storage Prediction Model **46**
3.1 Objectives of the Proposed Storage Prediction Model 47
3.1.1 Key Traits of Storage Prediction Model 48
3.2 Methodology 48
3.2.1 Trace Generation 49
3.2.2 Machine Learning Models for Data Frequency Prediction 52
3.2.3 Data Tagging Using Predicted Threshold Frequency 55
3.3 Evaluation Metrics: Best Prediction Model 56
3.4 Experimental Results and Analysis 59
3.4.1 Dataset Description: City's Scat Data 60
3.4.2 Trace Generation 60
3.4.3 Data Preprocessing 61
3.4.4 Validation and Selection of Best Prediction Model 61
3.4.5 Performance Validation: Selected Ensembled model 63
3.5 Comparative Analysis: Proposed Model With Existing Models 65
3.6 Conclusion 66

Chapter 4 An Intelligent Energy Aware Approach Using Storage Prediction Model . . . 67
4.1 Use of Prediction in Intelligent Energy Aware Approach . . . 68
4.1.1 Objectives of an Intelligent Energy Aware Approach . . . 68
4.2 An Intelligent Energy Aware Storage Framework . . . 70
4.2.1 Data Placement Approach Using Replica Optimization . . . 70
4.2.2 Disk Scheduling . . . 72
4.3 Mathematical Formulation for Measuring Reduced Energy Consumption 74
4.4 Evaluation Metrics: Intelligent Energy Aware Approach . . . 76
4.5 Experimental Results and Analysis . . . 79
4.5.1 Validation of Proposed Approach: Data Placement Technique . . 80
4.5.2 Validation of Proposed Approach: Disk Idle State Monitoring . . 82
4.5.3 Validation of Proposed Approach: Energy Reduction . . . 83
4.6 Comparative Analysis: Proposed Approach With Existing Approaches . . 86
4.6.1 Comparative Analysis: Data Placement Approach . . . 86
4.6.2 Comparative Analysis: Energy Reduction . . . 87
4.7 Conclusion . . . 88

Chapter 5 An Intelligent Energy Aware Approach for Cloud Data Centers: A Case Study on OLTP Applications . . . 90
5.1 OnLine Transaction Processing - Case Study as Big Data Applications . 91
5.1.1 Traces of OLTP Applications . . . 92
5.2 Experimental Results and Analysis . . . 92
5.2.1 Results: Data Frequency Based Storage Prediction Model . . . 94
5.2.2 Results: An Intelligent Energy Aware Approach . . . 96
5.2.3 Comparative Analysis: Data Replication . . . 101
5.3 Conclusion . . . 102

Chapter 6 Conclusion and Future Scope . . . 104

6.1 Conclusion 105

6.2 Future Scope 108

References **109**

List of Figures

1.1 Characteristics of Big Data . . . 2
1.2 Emerging cloud technology for big data applications . . . 5
1.3 Taxonomy of energy efficiency techniques for data centers . . . 12
1.4 Generalised solution for storage energy efficiency challenge . . . 15

2.1 Systematic View of storage energy efficiency challenge and solution . . . 32
2.2 Prediction modeling and process . . . 35

3.1 Methodology used in proposed storage prediction approach . . . 49
3.2 Synthetic workload generation procedure . . . 52
3.3 Characteristics of trace generated . . . 60
3.4 Important feature selection using RRF. . . . 62
3.5 ML model selection using performance parameters . . . 63
3.6 Validation for selected ensembled model: (a) Repeated k-fold cross validation for accuracy, average class error, overall error, precision, recall, F-measure, (b) RoC curve for ensembled model . . . 64
3.7 Performance comparisons for machine learning models with existing techniques and datasets . . . 66

4.1 Intelligent energy aware framework . . . 69
4.2 Experimental testbed. . . . 79
4.3 Data placement validations . . . 81
4.4 Disk idle state analysis using analyser monitor . . . 82
4.5 Impact of data replication . . . 83

4.6 Impact of request size 84
4.7 Impact of γ 85
4.8 Data placement comparisons with existing literature 87
4.9 Energy and time comparisons with the existing techniques 88

5.1 Energy aware approach in cloud environment 91
5.2 ML model selection using performance parameters: 95
5.3 Validation for selected ensembled model 96
5.4 Energy efficient data allocation 97
5.5 Impact of data replication factor 99
5.6 Impact of request size 100
5.7 Impact of γ 101
5.8 Energy and time comparisons for data replication with existing literature 102

List of Tables

1.1 Comparisons for data centers 6

2.1 Challenges of big data applications 24

2.2 Existing energy efficiency techniques 25

2.2 Existing energy efficiency techniques 26

2.2 Existing energy efficiency techniques 27

2.2 Existing energy efficiency techniques 28

2.3 Operational issues for energy efficient techniques 30

2.4 Existing prediction techniques on various workload traces 33

2.4 Existing prediction techniques applied on various workload traces 34

2.5 Existing energy efficient prediction placement, scheduling techniques for storage energy efficiency challenge 39

2.5 Existing energy efficient prediction, placement and scheduling techniques for storage energy efficiency challenge 40

2.5 Existing energy efficient prediction, placement and scheduling techniques for storage energy efficiency challenge 41

2.6 Energy efficiency parameters covered in literature 42

3.1 Basic parameters of the trace 50

3.2 Illustration of features of trace 53

3.3 Model selection criteria considering various performance metrics 64

4.1 Disks configuration 80

4.2 Disk states and power consumption 80

5.1 Characteristics of OLTP traces 93
5.2 Disk state and power consumption 93

List of Notations

β_i	start of the queries
γ_i	start and end times of the queries
I_i	idle time
TI_i	Idle time threshold
$freq(i)$	frequency
$pfreq(i)$	predicted frequency
Y	dependent variable
X	independent variable
$E(Y)$	expected value
g	link function
A_t	threshold for predicted frequency
q_i	input query
Q_h	popular query
Q_c	unpopular queries
$\hat{y}$	predicted value
y_i	corresponding value
N_{fp}	number of false prediction
P_r	prediction for a particular row
w_p	wrong predictions
D_h	hot disk
D_c	cold disk
D_cap	required storage capacity
t	total number of unique files
s_i	size of ith query in[GB]
rep_i	replication factor

γ	ratio of the hot and cold disk
P	P are the total number of assigned hot disks
T	T are the total number of assigned cold disks
R	total replication factor
rep_h	replication factor for hot file
rep_c	cold replicas
l	log table
D_j	searched disk
D_k	selected disk
RI_i	maximum remaining idle time
W_t	minimum waiting time
r_i	current request
E_{alloc}	allocating
E_{Search}	energy consumed during searching and selecting E_{Select}
E_{Select}	energy consumed during selecting
P_{active}	power consumed by the disk in active state.
P_{idle}	power consumed by the disk in idle state .
$P_{standby}$	power consumed by the disk in standby state .
t_m	duration for the disk in active state
t_j	duration for the disk in idle state
t_k	duration for the disk in standby state
$n1^{'}$	number of times an active state is encountered
$n2^{'}$	number of times an idle state is encountered
$n3^{'}$	number of times an standby state is encountered
$N1$	number of times the disk is spin from idle to standby
$N2$	number of times the disk is spin from standby to active
tbr_i	burst time of the current request
ta_l	arrival time of current request

te_{l-1}	depicts the end time of previous request
P_{is}	spinning power from idle state to standby state
t_{jk}	time consumed from idle state to standby state
P_{sa}	spinning power from standby state to active state
t_{ki}	time consumed in spinning from standby state to active state
t_{exe}	execution time taken to accomplish requests submitted
t_{wt}	waiting time
$t_{ActiveEndT}$	time when disks are executing the requests
P_{exe}	power

List of Abbreviations

MRI	Magnetic Resonance Imaging
NASA	National Aeronautics and Space Administration
ISRO	Indian Space Research Organisation
ESA	European Space Agency
OLTP	Online Transaction Processing
IO	Input-Output
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
EC2	Elastic Compute Cloud
S3	Simple Storage Service
EMR	Elastic Map Reduce
kWh	kilowatt-hours
SLA	Service Level Agrement
PUE	Power Usage Effectiveness
SPM	Static Power Management
DPM	Dynamic Power Management
GDC	Green Data Centers
GPU-CPU	Graphics Processing Unit-Central Processing Unit
DSC	Dynamic Smart Cooling
RAID	Redundant Array of Independent Disks
SAN	Storage Area Network
NAS	Network Attached Storage
HDFS	Hadoop Distributed File System
DCN	Data Center Network

BIM	Building Information Modeling
WSN	Wireless Sensor Networks
HVAC	Heating, Ventilation, and Air Conditioning
EDOM	energy efficiency of database operations on multicore servers
NNP	Neighbourhood Partitioning
PS	Performance Directed Static
TPES	Three-Phase Energy-Saving Strategy
NN	Neural Networks
SVM	Support Vector Machines
ARIMA	Autoregressive Integrated Moving Average techniques
RMSE	Root Mean Square Error
ANN	Artificial Neural Network
CBR	Case-Based Reasoning
PAM	Power-Aware Multi-Level
SSD	Solid State Drive
RIMAC	Redundancy-based Cache Architecture called
SRCMap	Sample-Replicate Consolidate Mapping
MCS-SSD	Multi-Level Caching Strategy
PEARL	Performance, Energy, and Reliability balanced
EERAID	Energy-Efficient RAID system architecture
WRR	Windows Round-Robin
PRF	Power and Redundancy-Aware Flush
SEA	Striping Based Energy Aware
STAAS	Storage as Service
HDD	Hard Disk Drive
MCV	Multi Class Voting
LM	Linear Model
SCATS	Sydney Coordinated Adaptive Traffic System

TP	True Positive
TN	True Negative
FN	False Negative
FP	False Positive
PSF	Pattern Sequence-based Forecasting
DL	Deep Learning
GPR	Gaussian Process Regression
DPRP	Distance-based Predicted Region Policy
PMCR	Predictable Markov based Cache Replacement
BlEX	Block Exchange
RESS	Reliable Energy-Efficient Storage System
PLFS	Modified Parallel Log-Structured File System
LBA	Logical Block Address
ASU	Application-Specific Unit
SPC	Storage Performance Council
WCS	Weighted Set Cover
MWIS	Maximum Weighted Independent Set

Fellowship Awarded

Awarded Senior Research Fellowship (SRF) by Council of Scientific and Industrial Research (CSIR), Government of India, under ACK.NO: 143253/2K17/1 and File No: 09/677(0030)/2018-EMR-I from May 2018 to May 2020. This fellowship is awarded to pursue Doctorate of Philosophy in Sciences and Engineering. It helped me in funding the required resources to carry out this research work which leads to the successful completion of this thesis.

Chapter 1

Introduction

The growth and development of the information and communication technology industry has led to a rapid rise in big data applications. The development of cloud data centers serves as an appropriate platform for delivering services to these applications. The applications hosted in cloud data centers brings with them various challenges like energy efficiency, scalability, heterogeneity and storage etc. Also, the E-commerce applications targeting real time transactions using SQL queries demands large quantity of servers equipped with disks that consume high energy. So, among all the other challenges, storage energy efficiency has become a major concern.

This chapter exhibits an overall glimpse of the thesis and covers all aspects of energy efficiency techniques for big data applications hosted in the cloud data centers. It explores cloud platforms for big data applications that extends to cloud data centers. Subsequently, the energy efficiency challenge and its impact on the environment have been highlighted. Besides, various energy efficiency techniques for big data have been reviewed. Among all the approaches, the storage energy-efficient techniques has been identified to be in infancy that requires attention. Therefore, this chapter reviews some generalised techniques to resolve storage energy efficiency challenge. Finally, the chapter has been concluded with appropriate discussions on thesis contribution and organization.

1.1 Big Data: An Overview

Big data refers to datasets with sizes beyond the capability of the traditional database software tools that are used to capture, manage, store, and analyze information. The term denotes voluminous data that is collected from sensors or online portals that gather climate information, flight details, traffic updates from GPS signals, posts at social media sites, digital pictures, videos with transaction records, etc. As shown in Figure 1.1, the data is characterized by the 5 V's, namely (volume, velocity, variety, and veracity and value).

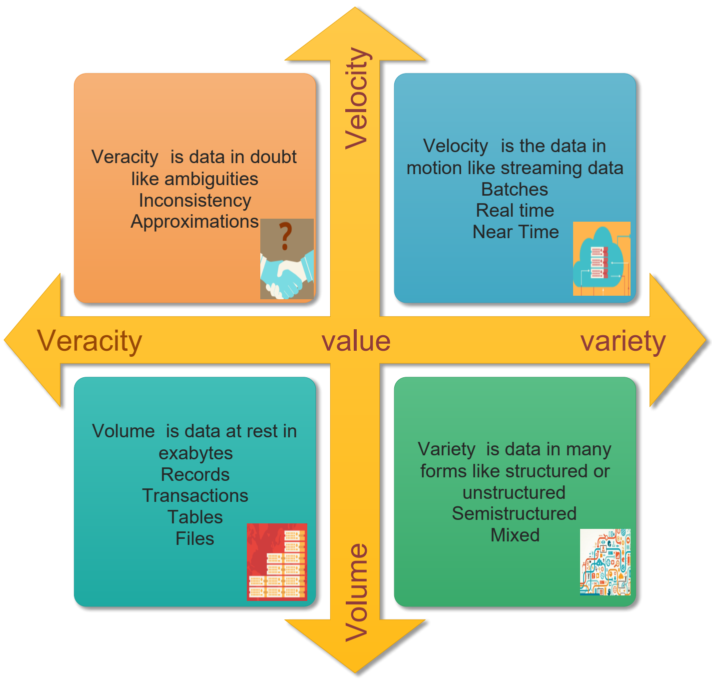


Figure 1.1: Characteristics of Big Data

Volume refers to the enormous datasets that keep on increasing in size starting from gigabytes (2^{30} byte) and reaching terabyte(2^{40} byte), and petabyte (2^{50} byte) levels; sometimes even crossing exabytes (2^{60} byte) and zettabytes (2^{70} byte). This vast amount of data is migrated with a high speed which is termed as the velocity of data. Variety focuses on the complexity of the big data. Complex computing is required to extract valuable information from a large dataset which is known as value. Veracity of the data refers to the ambiguity and inconstancy that needs to be analysed using tools such as Hadoop, BigTable, etc. [1].

1.1.1 Big Data Applications: Categories

Big data is involved in various fields such as health care [2], criminal justice, space science, and other domains that provide powerful outcomes. These disciplines are termed as big data applications. These applications can be broadly categorized into three fields, i.e. commercial, scientific and network. The categorization has been done on the basis of the sources that consist of remotely-controlled sensors and actuators. Besides, the classification of the applications depends upon whether the data is structured, semi structured or steaming. They differ in the characteristics, processing needs, analytical tools and technologies required to analyse them. The major categories of such applications have been described below:

- Scientific Applications

  Scientific applications include astrophysics, oceanology, bioinformatics and healthcare. These domains generate a massive amount of experimental, reference and observational data from high resolution sensors. For instance, the healthcare data is generated by scanning technologies such as Magnetic Resonance Imaging (MRI) [3]. The data for astronomical applications is collected by various space agencies such as National Aeronautics and Space Administration (NASA), Indian Space Research Organisation (ISRO), European Space Agency (ESA), etc. This unstructured or semi-structured data is stored in NoSQL databases such as Cassandra or MongoDB and processed using big data tools such as Hadoop or Spark.

- Commercial Applications

  Commercial applications involve day-to-day commercial activities. These applications include E-commerce, precise farming and habitat monitoring. E-commerce applications targets a specific class of systems, called Online Transaction Processing

(OLTP). The data is mostly structured in the format consisting of reports which are stored in databases. Such applications exercises large file management, concurrency and random writes/reads. For example, the banking sector generates transactional data which are then stored using mysql, oracle or SQLite.

- Network Applications

  In network applications, the web client programs such as Internet Explorer or Firefox, are made to run on the client computer whereas the server runs a web server program such as Apache or internet information Server. These applications involve data regarding social media, traffic monitoring and crime management. These applications demand strong and secure network connections. The data is never in a structured form and would require Flume, Kafka, Sqoop or Hive for reading, writing and interacting [4]. For example, the data for military applications would include the commands and control messages. Social media applications would be in the form of e-mails, word processing, web browsing, etc. In case of smart city applications, the data would comprise traffic patterns, routes to the cities, or traffic information including speed and location, which is processed using graph algorithms [1].

Traditional data processing techniques have failed to meet the high real-time demand for big data applications specifically for high Input-Output (IO) applications such as e-commerce that produce intense storage access [5]. Cloud platform helps in dealing with the big data outbreak by providing high computing facilities[6].

1.1.2 Big Data Applications: Emerging Cloud Technology

It is a platform that offers the ability to pay only for those resources that are needed [7]. It provides applications as service over internet which is termed as Software as a Service(SaaS). When it acts as a platform for big data applications, it is termed as Platform as a Service(PaaS). Services provided by cloud also include Infrastructure as a

Service(IaaS) which hosts all the applications giving rise to cloud data centers as shown in Figure 1.2.

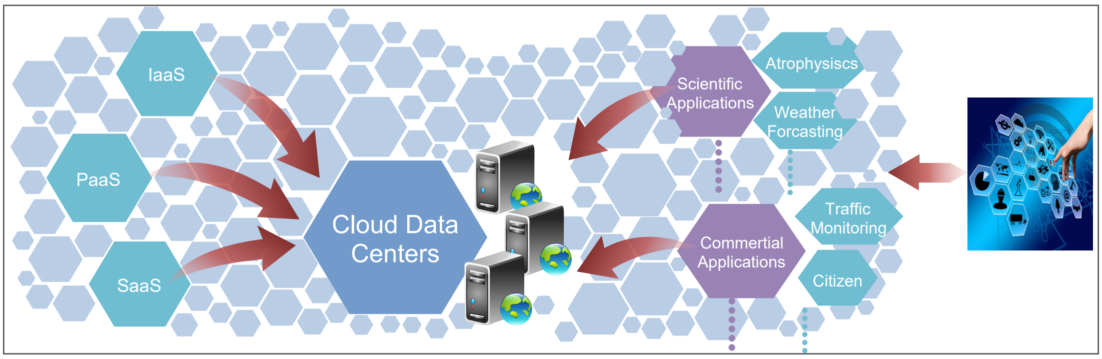


Figure 1.2: Emerging cloud technology for big data applications

1.1.3 Big Data Applications: Cloud Data Centers

Data centers provide facilities for housing the computer systems(servers) and associated components [8]. Such a physical environment also ensures power supplies with backup, redundant communication cabling systems including air conditioning, fire suppression and physical security devices for staff entrances. High computing servers in the data centers help in providing infrastructure as a service for analysing big data. These servers offer high computation capabilities by supporting the virtualization technology. Besides, the servers provides big data with high performance computing and storage resources. Instead of utilizing a local computer, one can directly make use of the distributed storage technology of cloud. Cloud data centers includes Amazon, Microsoft and Google, which provide various services that the user can avail in pay-as-you-go fashion [9]. Examples of such services provided by Amazon includes Elastic Compute Cloud (EC2), Simple Storage Service (S3) and Elastic Map Reduce (EMR.)

Table 1.1 furnishes a comparison of today's data centers from 2013 to 2019. Different data centers from all over the world are compared on the basis of their size, number of locations, energy and servers [10]. Amazon data centers are constructed in 1.5 million square feet and can thus can accommodate 1.4 million servers at a time. Amazon appears to be

Table 1.1: Comparisons for data centers

Characteristics	Data Centers			
	Amazon DC	**Facebook**	**Google**	**Microsoft**
No of Servers	1.4 million	80000	2.5 million	80000
Location	100	9	14	100
Data Size	554 exabyte PM	7 petabytes gear every month	507.5 Zettabytes	120 exabyte PM
Energy	5500 megawatt	3.43 terawatt hours	10.6 terawatt hour	7.5 billion kilowatt-hours (kWh)
Dimensions of Data Centers	1.5 million Sq feet	15 million Sq feet	30 million Sq feet	1.2 million square feet

expanding their data center at northern Virginia in 2020. According to the "Amazon Atlas", Amazon has eight centers in San Francisco, 38 in Northern Virginia, seven in Oregon and eight in Seattle. In Europe, it has 7 fully operated data centers at Dublin, Ireland, while four are located in Germany, and three are in Luxembourg. In the Asia-Pacific region it owns 12 data centers in Japan, six in Singapore, nine in China, and eight in Australia. It also occupies six sites in Brazil.

Google owns eight data center locations all over the world: four in Europe, one in South America and two in Asia [11]. Newly constructed data center of Google in Oregon is spread across 164,000-square foot building. Its energy consumption has been raised to 10.6 terawatt hours in 2018 from 2.86 terawatt hours in 2011 [12].

The Microsoft Cloud spans over 100 data centers around 54 regions and delivers service in 140 countries. The company currently owns approximately 80000 severs to process the data. Microsoft data centers covers around 6.3 million square feet of space and has its largest data center in Chicago. According to 2018 Data Factsheet: Environmental Indicators for Microsoft, the market-based emissions crosses 7.5 billion kilowatt-hours (kWh) of renewable energy attributes.

Facebook owns 80,000 servers which are distributed in nine worldwide locations. Its datacenters are operated in:Prineville, Oregon, North Carolina, Forest City, Lulea, Altoona, Sweden, Iowa and ready to be operated in Fort Worth, Texas Clone, Ireland. Facebook has also announced to expand its Eagle Mountain branch from 1 million square feet to 1.5

million square feet. In may 2018, it is the first company that has broken the ground on the campus. Its electricity consumption has been raised over the last years and reached to 3.43 terawatt hours in 2018.

The development of huge data centers lead to the various challenges that effects smooth functioning for implementing big data applications in cloud data centers. Subsequent section, enlightens some key research challenges centers.

1.1.4 Big Data Applications: Challenges to Cloud Data Centers

The servers embedded in data centers for processing big data applications cause several challenges like such as scalability, energy efficiency, data integrity, data transformation, data heterogeneity, data quality, privacy and legal issues-

- Scalability

  Scalability is the capability of a system to increase the number of resources with the growing amount of data [13]. It is a process of increasing the output of system with an increased load. The techniques that improves the scalability of system includes-

  (i) Virtualization: Virtualization is a process of sharing a resource in isolation, which gives the illusion of increased number computer resources to the user.

  (ii) Map-Reduce: Map reduce is used for performing scalable distributed operations on various applications by providing an interface for distributed and parallel computing in a cluster.

- Heterogeneity

  Heterogeneity refers to the data that is not fit for the analysis purpose. The reasons for the heterogeneity is the growth in virtually unlimited sources of data. Heterogeneous data generated is either structured or unstructured. However, in the case of unstructured data, prior to the processing of the data for analysis purpose, it is stored in distributed databases, such as HBase. Once it meets the schema-on-read constraints then only it is retrieved.

- Storage and Processing Issues

  While dealing with big data, the storage available is not enough for storing voluminous data. Hosting and uploading the terabytes of data is a time consuming process due to rapid changing of data. This results in a chaos in a distributed nature of a cloud. To handle this chaos various database systems such as big table, dynamo, cassandra and mongo have been developed that can support various big data applications.

- Data Security

  Data security provides the privacy of data [14]. It is one of the major issue which is encountered while outsourcing the confidential data into the cloud storage. Encryption and cryptography are current techniques that ensures data privacy in the cloud environment.

- Energy Efficiency

  Energy efficiency is measured with the total power consumed by computing systems in data center [15]. High energy is consumed in data processing as well as in transmission process. Therefore, power consumption has become a major concern for the development and design of modern data centers. This major challenge can be resolved using-

  (i) More efficient data transmission and compression technologies can be used before storing it, so that it takes less space which ultimately helps in achieving energy efficient storage.

  (ii) Data Caching mechanisms reduces energy and time consumption by decreasing overhead caused by disk spins.

(iii) Multicore central processing units(CPUs) can be used with combination of virtual machine consolidation to improve the energy efficiency within the data centers. Through this VMs can be allocated to a single physical node which results in improved utilization of resources hence reduces energy consumption.

There are several other issues like interoperability, data transfer, data visualization, data redundancy, which causes inconsistency in data. Data captured from many sources cause interoperability issue which requires a conversion of data into a format that is easier and uniform to be used by users. Another issue termed as data redundancy refers to reputation of data which causes increase in the response time and makes the data unreliable. Data representation issue is encountered at time of representing heterogeneous data. These days data can be represented using visualization tools such as graphs and charts. Data transfer issue causes delay in time and analysis while copying a data from one storage device to memory, which can be controlled by increasing replication among disks.

Among all the issues discussed till now, it is noticed that most of the big data applications deployed in data centers consume large amount of energy. This results in high operational cost and large amount of carbon emission in environment. It even causes Service Level Violation (SLA) violation and also degrades the performance of applications running on the systems. So it has become essential to manage energy efficiency in data centers [16]. The following section enlightens the energy efficiency challenge in detail.

1.2 Energy Efficiency Challenges for Cloud Data Centers

Recent surveys reveals the continuous rise in the infrastructure, power consumption, cost and carbon emission in the Google data center.

- Raised Infrastructure

  The burgeoning data density has led to a significant rise in cloud computing users and data centers providers. Hence, a continuous increase in the square feet has been noticed reaching 30 million square feets in 2020.

- Raised Cost

  The infrastructure consists of extra switches and servers that largely contribute to the operational costs. The cost varies with the installations and is predicted to increase in the coming years. Google LLC has announced to spend over $10 billion this year for growing its network of data centers in the U.S [17]

- Increased Power Consumption

  Big data is accomplished with heterogeneous workload from various sources. The excessive load placed on the local utilities in the data centers requires high-end servers and power-hungry disks that escalate the power consumption. Google's data centers currently use around 260 million watts of power which accounts to 0.01% of global energy consumption. Hence, there is a need to address the green issues regarding big data computing and processing.

- Environmental Concerns

  The voluminous load placed on the utilities present in the data centers increases the heat dissipation and leads to environmental concerns [18]. Energy-related CO_2 emissions account for 60% of the global emissions of the gas [19]. Nevertheless, conversion of fuel into electricity within the data center can slash 88% of the emissions at a time [20]. 2015 is the year with maximum carbon emissions. Thus, the rapidly increasing carbon emissions from servers and air conditioners needs to be monitored and reduced through computational and thermal techniques.

- Degradation in System Performance

  Applying energy-aware optimization mechanisms could, at times, affect the overall performance of a system. The performance degradation causes server throttling and SLA constraints in terms of time and cost. The survey reveals that the Power Usage Effectiveness (PUE) has dropped significantly which is an industry standard for understanding and improving the energy efficiency of the data center infrastructure systems. High PUE is required for a data center to be called as a green center. Google data centers provides 1.12 as an average PUE value. The value has remained constant since 2013, whereas it reaches to 1.09 in the year 2020 [21].

This increased negative impact of the energy efficiency challenge has convinced the experts to focus their attention on the necessary energy efficient techniques. The following section covers taxonomy of all the techniques employed till date to gain energy efficiency in cloud data centers.

1.2.1 Energy-Efficiency Techniques

Based on energy consumption analysis, the energy-aware approaches have been reviewed for maximising a data center's efficiency. As shown in Figure 1.3, the energy efficient techniques can be divided into infrastructure, storage, networking, scheduling and analytical techniques. These approach are further categorized as Static Power Management (SPM) and Dynamic Power Management (DPM) [22]. In SPM, optimization methods are applied at the time of design and cannot be modified later. However, in DPM, the optimization methods and strategies are applied at run time depending on the state of the system.

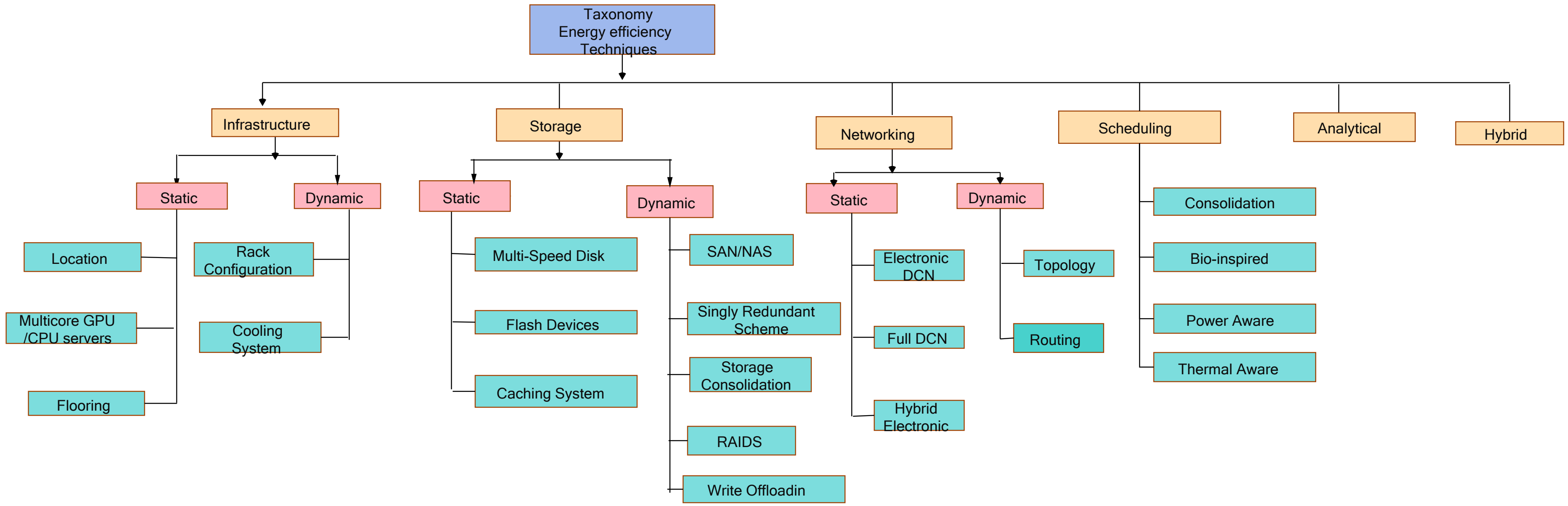


Figure 1.3: Taxonomy of energy efficiency techniques for data centers

- Infrastructure Techniques

  Infrastructure techniques are used to minimise the energy consumed in the data centers by constructing Green Data Centers (GDC). GDC consists of flooring with perforated tiles and a raised floor plenum for cool air intake. Infrastructure techniques also include the setting up of multicore servers encompassing the Graphics Processing Unit-Central Processing Unit (GPU-CPU) architecture and Dynamic Smart Cooling (DSC) system for saving energy in data centers.

- Storage Techniques

  Storage subsystem is the second largest energy consumer. The storage energy consumption can be further reduced using dynamic techniques such as storage consolidation, Redundant Array of Independent Disks (RAID), write offloading, caching policy, Storage Area Network (SAN) and Network Attached Storage (NAS)

- Analytical Techniques

  Energy efficiency through analytical techniques can be achieved with clusters of nodes, which can be either homogeneous or heterogeneous [23]. Each of these classes is further categorized into mapreduce based and Hadoop Distributed File System (HDFS)based technique.

- Networking Techniques

  Networking is considered an important technique that can improve energy efficiency in the data center [24]. These techniques deal with numerous servers which are connected through switches and routers. Static networking describes the categories of the data center based on electronic Data Center Network (DCN), full optical DCNs and hybrid electro-optical DCNs

- Scheduling Techniques

  Scheduling techniques include thermal-aware, bio-inspired, voltage-aware and consolidation-based methods. Scheduling techniques can be used with any of the above-mentioned techniques [25].

- Hybrid Techniques

  Standalone techniques do not provide the benefits that can be gained by combining various energy-efficient approaches, which are termed as hybrid techniques. Such hybrid techniques are quite effective as they provide a trade-off among energy, cost and performance.

Although these techniques help in increasing the overall energy efficiency of the system, however, they bear various limitations. Energy efficient storage technique suffers from several issues such as disk overhead, data replication, operational cost, and load balancing. Also, the storage subcomponent like disks consumes about 26 % of the overall energy[26]. Therefore, the storage techniques is still seen to be in infancy specifically for big data applications that requires analysis to understand which queries are being fired frequently such that the disk management can be performed to reduce the storage energy consumption. Hence there is a need to addresses this challenge by employing energy-efficient prediction, placement, and scheduling techniques as described in the following subsection.

1.2.2 Storage Energy Efficiency Techniques

The storage energy efficiency can be achieved through energy efficient data prediction which followed by data placement techniques. Further, an intelligent disk scheduling can be applied which includes various threshold techniques to reduce overall storage energy consumption (Refer Figure 1.4).

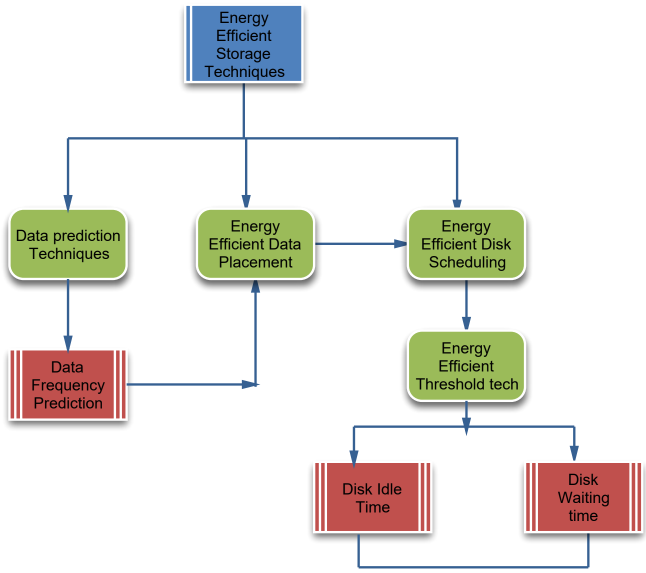


Figure 1.4: Generalised solution for storage energy efficiency challenge

- Data prediction

  Data prediction evolves predicting the data frequency that would help in identifying data usage patterns. The predicted results can be utilized in data placement approaches using spacial and temporal locality.

- Data Placement

  Energy efficient data placement evolves placing the most used predicted data in the hot disk and the remaining in the cold disk. Such placement can reduce the high spins and hence elongates the standby state of the disk.

- Disk scheduling

  Finally, energy-efficient disk scheduling is adopted where the most energy-efficient disk is searched and selected based on various threshold conditions applied on disk idle and waiting time.

This section briefed about the generalized solution for energy-efficient storage techniques that can be implemented together in cloud data centers. It motivates us to conduct the research. The following section discusses the research motivation for this thesis.

1.3 Research Motivation

Energy has become a critical resource in modern computing systems, which presents challenges to the traditional storage systems [27]. The energy consumed by the storage sub-system surpasses all other sub-components present in the server. The disks in high-end servers are responsible for the high power consumption [28]. Hence, the disk's power consumption needs to be decreased by using data prediction approaches [29] specifically for transactional oriented E-commerce applications which is the core of many data centers. Although, there are many impendence to research in energy-aware disk storage algorithms. But the previous research does not cover all aspects such as data prediction [30], data placement including replica management [31], monitoring of minute disk states, disk idle and waiting timings, etc in the cloud environment [32].

Data prediction helps in an efficient data placement among the disks to power it down for the longer duration. Prediction also helps in identifying which data objects need to be replicated [33]. Therefore, an integration of data prediction with placement along with the disk scheduling provides an optimal solution to reduce energy and time consumption. Moreover, to the best of our knowledge, very few researchers in the field of storage have implemented disk storage management in the real time environment as well as in a cloud environment [34].

Therefore, the main motive of the current research work is to reduce storage energy consumption and enhance performance by minimizing the execution time for the cloud data centers. The following section describes the pertinent contributions of the thesis.

1.4 Thesis Contributions

The pertinent contributions of the thesis are mentioned below:

- Various characteristics and challenges of big data applications in cloud data centers have been explored. Subsequently, the relationship between big data and cloud computing has been studied. The existing energy efficiency challenges pertaining to big data storage in cloud data centers have been recognized.

- Several existing storage prediction models and energy-efficient storage techniques for analyzing big data storage have been explored and analyzed.

- The data frequency based storage prediction model has been proposed using classified ensemble approach for identifying data usage patterns that predict data frequency . It further tags the predicted popular and unpopular data based on threshold. The experimental results are validated in terms of accuracy, recall, precision, error rate, F-score

- An intelligent energy-aware approach has been framed that takes predicted popular and unpopular requests as an input to distribute them in hot and cold disks respectively using data replication. It further schedules the request to the best-identified disk that can save maximum disk spins and consume the least energy and time.

- Finally, the proposed intelligent energy efficient framework is validated in real world environment using data obtained from the SCATS sensors of Dublin city, taken as a commercial application. The proposed approach is further simulated in cloud environment on OLTP applications benchmarked with financial and websearch traces. Besides, the best replica and disk ratio for each application has been recognised based on execution time, energy savings.

1.5 Thesis Oraganisation

The introduction to this thesis is presented in Chapter 1 and the rest is structured as follows:

Chapter 2 Literature Survey:

Chapter 2 presents a detailed literature on big data and its challenges. It focuses on energy efficiency challenge within the cloud data centers. After exploring the existing energy efficient approaches, it displays research issues specifically in energy efficient storage techniques. It is followed by the deep survey of the solutions to storage energy efficiency techniques such as data prediction, placement and disk scheduling. It surveys the existing prediction approaches to predict data usage. Besides, the existing energy-efficient data placement techniques are reviewed. Additionally, an extensive survey of the existing disk scheduling that includes various threshold techniques are well explored. The chapter also presents the platforms and tools used by various researchers to bring energy efficiency within the data centers. Based on the gaps prevailing in the existing literature, the problem formulation and objectives of this thesis are sketched. This chapter is derived from:

- **Sumedha Arora**, and Anju Bala, "A survey: ICT enabled energy efficiency techniques for big data applications." Cluster Computing, 23:775-796, 2019. [**SCI Indexed, Impact Factor-3.458**]

Chapter 3 Data Frequency Based Storage Prediction Model

Chapter 3 describes the proposed storage prediction model that uses a classified ensembled approach for predicting the data frequency of each query fired on the real data streams obtained from the SCATS sensor of a city. The methodology includes the trace generation procedure. The key characteristics of trace and the strategies used in preprocessing are well detailed. Further, the mathematical aspect of machine learning prediction models and proposed ensembled models are well highlighted. Next, the threshold criteria to segregate the predicted popular and unpopular queries are lighted. It is followed by an overview of the evaluation metrics that measure the performance of the proposed model. The performance of the proposed approach is validated based on the accuracy, error rates, precision, recall, F-score, ROC. The results clearly states the superiority of the proposed approached by improving the overall accuracy by 6% with an error reduction up to 11%. Further, the predicted result is used in an intelligent energy aware approach as described in the following chapter. This chapter is partially derived from:

- **Sumedha Arora**, and Anju Bala, "An Ensembled Data Frequency Prediction Based Framework For Fast Processing Using Hybrid Cache Optimization." Journal of Ambient Intelligence and Humanized Computing. [**SCI Indexed, Impact Factor-4.594**][Status-Accepted].

Chapter 4 An Intelligent Energy Aware Approach Using Storage Prediction Model:

Chapter 4 details an intelligent energy aware approach that distributes the predicted popular data in the hot disk with the replication. The remaining data is stored in the cold disk. It states the mathematical function for the data placement approach. When the user inputs the request, an intelligent scheduling technique is applied for searching and selecting the most available disk that can execute the request. The replication allows the scheduler to select the disk that would execute the request with minimum energy and time. The disk selection is based on the maximum remaining idle time and minimum waiting time for the disk in active and idle state respectively. Likewise, the identified disk would save maximum disk spins. The standby disk would not be given any request until it can be satisfied by idle and active disk. It also unveils the mathematical formulation for measuring the reduced energy consumption by the disks. The proposed framework is validated against data placement approach, disk idle time monitoring, energy consumption, and execution time in the actual real world environment using clampmeter and multimeter. Energy and response time has been measured with respect to query size, replication number, disk ratio, etc.

It would save the spinning power by letting the disk to stay in an idle mode whenever it has a request in the near future, elongates the standby duration of the disk. Hence, 87.5% accuracy in data frequency prediction leads to 9.7% decrease in the total execution time.

This chapter acquired the content from:

- **Sumedha Arora**, and Anju Bala, "PAP: power aware prediction based framework to reduce disk energy consumption." Cluster Computing, 1-18, 2020. [**SCI Indexed, Impact Factor-3.45**][**Status-Accepted**]

Chapter 5 An Intelligent Energy Aware Approach for Cloud Data Centers: A Case Study on OLTP Applications:

An intelligent energy aware approach has been categorically validated on OLTP (e.g., e-commerce) applications benchmarked with financial and websearch I/O trace, in cloud environment. Among all the features,the ASU is selected whose frequency has been calculated, predicted and tagged the popular and unpopular files using storage prediction model. The popular predicted data is prefetched into the hot disk with the replication to reduce the disk spins. Subsequently, scheduling is performed for all the incoming requests. It selects the best disk to execute the request such that it selects either idle or active disk without spinning standby disk. The selection procedure considers minimum waiting time for active disk and maximum remaining idle time for idle. The proposed system has been implemented by adding disk management in the cloud environment using CloudDiskSim. The experimental section in this chapter is segregated into two parts. The first part presents the prediction results by storage prediction model in terms of accuracy, errors, ROC, F-measure. It is followed by energy consumption and execution time results by intelligent energy aware approach that proves the efficacy of our work. The content of this chapter is derived from:

- **Sumedha Arora**, and Anju Bala, "Energy-Aware Disk Storage System for Cloud Data Centers" International Conference on Innovations in Information and Communication Technologies (ICI2CT 2020) [Status-Accepted].

- **Sumedha Arora**, and Anju Bala, "An Intelligent Energy Efficient Storage System for Cloud Based Big Data Applications." simulation modelling practice and theory journal elsevier.[**SCI Indexed, Impact Factor-3.58**][Status-Communicated].

Chapter 6 Conclusion and Future Directions:

This chapter summarizes the conclusions drawn from the thesis along with the possible future directions.

Chapter 2

Literature Survey

Big data applications hosted in cloud data centers exhibits various challenges such as privacy, storage interpretability, energy efficiency, heterogeneity, visualization, and scalability etc. Among them, the energy efficiency challenge has been recognized as the prime concern. Several existing energy efficiency techniques can be categories as infrastructure, storage, analytical, networking, scheduling, and hybrid. An impendence to research in energy-aware storage techniques are recognised which needs to be resolved.

Storage techniques presents several open issues like disk overhead, replication and disk high power consumption that demand efforts for achieving the full potential of the technology. These operational issues can be removed by following the predictable data trends that can be used in data placement to schedule the request to the disk which would consume minimum energy.

This chapter explores big data challenges and investigates the taxonomy on energy efficiency techniques. It recognises the operational issues for storage energy efficiency techniques. So, to improve the storage energy efficiency techniques, various prevailing energy-efficient data prediction, data placement, and disk scheduling techniques are reviewed. This chapter also reveals various tools and platforms used by various authors to implement storage energy-efficient techniques. Further, the limitations pointed out in the existing research help us to bridge the research gaps. Based on these gaps, the problem formulation and objectives of this thesis are outlined.

2.1 Cloud Data Center: Big Data Challenges

Technological advancements have led to an exponential growth in big data. The Authors in the field have worked on the topics ranging from the energy effect on hosting big data applications to processing it in data centers. Fahim et al. [4] investigated the whole life cycle of the big data system including data acquisition, cleansing, wrangling, transformation, loading, storage and processing. Jinsong et al. [35] described analytical tools for the processing of big data applications. Various authors have explored big data processing and its challenges. After acquisition, the data is transferred to cloud data centers [36]. Hence, there is a need to identify the most energy-efficient path for the application to reach the cloud data centers. Baker et al. [37] provided an energy-efficient routing solution along with the meta-director framework that shows the shortest path to the cloud data centers. The data is then processed within the centers that exhibits various challenges like scalability, data integrity, data transformation, data quality, data heterogeneity, energy efficiency, legal issues and privacy that effects smooth functioning for implementing big data applications. Authors in [13] has explored virtualization and mapreduce to address the scalability issue. Moghram [38] introduced check point method which increases level of fault tolerance in the system. Authors in [39] exposed various database systems such as big table, dynamo, cassandra and mongo DB to efficiently store the data in cloud data centers. Large amount of work has also covers the data security challenge by highlighting encryption and decrytion by various authors [14]. Several efforts have been done to reduce energy efficiency using more efficient data transmission and compression technologies [40].

Table 2.1 discusses these challenges where the cross symbol depicts the issues that are yet unresolved and check symbols indicate that some work has already been done regarding the particular problem. Energy efficiency is seen as a major challenge that requires to be solved in applications including health care, crime analysis, weather forecasting, precision agriculture, bioinformatics and E-commerce . It is also noticed that among all the applications, an E-commerce applications like OLTP evolves real-time transaction oriented applications which produce intense random storage accesses resulting in high disk spins. This results in high operational cost and large amount of carbon emission in environment, this even causes in violation of SLA constrains and sometimes degrades the performance of application running on the systems. Therefore, it has become essential to save energy for E commerce applications using energy efficiency techniques. To address this problem plethora of energy efficient techniques can be used such as energy efficient infrastructures, DCN, energy efficient storage. Following section surveys various existing energy efficiency techniques.

Table 2.1: Challenges of big data applications

Applications	Challenges								
	Privacy	Storage	Interpretability	Data Transfer	Energy efficiency	Heterogeneity	Visualisation	Scalability	Fault Tolerance
Health care [41]	✓	✓	×	✓	×	×	✓	✓	×
Crime Analysis	✓	✓	×	×	×	×	×	×	✓
Military [42]	×	×	×	×	✓	×	×	×	✓
Environmental Monitoring	×	×	×	×	✓	×	×	×	×
Habitat Monitoring	×	×	×	×	✓	×	×	×	×
Social Media [43]	×	✓	×	×	✓	×	×	×	×
Weather [44]	×	✓	×	×	×	×	✓	×	×
Astronomy [45]	×	✓	×	×	✓	✓	✓	×	×
Precise agriculture [46]	✓	✓	×	×	×	×	✓	×	×
Bioinformatics [47]	×	✓	✓	✓	×	×	×	×	×
e commerce [48]	✓	✓	✓	✓	×	×	×	✓	×

2.2 Energy Efficiency Techniques

Several Authors have worked on various energy efficiency techniques like infrastructure, storage, networking, and scheduling. Ricardo et al. [49] used power models in a linear combination for multiple applications hosted on the same server to lower the power consumed by the servers present in the data centers. However, the applications hosted in the data centers were assumed to be in a stable state, which is not possible in the real world.

Table 2.2: Existing energy efficiency techniques

Authors	**Type**	**Static Dynamic**	**Techniques**	**Methods**	**Limitations**
Chao Li [50]	Infrastructure	Static	Multicore	solar power harvesting with multicore architecture, DVFS	Cost overhead, Performance depends on load adaptation.
Huigui [22]	Infrastructure	Static	Flooring, thermal aware servers	DVFS, Server consolidation, improved flooring design	tradeoff between cost and energy
Chao Li [50]	Infrastructure	Static	Location	Solar-Grid as a power source	Dependence on renewable resources
Ali Hammadi [51]	Infrastructure	Static	Cooling System, layout of racks	Dynamic smart cooling, Virtualisation,DVFS	Cooling cost increases drastically
Qingbo zhu [52]	Storage	Dynamic	Cache management	Write back with eager updated, power aware storage cache algorithm, offline power aware greedy algorithm.	Require reduction in response time .

to be cont'd on next page

Table 2.2: Existing energy efficiency techniques

Authors	Type	Static Dynamic	Techniques	Methods	Limitations
Saiqin [53]	storage	Dynamic	Storage	Replication management, cluster reconfiguration	Performance varies with different workloads.
Akshat Verma [54]	Storage	Dynamic	Storage consolidation	Storage consolidation, caching system, write offloading	Required Performance needs improvement
Chao Li [50]	Storage	Satic Dynamic	Energy Storage Devices (ESD)	ESD devices with under different powers where	Requires workload awareness
Farhad Mehdipour [55]	Analytical	Dynamic	Energy efficient map reduce, HDFS	Virtualization, DVFS, clustering	Unable to support an instant execution.
Shadi Ibrahim [56]	Analytical	Dynamic	Energy Efficient Mapreduce	Clustering reconfiguration, DVFS, VM Consolidation	No improvement in energy consumption of Hadoop cluster.
Donato Barbagallo [57]	Scheduling	Dynamic	Bioinspired	VM migration, Ant Colony Optimization	VM migrations causes network Traffic

to be cont'd on next page

Table 2.2: Existing energy efficiency techniques

Authors	**Type**	**Static Dynamic**	**Techniques**	**Methods**	**Limitations**
Tadashi Nakano [58]	scheduling	Dynamic	Bioinspired	AntNet, Virtual machine migration , replication algorithm mechanism	No support for Dynamic workload.
Ashok Gautham [59]	Scheduling	Dynamic	Multicore Parallelism	Multicore, DVFS	Workload varies
Lorenzon [60]	Scheduling	Dynamic	Multicore Parallelism	Embedded , general purpose multicore processors (GPPs), Threadlevel parallelism	Heterogeneous architectures are required for more data flow applications.
Zhang et al. [61]	Scheduling	Dynamic	Consolidation	Virtual Machine consolidation	Required workload prediction and performance improvement

to be cont'd on next page

Table 2.2: Existing energy efficiency techniques

Authors	Type	Static Dynamic	Techniques	Methods	Limitations
Liao et al. [62]	Scheduling	Dynamic	consolidation	VM migration	Ignores power consumption by the network, I/O devices, and GPU
Eugen Feller [63]	Scheduling	Dynamic	voltage	DVFS	Need to evaluate multi job workload
Fbio D. Rossi [64]	Scheduling	Dynamic	voltage	DVFS, PM deallocation algorithm	Least complex algorithms are required.
Christoforos Kachris [65]	Hybrid	Dynamic	Storage and multicore	Scratchpad memory architecture, hashing scheme, map reduce	No support for larger files
Pedro H.P. Castro [66]	Hybrid	Dynamic	Storage and Consolidation	DVFS for RAM,VM migration, CREW(IQRMMT), CREW(LRMMT)	tradeoff between performance energy and SLA
Anton Beloglazov [67]	Hybrid	Dynamic	Consolidation and Voltage	VM consolidation, DVFS	Workload needs to be managed well

Baker [68], in their work, developed the E2C2 algorithm that searches the least possible number of services to fulfill the user requirements. Yoon et al. [69] discussed the adaptive datacenter activation model that consolidates the adaptive activation and deactivation of servers, switches, and hosts for enhancing energy efficiency. Research on energy efficiency is not limited to the servers, but also covers the other parts of the data centers that consume high energy. First of all there is a need to construct energy efficient data center that requires infrastructure techniques. Wei Wu et al. [70] built a framework by integrating Building Information Modeling (BIM) with Wireless Sensor Networks (WSN). The wireless sensor monitored the thermal performance parameters in accordance with the load distribution on the servers [71]. The main focus of this study was on thermal management within the data centers for saving energy. H.S. Sun et al. [72] focused on optimizing space utilization, lighting, zoning control of Heating, Ventilation, and Air Conditioning (HVAC) and room temperature. Their work is limited to data center infrastructure, and they ignored all other factors such as storage and networking that can also be used to save energy.

In addition to infrastructure techniques, it has become essential to store the data using energy efficient storage techniques. Guerra [73] advocated different storage techniques including (consolidation, tiering/migration, compression and write off-loading) which are used in various big data applications. Further, various scheduling techniques are exposed to reduce more energy consumption through different dynamic techniques. Gautham [59] et al. used various scheduling techniques in multicore system to make energy efficient data centers but could not guarantee about high performance and cost of the system. These standalone techniques do not provide the benefits that can be gained by combining some energy efficient approaches. Therefore hybrid energy efficiency technique were brought into consideration to manage tradeoff among energy, cost and performance [74]. Christoforos et al. [65] used scratchpad memory architecture with hashing scheme to re-

Table 2.3: Operational issues for energy efficient techniques

Type	**Issues**	**Solutions**	**Tools/Simulator**
Scheduling	Poor utilization of resources	To reduce workload limitation	Starfish's Elastisizer, All-in-Strategy(AIS)
Storage	Data replication, operational cost, and load balancing	To accelerate domain –specific query language	LINQits
Storage	Disc overhead	To overcome disc I/O limitation and to improve performance of the system, To have scalability,To perform in-memory computation	Spark, DiscSim
Network/Storage	Traffic, Load balancing	To minimize communication overhead	Hadoop
Networking	performance	To reduce data movement	IBM's Netezza
Scheduling/Stoarge	Performance degradation	To improve the performance	DyScale, GPU, CoudSim

duce energy consumption. It could accelerate mapreduce applications by using indexing feature with key/value pairs but it could not support larger files. This proves that applying energy efficiency techniques in cloud computing environment also suffer from certain research challenges, which can be resolved using tools and techniques[75]. From above Table 2.2, its clear that, although techniques help in gaining over all energy efficiency of the system but at the same time many authors have also suffered by various limitations and challenges as described below.

2.2.1 Energy Efficiency Techniques: Research Challenges

Implementing energy efficiency techniques in cloud environment involves various tenacious issues. Most of the issues are common to all the energy efficiency techniques, whereas some are unique and exist only for a particular type of energy efficiency technique.

- For instance, disk overhead is encountered during data storage. The storage techniques also suffer from several other issues such as data replication, operational cost, and load balancing. Saiqin Long et al. [53] addressed these three concerns by designing a Three-Phase Energy-Saving Strategy (TPES) involving the dynamic replica management algorithm, clusters reconfiguration, and dynamic switching.

- Few techniques [53, 61, 66]lacked the performance factor and are unable to keep up with the real-world workloads.

- Poor utilization of resources is sometimes caused by various energy efficiency scheduling algorithms which affect the overall performance of the system.

- Heavy data traffic is seen while using the energy efficiency networking techniques. These generalized problems related to each category, along with the solutions to deal with them, have been depicted in Table 2.3.

- Energy efficient approach given in [5] did not result in significant performance improvement under OLTP workload.

Practitioners and policy analysts have tended to ignore the implications of these methodological problems but it requires attention of researchers to apply them effectively. Description of issues related to each category of energy efficiency techniques along with the solutions to deal with them have been depicted in Table 2.3. It states the maximum issues for storage techniques as storage system in the data centers are dominated by power hunger disks that consumes high latency and energy consumption due random access pattern of applications, specifically for E-commerce e.g. OLTP applications. Therefore, storage energy efficiency in cloud data centers is still in its infancy which can be improved. These storage issues can be removed following the predictable trends for these workloads. Based on the predicted behaviour, the data placement can be performed followed by the disk scheduling that can lead to the energy efficiency [76].

Following section explains the work related to storage energy efficiency techniques in detail.

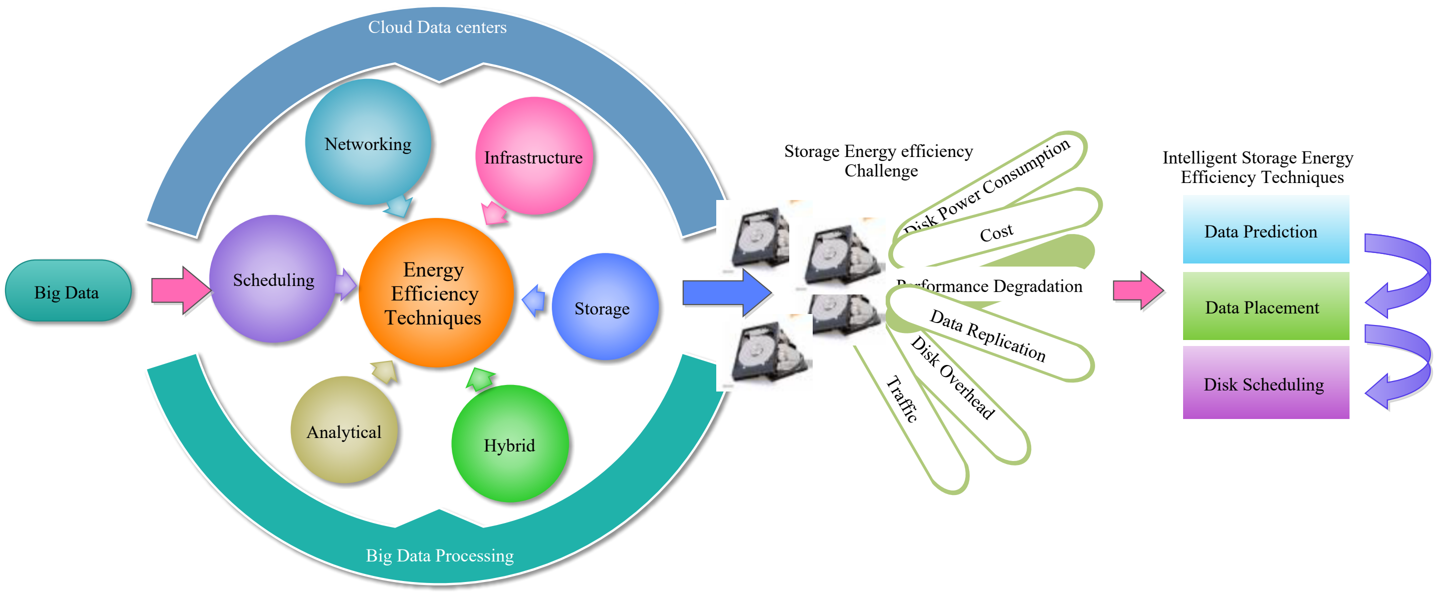


Figure 2.1: Systematic View of storage energy efficiency challenge and solution

2.3 Storage Energy Efficiency Techniques

As shown in Figure 2.1, authors in the field have worked on the energy effect on big data using various energy-efficiency techniques. But various aspects in storage energy efficiency technique like disk overhead, performance degradation, data replication, and disk power consumption still needs to be worked upon. The storage energy efficiency techniques can be improved using data prediction, placement, and disk scheduling techniques altogether. An existing literature on these techniques is stated below-

2.3.1 Data Prediction Approaches

Various techniques such as Neural Networks (NN), clustering models, Support Vector Machines (SVM), decomposition models, Autoregressive Integrated Moving Average techniques(ARIMA) models, gray prediction and regression models have been used in a literature for the prediction purpose.

Table 2.4: Existing prediction techniques on various workload traces

Authors	**Workload characteristics**	**Tools**	**Techniques**	**Target Variables**
Axel Busch et.al [77]	File size, read write max, workload intensity, avg request size access pattern	BLKTRACE	Access pattern recognition	Response times
Jaehyung Kim et.al [78]	I/O type, I/O size, I/O access Patterns, No of Threads, read /Write Ratios	Libaio	ANN	I/O saturation
Salvatore Capra [79]	Job arrival rate, Task duration, Task per job, running task	Faban	Heuristic	Unique job launches, task duration, and tasks per job
Hui Kang [80]	Data per inst, core ID, ADDR, size, load per store, VM ID	DineroIV, QEMU	NA	NA
Tomislav et al [81]	Access type, data's virtual address, access size, thread id and scope	Gleipnir	NA	NA

to be cont'd on next page

Table 2.4: Existing prediction techniques applied on various workload traces

Authors	**Workload characteristics**	**Tools**	**Techniques**	**Target Variables**
Iqbal et al [82]	Response time,document size	real world traces	Non-Negative Least Square (NNLS), Ordinary least squares (OLS)	Uniform Resource Identifier
Matthieu Dorier [83]	Offset, size	C++	Omnisc'IO using formal grammars	Size of data
Noorshams et.al [84]	Request type, Access pattern, Request size, Number of threads, File set size	DRILL	Sequitur, Grammar	Cache block size
Sohail Sarwar [85]	Data part, frequency-counter and a flag	Matlab7	ML using ANN and CBR	Frequency count
James Oly [86]	HYDRO Trace contains satellite mounted radar data	Pablo	Markov model	Block size
A. Galicia [30]	144 attributes of total electrical energy consumption in Spain	spark	Ensemble Model	Data on days

Various equations has been used for predicting measures that can reduce energy consumption of storage sub component. A general approach to predict data is shown in Figure 2.2. Feature extraction phase identifies the most important features that would be used for building accurate model [87]. Finally, the identified model is used for predicting measures that can upbring energy efficiency within the data centers. The constructed model would be validated using accuracy, Root Mean Square Error (RMSE), precision etc. In this work we have been relied upon data frequency prediction so as to estimate frequently used data that can be placed in hot disks. Related work regarding data prediction is described below.

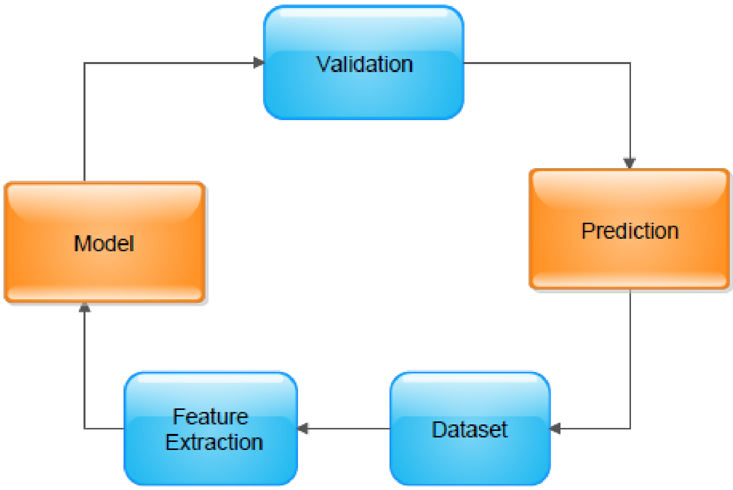


Figure 2.2: Prediction modeling and process

Trace generation is the first step towards the data prediction. Few authors have worked on synthetic traces [77],[78],[88], while others [79] have utilized Faban as a synthetic workload generator. Among all the trace characteristics presented in Table 2.4, a target variable is selected to be predicted using various tools and models. Dorier et al. [83] used 'OmniscIO' approach for building a grammar-based model to predict the I/O data for various applications. Their research has also provided the when and where of future data operations. Moreover, the amount of data to be accessed has also been predicted. OmniscIO failed to discover repetitive structures in the I/O data pattern for applications. Besides, the method is sensitive to data-dependable branches in the code. Noorshams et al. [84] created an I/O queueing model which is parameterized by the service time and request size of the file. The authors also planned to extend their model for mixed workloads. Sarwar et al. [85] came up with more accurate results by applying hybrid approaches using Artificial Neural Network (ANN) and Case-Based Reasoning (CBR). Moreover, oly et al. [86] utilized the markov technique for predicting data requests on scientific applications. They could achieve 90% accuracy for a single step, but attained only 40% accuracy for predicting 25 steps. Chilimbi et al. [89] computed and abstracted the hot data streams using the concept of data reference representation. The author also used a compression technique named sequitur, which could produce a series of representation with increasing compactness.

Vast explanation of traces used and prediction techniques applied on target variables by various authors has been detailed in Table 2.4. It could be inferred that only Sarwar [85] utilized machine learning approach on the frequency of data; however they did not use their technique for improving energy efficiency. To remove this discrepancy, we have made efforts to applied data placement techniques to distribute the predicted data according to its usage which can result in high energy efficiency. The work related to the data placement approaches techniques is explored in the following subsection.

2.3.2 Data Placement Approaches

An efficient data placement approach is an essential requirement for energy efficient data centers. The predicted data can be used for placement that results in concentrating the popular request within few disks. Many authors have worked on this concept but very few of them have discussed data placement techniques with respect to the energy efficiency. Wildani et al. [90] proposed a various methods for grouping a data using the predictive value to make layout decisions on the disk. However, the authors did not focus on reducing the energy consumption. To eliminate this discrepancy, Meng et al. [91] developed a Power-Aware Multi-Level (PAM) cache policy using Solid State Drive (SSD) that could extend the standby duration of the data disks and result in energy savings. Further to reduce the execution time, Xiaoyu Yao et al. [92] proposed a Redundancy-based Cache Architecture called (RIMAC). The proposed architecture helps in reducing the time taken to execute the requests by replicating the data on active disks or direct I/O caches. However, this concept increases the data migration. This limitation was removed by Farhad Medipour [55] by using IBM's Netezza, which successfully reduces the data movement by means of innovative hardware acceleration. A. Verma et al. [54] proposed Sample-Replicate Consolidate Mapping (SRCMap), a technique which consolidates the cumulative workload on a subset of physical volumes in a manner that facilitates energy proportionality for dynamic I/O workloads. Yuan et al.[93] suggested a Multi-Level Caching Strategy (MCS-SSD) to reduce energy consumption by clustering the correlated files that can be placed on the data disk and cache disk SSD via temporal and spatial locality. Islam Atta [48] et al. grouped similar transactions of OLTP application into teams to improve the data locality. Authors here did not reduce energy using data locality. To further improve the energy consumption along with the performance, Xie et al.[94] presented a hybrid disk storage called Performance, Energy, and Reliability balanced (PEARL) that distributes the data dynamically between hard disks and flash disks by adapting the changes in the data access patterns. The proposed architecture could reduce energy consumption. The authors in the work did not consider the scheduling of the request which is the most important phase that contributes in energy savings. The works on storage energy efficient disk scheduling using various thresholds techniques are explained in the following section.

2.3.3 Disk Scheduling Approaches

Disk scheduling evolves various threshold approaches that can be applied on disk idle time and waiting time. Various researchers have attempted to reduce the energy consumption using disk idle time productivity. Jerry et al.[32] designed a power management scheme that spins the disk down on experiencing a fixed idleness threshold. To refine this work further, Xiaodong et al. [95] proposed an optimized threshold-free algorithm that minds the state of the disk to save the spinning energy. Nevertheless, incorporating and detecting the predictable phases were left for future work. Golding [96] recommended various schemes to use the idle time productively and reduced the energy consumption.

The threshold techniques are also employed during the request execution. Chou et al.[32] has investigated energy-aware scheduling algorithms such as offline scheduling, batch scheduling using the weighted set cover, and online scheduling for accessing the requests on the disk in the storage systems. Li et al. [97] developed an Energy-Efficient RAID system architecture (EERAID) to conserves energy. EERAID 1 worked on Windows Round-Robin (WRR) and Power and Redundancy-Aware Flush (PRF) scheduling policy, they did not used energy efficient data placement. Xie [34] integrated energy-efficient placement with scheduling approaches by developing Striping Based Energy Aware (SEA) approach that could improve the energy efficiency. Further, the execution time should also be considered along with the energy efficiency which can be reduced using data replication. Boru [31] proposed a replication solution to optimize trade off between energy and time. Zhang [98] et al. also designed a power-aware data replication strategy by leveraging data access behavior, but the authors did not implement their technique in a cloud environment. To remove such discrepancy, recently, a few studies have focused on the energy efficiency of high end storage system in cloud environment. Sturm et al. [99] proposed a storage extension for CloudSim that enables the simulations of Storage as Service (STaaS)-components. But their work is pattern dependent that make simulation less realistic and also lacked allocation policies used in storage. Long et al. [100] added the file replica management function by extending the cloud simulation platform. Although the proposed model improved system performance, but it could not save energy and the cost.

Table 2.5: Existing energy efficient prediction placement, scheduling techniques for storage energy efficiency challenge

Authors	**Domain**	**Strategy**	**Advantages**	**Disadvantages**
Wildani et.al [90]	Prediction and placement	N-Neighborhood Partitioning(NNP)	Authors predicted future access probabilities and grouped similar kind of data.	Authors did not work on reducing disk power consumption.
Meng et.al. [91]	Data placement	PAM policy	The applied technique could extend the standby period of the data disks.	Prediction approaches need to employed.
Xiaodong et al.[95]	Energy efficient threshold tech	Performance Directed Static (PS) algo.	Algorithm is self-tuned and dynamically adjusts thresholds.	Detecting predictable phases is left as future work.
Jerry et al.[32]	Energy aware disk scheduling approach	weighted set-cover algorithm.	There is no interference with the existing data placement or power management policies.	Machine learning approaches were not employed to predict disk idle time.
Yuan et al.[93]	Energy eff placement and scheduling	MCS-SSD	Energy-saving approach is applicable for astronomical observations and uses both cache disks and data disks.	Data placement is done through priority queue which could be improved through request arrival prediction.
Skyprvazquez et al. [101]	Energy efficient prediction	MPC	MPC covered prediction model along with the cost function.	Work is more hardware oriented.

to be cont'd on next page

Table 2.5: Existing energy efficient prediction, placement and scheduling techniques for storage energy efficiency challenge

Authors	**Domain**	**Strategy**	**Advantages**	**Disadvantages**
Paul et.al [102]	Energy efficient prediction	Deep learning	Outlier in idleness patterns are discarded.	Predicting disk idleness is NP hard problem.
Galicia et.al [30]	prediction approach	Ensembled learning	It is the most accurate model that achieved only 2% of MRE.	Authors did not apply the proposed technique to reduce energy.
Behzadnia et. al [103]	placement	DPM	Supports inter-disk data migration and compatible with any multi-speed disk storage system.	Prediction error rate increases for farther epochs.
Helmbold et. al [104]	Energy aware threshold approach	Multiplicative Weight Algorithm	The proposed technique is used in mobile computing	Self tuning version is required.
Gao et. al [105]	Energy aware prediction approach	ant colony optimization, and control theory techniques	VM resizing and server consolidation are used to achieve energy efficiency.	Authors did not laid focus on disk energy consumption.

to be cont'd on next page

Table 2.5: Existing energy efficient prediction, placement and scheduling techniques for storage energy efficiency challenge

Authors	Domain	Strategy	Advantages	Disadvantages
Golding et.al [96]	energy aware prediction and threshold approach	Machine learning techniques	Algorithm automatically adapt with the changing workload.	Authors did not include placement techniques in their work.
Yin et. al [106]	Energy Efficient placemeny	Reliable Energy Efficient RAID(REED)	Improved reliability by integrating HDDs and SSDs.	The work is implemented only in simulated environment.
Mahajan et al. [107]	Energy Efficiency placemnts	Query optimization	Used Dynamic Voltage and Frequency Scaling. (DVFS)	Did not use prediction approaches for caching the data.

In order to save energy, Louis et al. [108] proposed Hard Disk Drive (HDD) models which is validated CloudSimDisk but needs to be compared with real-world measurements. To come up with the limitation and to advance the research efforts in terms of performance, we have proposed an intelligent energy efficient storage approach which has been tested in real world environment using servers and has also been simulated in cloud environment.

Table 2.5 displays the boons and banes of the existing works. It can be noticed that few authors have worked on prediction approaches while others have concentrated on placement approaches. Very few of them have combined both the techniques while ignoring scheduling techniques. Also, despite the various advantages, certain gaps remain, which

Table 2.6: Energy efficiency parameters covered in literature

Author	Trace Generation	Data Placement	Prediction	Energy Efficiency	Real time Execution	Replication	Performance	Scheduling
Xie et al.[34]	×	✓	×	✓	✓	×	✓	✓
Zhang et al. [98]	✓	✓	×	✓	✓	✓	✓	
sturm et al. [99]	✓	×	×	×	×	×	✓	
caulfield et al.[109]	×	✓	×	✓	×	✓	✓	
louis et al. [108]	×	×	×	✓	×	×	✓	
grozev et al. [110]	✓	✓	✓	×	×	✓	✓	✓
long et al. [100]	✓	✓	×	×	✓	✓	✓	✓
yan et al. [26]	✓	×	×	✓	×	×	×	×
xie et al. [94]	×	✓	×	✓	×	×	✓	×
Li et al. [97]	×	✓	×	✓	×	×	✓	×
Wildani et.al [90]	×	✓	✓	×	×	✓	✓	×
Meng et.al. [91]	×	✓	×	✓	×	×	✓	×
Xiaodong et al.[95]	×	×	✓	✓	×	×	✓	×
Jerry et al.[32]	×	✓	×	✓	×	✓	✓	×
Yuan et al.[93]	×	✓	×	✓	×	×	×	×
Skyprvazquez et al. [101]	×	×	✓	✓	✓	×	×	×
Paul et.al [102]	×	×	✓	✓	×	×	✓	✓
Galicia et.al [30]	×	×	✓	✓	×	×	✓	×
Behzadnia et. al [103]	×	✓	✓	✓	×	×	✓	×
Helmbold et. al [104]	✓	×	✓	✓	×	×	✓	×
Golding et.al [96]	×	×	✓	✓	✓	×	✓	✓
Gao et. al [105]	×	×	✓	✓	×	×	✓	✓

need to be filled. So, their is the need of the system that can club energy efficient prediction with an allocation using replica management along with an intelligent scheduling to save the energy consumption at larger extent.

2.4 Gap Analysis

Among the reviewed papers in Table 2.6, only the authors in [98], [99], [100], and [104] have used the trace generation procedure. Many researchers [91] have worked on energy efficient placement techniques but have failed to synchronize them with energy efficient prediction approaches. Although the investigators in [101] and [96] have executed their work in real-time environment using prediction, they lagged in data placement, which plays an important role in overall energy reduction. Based on the literature survey following gaps have been identified:

- Many authors have made use of online available traces rather than generating it. Also, the existing work till now has not covered SQL traces associated with the commercial applications [81], [111]. So, the storage energy consumption for E-commerce applications that target a specific class of systems, like OLTP needs to be explored using SQL traces [98].

- Moreover, authors have not predicted the data frequency using machine learning models. To the best of our knowledge, prediction of data access pattern for obtaining energy efficiency is still in infancy [84], [30].

- Many researchers have worked on energy efficient placement techniques but have failed to synchronize them with energy efficient prediction approaches.

- Even though the authors have worked on the energy efficient data placement, but, at the same time they have not introduced the replica optimisation in their work [94]. So, an energy-efficient data placement approach should include optimal replica management in order to avoid data redundancy [31].

- Further, there is also a need to examine the energy and time consumed while placing the data among the disks [31].
- Few authors have worked on energy efficient Redundant Array of Independent Disk (RAID) but they have not applied intelligent disk scheduling for request execution that can save maximum energy by reducing maximum disk spins [97],[32].
- The cloud data centers demand real-time execution on the server rather than depending only upon simulations [34].
- Also, there is a need to standardize the best replication factor and disk ratio for applications to save maximum energy[108].

2.5 Problem Statement

In the world of overflowing data, cloud storage has become a prevalent technology for saving information-driven data. Cloud data centers provide on-demand computing storage to users. It is used for storing and processing I/O-intensive applications. However, the cloud deployment of these applications raises various challenges including energy efficiency. Energy consumption for data centers has always been an major concern. Also the power consumed by the storage subsystem which includes disks occupy a remarkable proportion in data centers. Hence, prediction based an intelligent energy-aware approach is required that uses data prediction to segregates and place the data in the disks according to its usage and follows intelligent disk scheduling.

Therefore, in this work we have attempted to reduced storage energy consumption in the cloud environment using data prediction approaches. The frequency of data has been predicted using machine learning approaches such that the data that has been predicted to be used most frequently is placed in the hot disk using replication. In addition, disk scheduling has been implemented that selects the disk that consume least power to execute the request. It reduces the disk spins that accelerates the data execution time, hence reduces maximum energy. So, following summarises the research objectives.

- To study and explore various existing storage prediction models and energy efficiency techniques for analyzing big data storage.
- To propose a storage prediction model for identifying data usage patterns.
- To design and implement an intelligent energy aware approach using proposed storage prediction model.
- To validate the proposed energy aware approach in cloud environment for offering requisite performance to the end users.

2.6 Conclusion

This chapter has presented the view on big data and its journey in cloud data centers. It explicates various big data challenges, where storage energy efficiency challenge has been recognised as a major concern that requires attention. So, to improve the storage energy efficiency techniques, various prevailing energy-efficient data prediction, data placement, and disk scheduling techniques are reviewed. It has also focused on exploring the tools applied by authors to implement the storage energy-efficient techniques. From the literature survey, the gaps have been analyzed and the research problem has been formulated. The next chapter proposes a solution to the research problem by proposing storage prediction model using an machine learning ensembled approach for predicting data frequency. The proposed model addresses the gaps identified in the literature survey and fulfills the objectives of the research work.

Chapter 3

Data Frequency Based Storage Prediction Model

Technological advancements have led to an exponential growth in input-output intensive data applications that consumes high energy within data centers. Theses applications are accessed by firing same queries frequently. So, the energy consumption by these application can be reduced by predicting the frequency pattern of the queries. It can be used to identify future data usage pattens such that the popular and unpopular data can be recognised. The popular data can be placed in the SSD cache and remaining in the other disks such that the disk spins can be reduced that results in least energy consumption. Therefore, in this chapter a storage prediction model has been proposed that identifies the data usage patterns. The future query requests have been identified by predicting frequency of firing query using classified ensemble machine learning approach.

Initially, the traces have been generated from the SQL queries fired on the real data streams obtained from the with SCAT sensors of Dublin city, taken as a commercial application. The traces are analyzed and customized to calculate the frequency of each query. Feature selection is performed to reduce the size of the dataset. Next, ensembled approach is applied to predict the frequently used data using frequency as a target variable. Furthermore, the predicted classified results have been tagged as popular and unpopular data based on predicted threshold frequency such that the identified popular data is prefetched into the SSD cache as described in next chapter. The proposed approach is evaluated using accuracy, error rate, precision, recall, ROC. It outperforms the existing models with an improved accuracy and reduced error rate.

3.1 Objectives of the Proposed Storage Prediction Model

The cloud data centers provides infrastructure to host and execute various applications that can result in high energy consumption. Hence, to reduce the high energy consumption there a need to introduce various prediction models that can be integrated with energy aware approaches to reduce disk spins.

The literature has been published on data prediction as discussed in Section 2.3.1 of Chapter 2 but numerous challenges still exist.

- Various authors have used synthetic traces for identifying data usage patterns but SQL traces associated with the commercial applications needs to be explored.
- Various machine learning algorithms needs to be employed to predict data frequency. Their performance needs to be enhanced in terms of accuracy, error, precision, recall, f-score and ROC.
- Data tagging is still a necessary requirement to efficiently place the data into the disks according to its predicted popularity using threshold techniques.

To overcome the above-mentioned challenges, the storage prediction model is proposed in this chapter. It predicts the data frequency of the queries fired on the real data streams obtained from the SCATS sensors of Dublin city using classified ensembled machine learning approaches. The primary objectives of the proposed storage prediction are summarized below:

- The proposed storage prediction model generates the SQL traces for city's SCAT data which is taken as a commercial application. It customises the trace by extracting various other important features such as data frequency and idle time that can help in identifying current data usage pattern.

- A feature selection and importance has been performed on a trace for selecting the relevant features. It reduces the size of the data by eliminating the irrelevant and redundant features.
- It ensembles various machine learning model to predict the calculated frequency for each query present in the trace. The performance is validated using metrics such as accuracy, precision, recall, ROC etc.
- The predicted data is tagged as popular and unpopular using predefined threshold such that predicted frequency greater than the threshold is termed as popular and remaining as unpopular data.

3.1.1 Key Traits of Storage Prediction Model

The key traits of the proposed prediction model are mentioned below:

- It is the first step towards the energy reduction for the cloud data centers.
- The prediction of data frequency helps in identifying future popular and unpopular data.
- The prediction results can further be used to efficiently place the data among multiple disks such that disk spins can be reduced leading to least energy consumption.

After illustrating the key features, the following section presents the detailed methodology used in proposing the storage prediction model.

3.2 Methodology

The methodology used in our work is fully based on the storage prediction model depicted as a framework in Figure 3.1. It follows the sequence of phases, which includes (i) trace generation (ii) prediction using machine learning approaches (iii) data tagging:popular and unpopular. In the beginning, workload traces containing various important attributes have been generated as explained in subsection (3.2.1). Various machine learning models have been employed to predict the query data by selecting data frequency as the target variable as described in subsection (3.2.2). Predicted data that exceeds the threshold limit of frequency has been tagged as popular data as revealed in subsection (3.2.3).

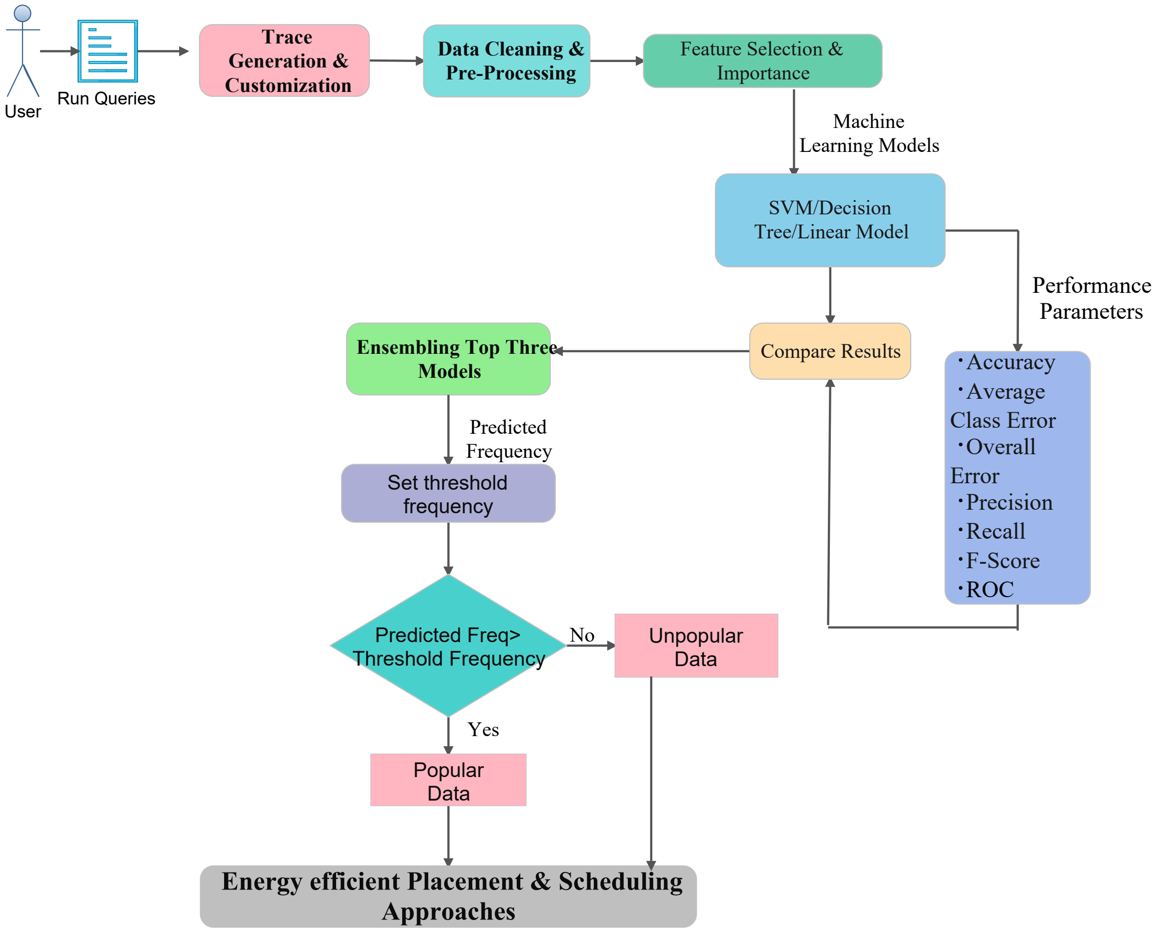


Figure 3.1: Methodology used in proposed storage prediction approach

3.2.1 Trace Generation

It is essential to analyze the trace of an activity to predict I/O data for the system. Traces can be categorized into real workload and synthetic. The former is usually generated from scientific production using supercomputers, whereas the latter is derived using probability distributions. In this work, traces, are generated through database access of application by the user. The procedure has been detailed in Figure 3.2. The SQL server traces

Table 3.1: Basic parameters of the trace

Sr no	Parameter	Description
1	Stored Procedures	Includes the event classes produced by the execution of stored procedures
	RPC:Completed	Occurs on the completion a remote procedure call
	SP:StmtCompleted	Illustrates the completion of a transact-SQL statement associated with a stored procedure
2	TSQL	Includes event classes produced by the execution of Transact-SQL statements passed to an instance of sql server from the client
	SQL:BatchCompleted	Occurs at the termination of Transact-SQL statement

are prepared on firing various queries on the dataset associated Dublin city, Ireland. The queries and stored procedure (Refer Table 3.1)are reconstructed by the software in response to some external mechanisms [112], which are taken as inputs by the profiler to produce a trace. The trace captures various features such as the start and end time of the query, its duration, text data of the query, SPID, database ID, database name, reads, writes, row count, application name, CPU and server name, etc (Refer to table 3.2). These features alone are not sufficient for recognizing the data usage pattern [113]. It requires additional features such as query frequency and requests idle time. The frequency of query is required to predict future queries to segregate the popular and unpopular data. The request idle time would be used to set a threshold to disk idle time which would be used in a next chapter. These two most important features are extracted by customizing the currently available features and are appended with the trace(Refer Figure. 3.2).

- Calculating Request Idle time

  Idle time is calculated by applying the arithmetic operations on the current characteristics of the trace. To compute the idle time, the start and end time attributes of the trace are used. These selected parameters are set in the form of a query metric M, where β_i and γ_i are the start and end times of the queries, respectively. Then, the idle time I_i calculated using Eq (3.1) and its threshold is calculated as TI_i using Eq (3.2) .

$$M = \begin{array}{l} \\ Query_1 \\ Query_2 \\ \vdots \\ Query_n \end{array} \begin{array}{c} Queries \quad StartTime \quad EndTime \\ \begin{pmatrix} \beta_1 & \gamma_1 \\ \beta_2 & \gamma_2 \\ \vdots & \vdots \\ \beta_n & \gamma_n \end{pmatrix} \end{array}$$

$$I_i = \beta_{i+1} - \gamma_i \tag{3.1}$$

where i= 1,2,3 … n-1, γ_i>β_i and β_{i+1}>γ_i

$$TI_i = \frac{\sum_1^n I_i}{n} \tag{3.2}$$

- Calculating Query Frequency

Initially, frequency was not present in the trace, which is an essential feature for recognizing the data usage pattern. Frequency of the data is one of the important features for recognizing the popular and unpopular data. Therefore, to extract the frequency of data usage, the generated trace is customized with the help of a predefined tool called analyzer. It calculates the frequency $freq(i)$ for the occurrence of each query fired by the user and appends it in the trace. This calculated frequencies would be used as a target variable to predict the most frequently used queries which would further be used at the time of data placement.

So, various machine learning models have been ensembled for predicting the data frequency against the calculated frequency as described below.

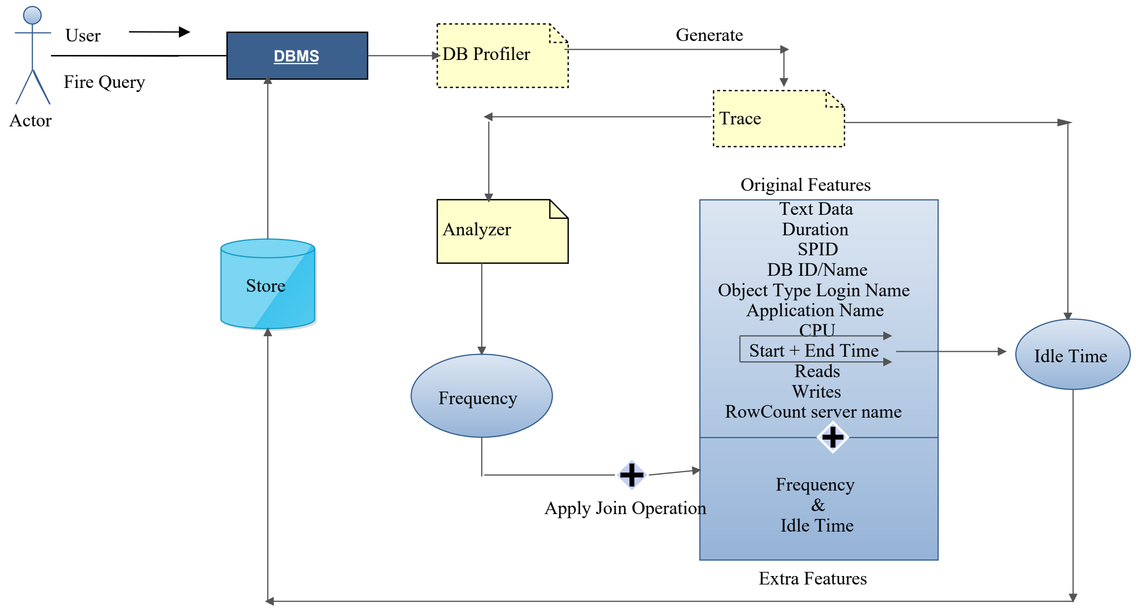


Figure 3.2: Synthetic workload generation procedure

3.2.2 Machine Learning Models for Data Frequency Prediction

Prediction is used for estimating the data that would be used in the near future [114]. The main purpose of using machine learning model is to classify and predict the queries and stored procedures on the basis of number of occurrences i.e, frequency. Various multi classification-based algorithms like linear, SVM, and decision tree have been ensembled using Multi Class Voting(MCV) technique. These models have been applied to the system-generated excel file of a trace. The mathematical aspect of all the models used in ensembling are described below-

(a) Linear and Generalised Linear Models

A generalized Linear Model, abbreviated as (LM) fits the statistical model with the data. It can be used for numeric as well as continuous target variables. The generalized algorithm is characterized by the distribution of the target variable and a link function that relates the mean of the target with the inputs. The parameters

Table 3.2: Illustration of features of trace

Feature	**Abbreviation**	**Information**
Text Data	td	Represents the text value for data captured in the trace
Duration	d	Amount of time taken by the event in milliseconds
SPID	sp_{id}	Represents a server process ID for an associated process assigned by server
Database ID	db_{id}	Depicts database ID used by the User
Database Name	db_{name}	Depicts the name of the database currently used by the user
Object Type	obj_{type}	Represents the type of the object involved in the event
Login Name	log_n	Specifies login name of the user
Application Name	App_{name}	Specifies the name of the application being accessed by the user
CPU	cpu	Calculates the amount of CPU time in milliseconds
Reads	r	Number of logical reads performed by server
RowCount	r_count	Depicts the number of rows accessed for particular query or stored procedure
Server Name	log_n	Provides Name of sql server being traced
Writes	w	Number of logical writes performed by the server

used here belong to families like logistic, poisson etc. The outcome dependent variable, Y is generated from an exponential distribution. The mean of the distribution depends on the independent variable, X, as given in Eq 3.3:

$$E(Y) = \mu = g^{-}1(X\beta) \tag{3.3}$$

where E(Y) represents the expected value of Y; X denotes the linear predictor i.e. a linear combination of unknown parameters; and g represents the link function.

(b) Decision tree

The decision tree model is a popular data mining model. Recursive partitioning approach is adopted by this algorithm using rpart package. The conditional tree algorithm is implemented in the package. The branching operations performed here may be termed as "tests" or "queries". Mathematically, the model is described through Eq 3.4.

$$f : \{0,1\}^n \rightarrow \{0,1\} \tag{3.4}$$

where the input refers to a series of queries that results in the final decision. Every query depends on previous ones. The depth of a tree is defined as the maximum number of queries that are fired before reaching a leaf node in order to obtain a result.

(c) Support Vector Machine

SVM is a supervised machine learning algorithm, covering both classification and regression challenges [115]. However, it is mostly used for solving classification problems. SVM finds support vectors termed as data points, which lie at the edge of an area in a space that frames the boundary within different classes [116]. An n-dimensional space with the value of each feature being the value of a particular coordinate is plotted (where n signifies the number of features). Finally, these three models are selected to be ensembled using MCV as described below.

(d) Majority Class Voting Ensembling

Ensembling is a method through which multiple models of similar or dissimilar types are combined to give rise to a robust one. This is done to enhance the model in terms of accuracy and reliability [117]. We have applied Multi Class Voting(MCV) for ensembling our models that considers the prediction with maximum votes or recommendations from multiple implemented prediction models as shown in Algorithm 1.

In the algorithm, the major element for the target variable, i.e., frequency, is voted by maintaining a count of the data frequency. The count is incremented and decremented according to the conditions applied. The basic idea is to cancel out each

Algorithm 1: Multi Class Voting

Initialize index and count of majority element
$maj_{index} = 0$, count = 1
for *(i=1 to size=-1)* **do**
  if $a[maj_{index}]$ == *a[i]* **then**
    count++
  end
  else
    count–
  if *count == 0* **then**
    $a[maj_{index}]$=i
    count=1
  end
end
return $a[maj_{index}]$
print Majority (a[i],size)

occurrence of an element with other such that the one that exists till the end will be the majority element. It provides the value of predicted frequency as $pfreq(i)$.

3.2.3 Data Tagging Using Predicted Threshold Frequency

Further, the queries with higher predicted frequency have a greater probability of being fired by a user in the near future. So, a predetermined threshold is defined to tag popular and unpopular queries. It is calculated by averaging the predicted frequency using in Eq. 3.5, were A_t signifies the threshold for predicted frequency. The query with predicted frequency $pfreq(i)$ greater than A_t are termed as popular Q_h= $\{q_i, \ldots, q_m\}$ and remaining as unpopular queries Q_c= $\{q_i, \ldots, q_m\}$ belonging to set of total queries Q= $\{q_i, \ldots, q_n\}$ as seen in Algorithm 2. Here s_i represents the size of a query and $freq(i)$ states query frequency. According to the skew degree, the popular queries can be written as $\left\{|Q_h|\right\} = (1-\theta) * m$ and unpopular files as $\left\{|Q_c|\right\} = (\theta) * m$. Further data associated with these tagged queries are saved in the disks as described in a next chapter.

$$A_t = \frac{\sum_{i=1}^{n} pfreq(i)}{n} \tag{3.5}$$

Algorithm 2: Data Frequency Prediction

Result: Popular Q_h and Unpopular Data Q_c
Input : $A_t \leftarrow$ Threshold Frequency
$pfreq(i) \leftarrow$ Predicted Frequency
Output: Tagged Data

Generate the trace using SQL Profile
Calculate the frequency $freq(i)$ of each Query using SQL Analyser
Predict frequency $pfreq(i)$ using Algorithm 1
Calculate A_t using Equation 3.5
for *(i=0;i$\leq$ $freq(i)$;i++)* **do**
  if $pfreq(i) \geq A_t$ **then**
    Tag it as a popular Query
  end
  else
    Tag it as unpopular Query
end

Finally, the prediction approach is evaluated using various metrics as described below.

3.3 Evaluation Metrics: Best Prediction Model

The storage prediction model tags the data by predicting its frequency by ensembling machine learning algorithms namely SVM, NN, Decision Forest. These models have been compared with each other by several evaluation criteria. Based on recent surveyed literature, the best classification algorithm is selected by evaluating its accuracy, average class and overall error, precision, recall, F-measure and ROC.

(a) Accuracy

In terms of classification, accuracy is defined as a measure which monitors the correctness of the classifier. Mathematically, it is calculated using the Eq 3.6

$$Accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples-1}} (\hat{y}_i = y_i) \tag{3.6}$$

where $\hat{y}$ denotes the predicted value of i^{th} sample and y_i represents the corresponding value.

(b) Error Rate

Assume that M is an error metric for multi-class classification, which contributes to the calculation of the average class error and overall error. The value for each cell of the metric is formulated using a number of predictions for each class.

$$M = \begin{bmatrix} & \beta & \alpha & \delta & \\ \beta & \beta & \delta & \alpha & \varepsilon 1 \\ \alpha & \delta & \beta & \alpha & \varepsilon 2 \\ \delta & \gamma & \gamma & \gamma & \varepsilon 3 \end{bmatrix}$$

- *Average Class Error*

  The average class error is the error rate of a particular class in multiclass classification. The measure can be calculated by using the value as provided in Eq 3.7.

$$Average\, Class\; Error = \sum \varepsilon \tag{3.7}$$

  where $\varepsilon = N_{fp}/\sum P_r$ and N_{fp} stands for the number of false prediction for each row and P_r denotes the prediction for a particular row in a matrix.

- *Overall Error*

  Overall error is defined as the sum of all the wrong predictions in a metric M. Mathematically, it can be derived through Eq 3.8, where w_p depicts the wrong predictions.

$$Overall\; Error = \sum w_p \tag{3.8}$$

Other important statistical parameters include precision, recall, F-measure and ROC. These values can be computed using True Positive(TP), False Positive(FP), True Negative(TN), and False Negative(FN).

TP = correctly identified frequency of predicted data

FP = incorrectly identified frequency of predicted data

TN = correctly rejected frequency of predicted data

FN = incorrectly rejected frequency of predicted data.

(c) Precision

Precision is used to describe the goodness of the model in predicting the positive class. Mathematically, it is calculated as a ratio of true positives and the sum of the true and false positives as shown in Eq 3.9.

$$Precision = \frac{TP}{(TP + FP)} \tag{3.9}$$

(d) Recall

Recall defines the proportion of correctly classified actual positives based on the frequency of the query. Eq 3.10 calculates it as the ratio of the number of true positives and the sum of true positives and false negatives.

$$Recall = \frac{TP}{(TP + FN)} \tag{3.10}$$

(e) F-Measure and ROC

F-Measure calculates the harmonic mean for recall and precision as seen in Eq 3.11, whereas, area under the curve method is used to access the ROC curve. The maximum value for the area under the curve signifies the best predictor algorithm.

$$\frac{2 * TP}{(2 * TP + FP + FN)} \tag{3.11}$$

Upon fulfilling the performance evaluation for prediction, the next step would be to prefetch and place the predicted data in a disk cache which is well explained in further chapter.

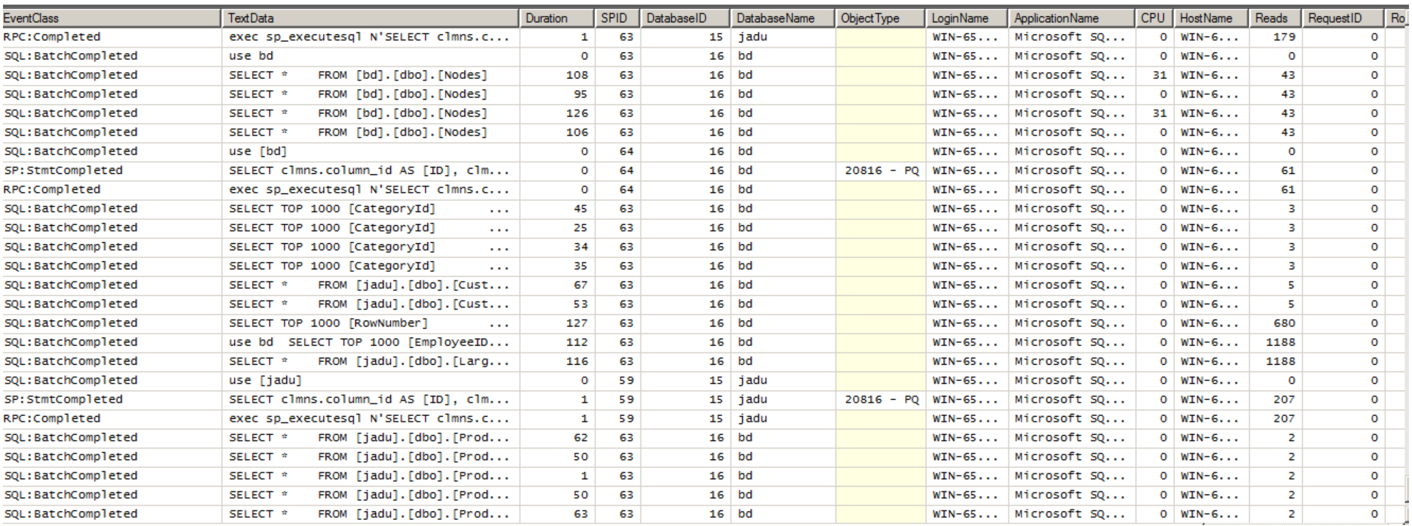
3.4 Experimental Results and Analysis

The experiments are performed using the server: Intel Xeon e5 2650 2.6 GHz, RAM -DDR3 8GB 1600MHz, HDD-SATA 500GB@7200RPM. The system has been installed with server 2008r2. Number of queries and stored are fired on the real data streams obtained from the SCATS sensors of Dublin city in SQL server management studio. SQL Server profiler has been used to generate the traces for these queries. These traces have been further customized using the SQL server analyzer. The analyzer tool calculates the frequency of the queries in the traces. The traces are refined using feature importance and feature selection. Various machine learning algorithms are implemented using Rattle and R studio 3.0.1 that predict the popular data. The performance of the prediction algorithm is measured using accuracy, average class error, overall error, precision, recall, F-measure and ROC. The experimental results are compared against previous strategies and highly used benchmarks. For the sake of simplicity, the experimental setup is categorized into the following phases.

- Dataset description & Trace generation
- Data preprocessing
- Validation and Selection best prediction model
- Performance evaluation of selected ensembled model

3.4.1 Dataset Description: City's Scat Data

The proposed storage prediction model has been evaluated considering SQL queries fired on [3] [118] the real data streams obtained from the SCATS sensors of Dublin city, taken as a commercial applications. Nine GB of data from 966 sensors has been collected from January to April 2013. Forty-seven datasets are publicly available, which incorporate the real-time information gained about the shortest route, most famous areas, parking sites, tower locations, libraries, fire stations, hospitals, etc. Further, each data set contains unique columns such as ID, X coordinate, Y coordinate, height, address, town, telephone, email, etc. The queries have been invoked by the users using a dashboard and a reporting tool that provides an interface to construct them. Besides, the queries have also been reconstructed by the software in response to some external mechanisms [112]. Further, these fired queries have been taken as inputs to generate the trace as detailed below.



EventClass	TextData	Duration	SPID	DatabaseID	DatabaseName	ObjectType	LoginName	ApplicationName	CPU	HostName	Reads	RequestID	Ro
RPC:Completed	exec sp_executesql N'SELECT clmns.c...	1	63	15	jadu		WIN-65...	Microsoft SQ...	0	WIN-6...	179	0	
SQL:BatchCompleted	use bd	0	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	0	0	
SQL:BatchCompleted	SELECT * FROM [bd].[dbo].[Nodes]	108	63	16	bd		WIN-65...	Microsoft SQ...	31	WIN-6...	43	0	
SQL:BatchCompleted	SELECT * FROM [bd].[dbo].[Nodes]	95	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	43	0	
SQL:BatchCompleted	SELECT * FROM [bd].[dbo].[Nodes]	126	63	16	bd		WIN-65...	Microsoft SQ...	31	WIN-6...	43	0	
SQL:BatchCompleted	SELECT * FROM [bd].[dbo].[Nodes]	106	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	43	0	
SQL:BatchCompleted	use [bd]	0	64	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	0	0	
SP:StmtCompleted	SELECT clmns.column_id AS [ID], clm...	0	64	16	bd	20816 - PQ	WIN-65...	Microsoft SQ...	0	WIN-6...	61	0	
RPC:Completed	exec sp_executesql N'SELECT clmns.c...	0	64	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	61	0	
SQL:BatchCompleted	SELECT TOP 1000 [CategoryId] ...	45	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	3	0	
SQL:BatchCompleted	SELECT TOP 1000 [CategoryId] ...	25	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	3	0	
SQL:BatchCompleted	SELECT TOP 1000 [CategoryId] ...	34	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	3	0	
SQL:BatchCompleted	SELECT TOP 1000 [CategoryId] ...	35	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	3	0	
SQL:BatchCompleted	SELECT * FROM [jadu].[dbo].[Cust...	67	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	5	0	
SQL:BatchCompleted	SELECT * FROM [jadu].[dbo].[Cust...	53	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	5	0	
SQL:BatchCompleted	SELECT TOP 1000 [RowNumber] ...	127	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	680	0	
SQL:BatchCompleted	use bd SELECT TOP 1000 [EmployeeID...	112	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	1188	0	
SQL:BatchCompleted	SELECT * FROM [jadu].[dbo].[Larg...	116	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	1188	0	
SQL:BatchCompleted	use [jadu]	0	59	15	jadu		WIN-65...	Microsoft SQ...	0	WIN-6...	0	0	
SP:StmtCompleted	SELECT clmns.column_id AS [ID], clm...	1	59	15	jadu	20816 - PQ	WIN-65...	Microsoft SQ...	0	WIN-6...	207	0	
RPC:Completed	exec sp_executesql N'SELECT clmns.c...	1	59	15	jadu		WIN-65...	Microsoft SQ...	0	WIN-6...	207	0	
SQL:BatchCompleted	SELECT * FROM [jadu].[dbo].[Prod...	62	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	2	0	
SQL:BatchCompleted	SELECT * FROM [jadu].[dbo].[Prod...	50	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	2	0	
SQL:BatchCompleted	SELECT * FROM [jadu].[dbo].[Prod...	1	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	2	0	
SQL:BatchCompleted	SELECT * FROM [jadu].[dbo].[Prod...	50	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	2	0	
SQL:BatchCompleted	SELECT * FROM [jadu].[dbo].[Prod...	63	63	16	bd		WIN-65...	Microsoft SQ...	0	WIN-6...	2	0	

Figure 3.3: Characteristics of trace generated

3.4.2 Trace Generation

Profiler, takes the queries as an input to generate the traces. Figure 3.3 displays the features of trace which includes reads, writes, database name, database id, duration, CPU,

[3] http://www.dublinked.ie.

number of rows, and TextData. Various queries fired by the users are analysed by analyser tool. It adds the addition feature i.e. frequency in a trace to keep count on each fired query. Frequency and TextData are used as the key features, where TextData is showing all the queries fired by the user for which the analyser would calculate its frequency. Once the trace has been generated, it is saved in a form of excel file. Data preprocessing has been performed to reduce and refine the large number of features present in a traces. Feature importance and selection used in the process of preprocessing have shaped the trace with relevant features. The next subsection highlights the same.

3.4.3 Data Preprocessing

Data preprocessing includes feature selection and feature importance. The former is employed to reduce the features and attain better prediction accuracy. The latter is used to derive the significance of the individual variables. Feature importance ranks the features that can provide the best and optimized results by filtering the biases; hence, all other unnecessary noises from the datasets are removed. Regularised Random Forest(RRF) is used for ranking the features using Rattle in R. Among 50 extracted features, only 25 features are selected for further prediction. Figure 3.4 shows the ranking of the features. Features such as object type, database name, writes, etc are considered as least important, while row count, application name and SPID are given utmost importance for predicting data frequency. Once the trace is shaped, various machine learning algorithms are ensembled using MCV for predicting the frequency of the queries, which are validated in the forthcoming subsection using performance parameters.

3.4.4 Validation and Selection of Best Prediction Model

Models such as decision tree, linear and KSVM are implemented and ensembled to predict the frequency of data. To to get the best results, the parameters of the models are tuned and the best model is selected based on accuracy, errors, precision, recall, F-measure and ROC.

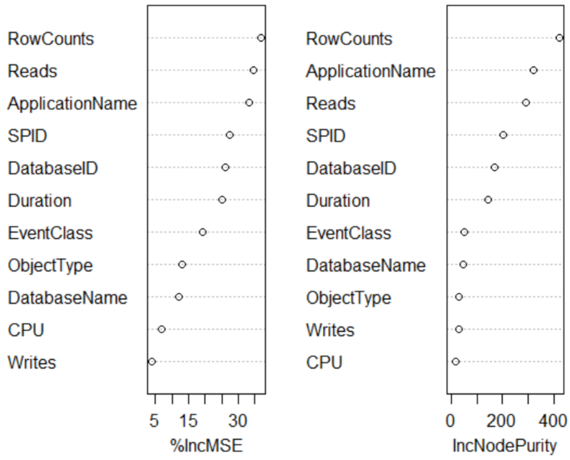


Figure 3.4: Important feature selection using RRF.

- Accuracy and Error Rate

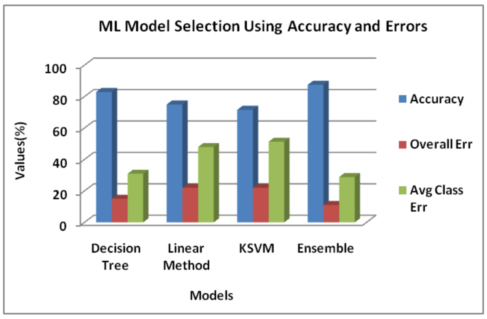
  Results for accuracy, overall error, and average class error have been evaluated in Figure 3.5a. The ensembled model gives the best prediction accuracy of 87.5%. It successfully lowers the average class error and overall error up to 28.81% and 11% respectively. It is followed by the decision tree which yields 82.95% accuracy with 15% and 30.9% overall error and average class error respectively. The linear model provides 75% accuracy with 22% overall and 47.90 % average class error. The overall error for KSVM is the same as that of the linear model but it lagged with 71 % accuracy.

- Precision and Recall

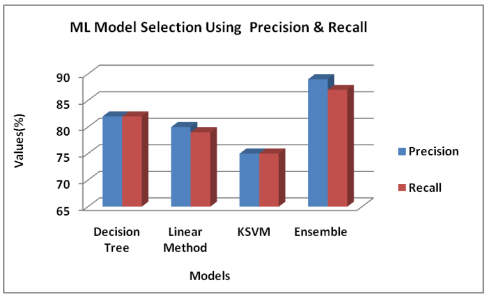
  Figure 3.5b, highlights the precision and recall values for the models. The highest measure of precision possible with the proposed model is 89% with 87% recall; whereas, decision tree provides 82 % precision as well as recall. The linear model closely follows the decision tree with 80% precision and 79 % recall. On the other hand, KSVM generates the lowest values of 75% as precession and recall.

- F-measure and ROC

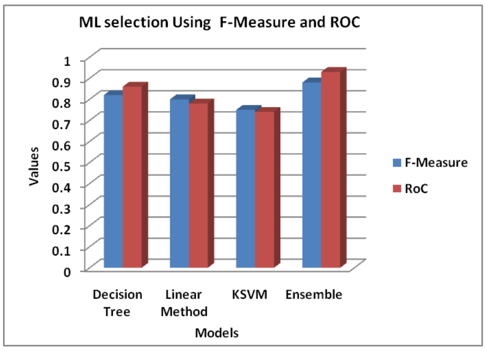
  The selection results for F-measure and ROC are well revealed in Figure 3.5c. It is



(a) Accuracy and error rate



(b) Precision and recall



(c) F-measure and RoC

Figure 3.5: ML model selection using performance parameters

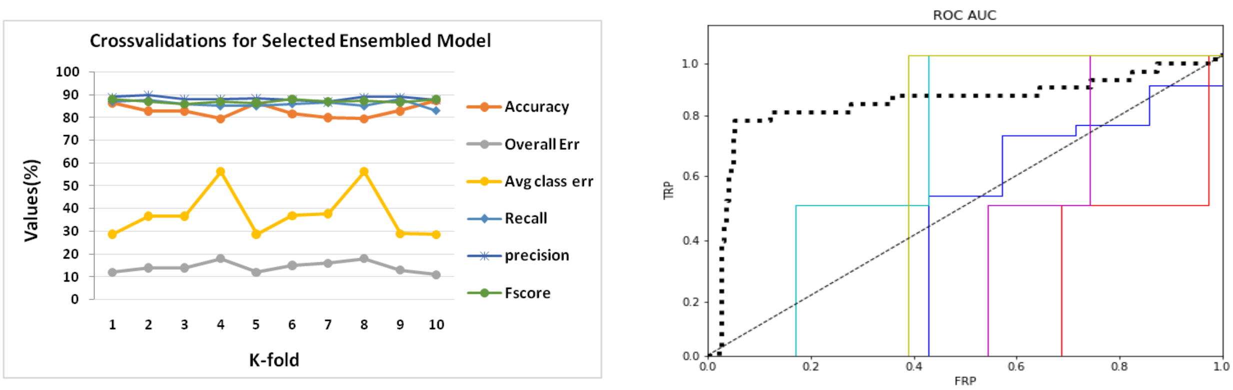
observed that the ensemble model provides 0.88 F-measure which is highest among all the other models. It is followed by a decision tree, linear and KSVM which provides F-measure of 0.82, 0.8, 0.75 respectively. Similarly, ROC for an ensembled model is 0.93. The decision tree ranked second with the ROC of 0.86, whereas the linear model provides 0.78 and KSVM ranked last with 0.74 ROC. Hence, from the Table 3.3, an ensembled model is selected for predicting the frequency of queries such that efficient data placement approach can be applied.

3.4.5 Performance Validation: Selected Ensembled model

Additionally, the performance of the selected ensembled model is validated using K-fold cross-validation with different ratios of training and testing subsets of data. K-fold cross-validation is a process in which k comparisons are performed with random data provided in each fold. The ten-fold cross-validation for accuracy, average class error, and overall error, precision, recall, and F-measure is illustrated in Fig 3.6(a). It reveals minimum of 80% accuracy in each fold with a maximum of 87.5% at 10th fold. Similarly, cross-validation of the overall error provides an average of 14% with a minimum of 11% at the tenth fold. Among all the folds, minimum average class error is 25.81%. Precision and recall are validated with an average of 89% and 87% respectively. The efficiency of the model is summarized with a maximum F-score of 88% at first, seventh and tenth folds. Figure 3.6(b) clearly represents the best prediction capability of the proposed ensembled model with the ROC value of 0.93.

Table 3.3: Model selection criteria considering various performance metrics

Model Name	**Accuracy**	**Overall Error**	**Averaged Class error**	**Recall**	**Precision**	**F-Measure**
Decision Tree	82.95%	15%	30.9%	82%	82%	82%
SVM	71%	22%	51.27%	75%	75%	75%
Linear	75%	22%	47.90%	79%	80%	80%
Proposed Ensembled Model	**87.5%**	**11%**	**28.81%**	**87%**	**89%**	**88%**



(a)

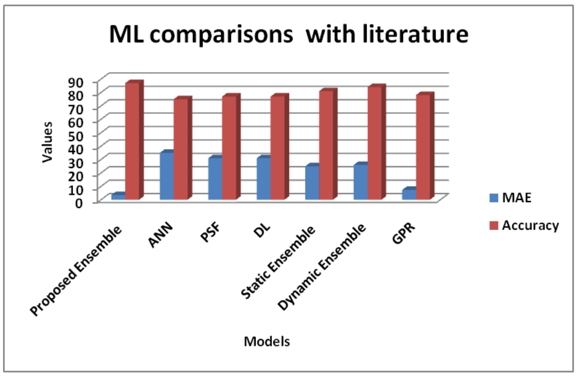
(b)

Figure 3.6: Validation for selected ensembled model: (a) Repeated k-fold cross validation for accuracy, average class error, overall error, precision, recall, F-measure, (b) RoC curve for ensembled model

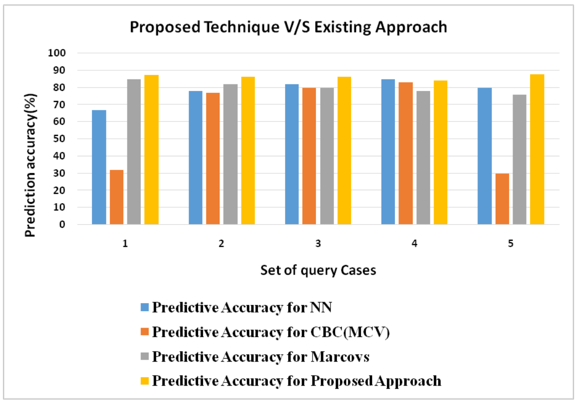
3.5 Comparative Analysis: Proposed Model With Existing Models

The performance of the current approach has been verified using the results of peer researchers [30]. Figure 3.7a corroborates the better performance of the proposed prediction model when compared with those developed by other researchers. Our proposed ensembled model provides the highest accuracy of 87.5% when compared with ANN that yields 75%, Pattern Sequence-based Forecasting (PSF) and Deep Learning (DL), both provide an accuracy of 77%. The static and dynamic ensemble models provide 81% and 84% accuracies, respectively, which are much lesser than that of our model. Also, the optimal ensembled scheme [30] produces 25% and 26% average error, respectively, while our proposed scheme could reduce it to 3.5%. The ANN-based scheme results in 35% error, whereas PSF and DL generate 31%. Moreover, the Gaussian Process Regression (GPR) [118] approach used on the SCATS dataset yields 78.11% accuracy with 7.37% error rate. Their work lags behind our ensembled approach, also, the authors did not utilize their work for reducing energy consumption.

The prediction results are also compared with respect to the number of the queries fired by the users. Figure 3.7b shows comparison results for accuracy on firing 100 to 500 queries. Our proposed model shows 87.5% accuracy when 100 queries are fired. It decreases to 84.09% on executing 400 queries and again it shows its best accuracy of 87.8 % on 500 queries. CBC(MCV) [85], [119], [30] could not work well as they merely yield accuracies of 32% and 30% for executing 100 and 500 queries, respectively. Although Markov [86] could compete well by providing accuracy of 85% on executing 100 queries, it lagged behind the proposed technique in executing 200 to 500 queries with an accuracy of 82%, 80%, 78%, and 75% respectively. While, NN could provide only 67% accuracy to run 100 queries. Hence the selected ensembled model proves to be better in generating most accurate prediction results.



(a) Prediction accuracy



(b) Prediction accuracy wrt number of queries

Figure 3.7: Performance comparisons for machine learning models with existing techniques and datasets

3.6 Conclusion

This chapter discussed the proposed storage prediction model. Memory traces consisting of queries have been generated. The frequency characteristic extracted from the trace has been chosen as a target variable. Various machine learning models have been ensembled to predict data frequency to identify the future requests. The predicted requests have been classified into two categories: popular and unpopular, using the threshold on predicted frequencies. Furthermore, the popular data will placed into the hot disk i.e. SSD cache and the rest in other disks using intelligent energy efficient approach as explained in the further chapter.

Chapter 4

An Intelligent Energy Aware Approach Using Storage Prediction Model

The previous chapter expatiated storage prediction model that uses classified ensemble approach for data frequency prediction. In this chapter, an intelligent energy-aware approach is proposed that uses prediction results to place the data among the disks according to its usage. The main objective of the proposed approach is to integrate data prediction, placement and disk scheduling to achieve energy-efficient data centers.

An intelligent energy aware approach firstly allocates the predicted popular data among hot disks with the replication in cold disks, while the remaining data is placed in cold disks using data placement approach. Then, it takes a stream of requests as an input and intelligently determines the disks that can serve the request in accordance with its state using disk scheduling. Due to the multiple replication, the disk would selected in such a way that the standby disk would not be spun up to execute the request. Also, the maximum remaining idle time and minimum waiting time properties would be conserved for the disk in the active and idle state respectively. Likewise, the frequent spinning of the disk is avoided. The performance of the proposed technique is evaluated in terms of energy consumption and execution time.

4.1 Use of Prediction in Intelligent Energy Aware Approach

Storage has become a prevalent technology for saving information-driven data. Data centers provide large number of disks for storing big data applications. However, the deployment of data raises the storage energy efficiency challenge.

Storage power consumption comprising of disks is maximum among all the other entities present in data centers. According to the recent survey, the energy consumption of data centers is 1.5% of the world energy consumption where 40% of is consumed only by the storage systems [26]. This high energy consumption can be reduced if the popular data is predicted and stored in hot disks that remains active most of the time. Hence, the prediction can be used in efficient data placement among disks that would lead to the efficient disk scheduling, hence save energy and time. Prediction results also helps in identifying which data objects need to be replicated and where.

Although, there is much impedance to research in energy-aware disk storage algorithms. But unfortunately, previous research does not cover all aspects such as data placement including replica management, monitoring of minute disk states, disk idle and waiting for timings [120], [121]. So, there is a dire need to fill this gap and make a contribution by developing an intelligent energy-aware approach.

4.1.1 Objectives of an Intelligent Energy Aware Approach

The main objective of an intelligent energy aware approach in to execute the request with the minimum energy and time consumption.

- It efficiently places and replicates the tagged predicted data in hot and cold disk respectively using data replication. The popular data is concentrated on hot disk with the replication in cold disk and the remaining unpopular is stored in cold disk.

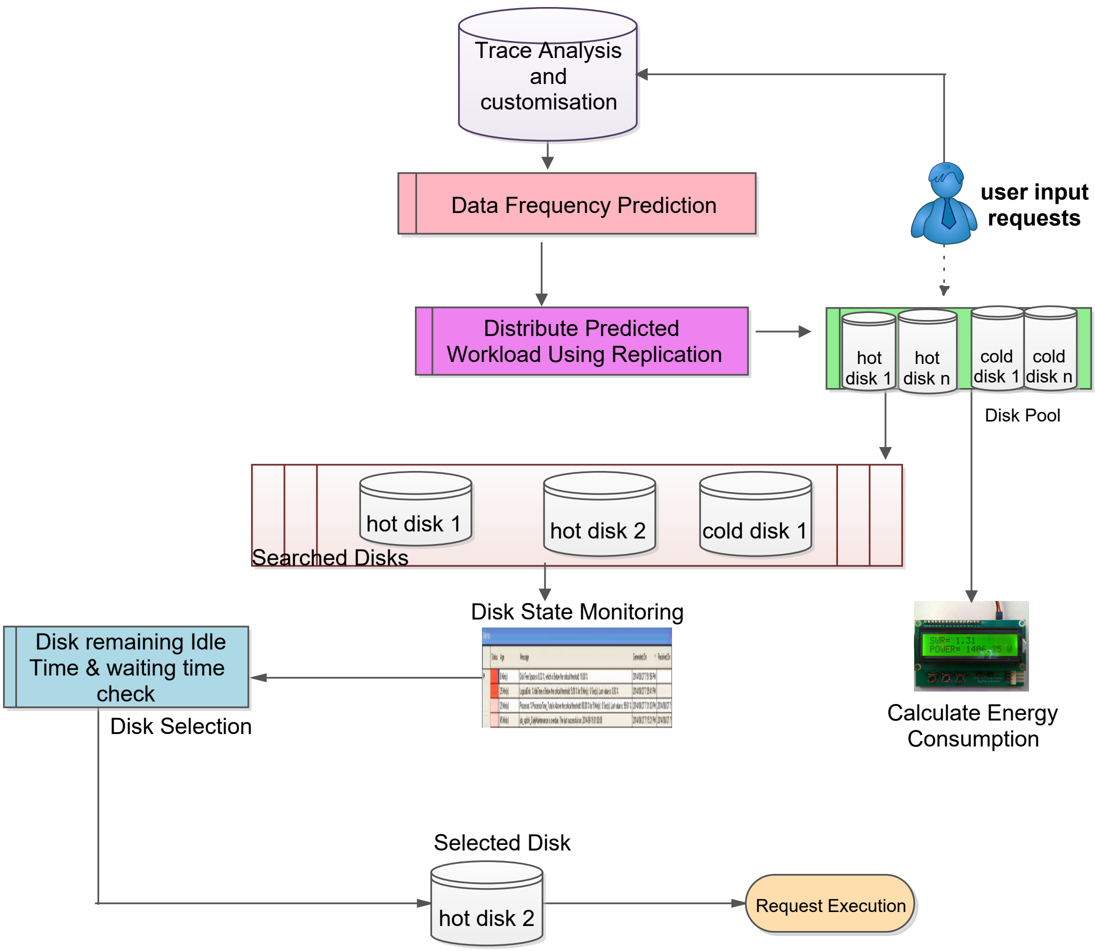


Figure 4.1: Intelligent energy aware framework

- Such a placement reduces a disk spins and makes the scheduling task easy by executing the request through most available and least energy consuming disk.

- The scheduling approach intelligently searches and selects the disk using remaining idle time and wait time threshold policies that results in least energy and time consumption.

Following section explains the methodology of proposed intelligent energy efficient storage framework.

4.2 An Intelligent Energy Aware Storage Framework

An intelligent energy-aware approach has been framed in Figure 4.1. It considers the SQL traces of the queries fired on the real data streams obtained from the SCATS sensors of Dublin city, taken as commercial application. It analysis the trace of an application and tags the popular and unpopular files based on the predicted frequency as described in the previous chapter. Then, the frequently used predicted query requests, also known as popular requests are distributed in hot disk with the replication in cold disk, while the remaining is settled in the cold disks. It also considers partial and temporal locality [122]. This process is defined as data placement approach, which is well explained in subsection 4.2.1 Next, the request is executed by the most favorable and available disk using intelligent search and selection also known as disk scheduling which is well explained in subsection 4.2.2. Disk searching locates all the disks that contain query data for the request whereas disk selection finally selects the disk that would execute the request based on their current states. Data replication allows the scheduler to select a disk among various searched disk. If searched disk is in an active state then the minimum waiting policy would finalize the disk to execute the request. If the searched disks are in idle mode then the disk with maximum reaming idle time will execute the request. In any case, the standby disk would not be disturbed. Finally the overall system performance is measured in terms of energy and time.

4.2.1 Data Placement Approach Using Replica Optimization

The data associated with popular and unpopular queries are distributed in the disks array D= $\{d_m, \ldots, d_n\}$ which are categorized as hot disk D_h= $\{d_1, \ldots, d_e\}$ and cold disk D_c= $\{d_1, \ldots, d_e\}$ where D= $D_h \cap D_c$ and $D_h \cup D_c = \phi$. The hot disk would store the predicted popular while unpopular query data are stored in cold disk. The total number of required disks D_n to save the data is estimated using Eq 4.1. Here, D_{cap}[GB] is the required storage capacity, t depicts the total number of unique files with s_i as the size of

Algorithm 3: Energy efficient data placement

Result: Data placement among the disks

Input : Q, S_i, P, T,

$C_h \leftarrow$ Disk configuration

$R \leftarrow$ Total replication factor

Output: Energy efficient data placement

```
rep_h=R-1, rep_c=R-rep_h
for (i=0;i ≤ Q;i++) do
    if q_i ∈ Q_h then
        for (j=0;j≤R;j++)) do
            if j ≤ rep_h then
                StoreList.add(new MyPowerHarddriveStorage(i,P,S_i,C_h))
            end
            else
                StoreList.add(new MyPowerHarddriveStorage(i,T,S_i,C_h))
        end
    end
    else
        StoreList.add(new MyPowerHarddriveStorage(i,T,S_i,C_h))
end
```

ith query in[GB] using replication factor rep_i.

$$D_n = \frac{\sum_{i=1}^{t} rep_i s_i}{D_{cap}} \tag{4.1}$$

The ratio of the hot and cold disk is calculated as γ using Eq 4.2 and Eq 4.3 where P and T are the total number of assigned hot and cold disks respectively.

$$\gamma = \frac{\sum_{i=1,q_i \in Q_h}^{(1-\theta)*m} rep_i s_i}{\sum_{i=1,q_i \in Q_c}^{\theta*m} rep_i s_i} \tag{4.2}$$

where $\sum_{i=1,q_i \in Q_h}^{(1-\theta)*m} r_{rep} q_{size} = \text{P}$

and $\sum_{i=1,q_i \in Q_c}^{\theta*m} r_{rep} q_{size} = \text{T}$

$$\gamma = \frac{P}{T} \tag{4.3}$$

Finally, the data distribution function $X(P, T, Q_h, Q_c)$ keeps the track of predicted frequency and places them on hot and cold disks using Eq 4.4. The popular data with high frequencies Q_h are laid in the hot disk P as well as in cold disk T with the replication rep_i while remaining data Q_c are stored in T.

$$X(P, T, Q_h, Q_c) =$$

$$\begin{cases} \sum_{i=1}^{P} rep_i q_i \in Q_h; \text{If } q_i \text{ is the } ith \text{ freq used file.} \\ \sum_{i=1}^{P} rep_i q_i + \sum_{i=1}^{T} rep_j q_j \in Q_c; \text{Otherwise.} \end{cases} \tag{4.4}$$

The value of rep_i for hot and cold file is decided using replication criteria as explained in Algorithm 3. Replication factor for hot file rep_h is calculated by taking difference of one from the total replication value R and rest are considered as cold replicas rep_c which is stored in a cloud using MyPowerHardDriveStorage function. Further, all the information is regularly recorded in a log table l which is used for searching and selecting the disk that can execute the request using intelligent disk scheduling.

4.2.2 Disk Scheduling

Data replicated in multiple disks requires an intelligent energy-efficient scheduling such that the most available disk that consumes least energy can be selected for executing the request. It also helps in reducing disk spins and execution time. The energy-efficient scheduling is performed in two steps that include disk searching and selecting as described in subsequent subsections.

4.2.2.1 Disk Searching and Selecting

Disk searching means locating all the disks that contains the data associated with the current request. It is also known as disk-data mapping using function $D(r_i, j, n, k)$ as shown in Eq 4.5. Among the searched disk D_j, the most efficient disk D_k is selected

Algorithm 4: Disk scheduling algorithm for request execution

Result: Selected energy efficient disk D_K
Input : $r_i \leftarrow$ Request
$D_I \leftarrow$ Disk idle time
$T_I \leftarrow$ Idle time threshold
Output: D_K

while(l.Next)
if $r_i == L_{data}$ **then**
    fetch(d_j)
end
else
    disk not found
for *(i=0;i¡=*d_j*;i++)* **do**
    switch D_j **do**
        case $state_{d_j}$==*active*, $state_{d_{j+1}}$==*active*, *Wait*(D_j) $\geq$ (D_{j+1}) **do**
            $d_k = d_{j+1}$
            d_{j+1} is selected to excute the request
        case $state_{d_j}$==*idle*, $state_{d_{j+1}}$==*idle*
        RI(D_j) $\geq$ *RI*(D_{j+1}) **do**

        $d_k = d_j$
        d_j is selected to excute the request
        break
        case $state_{d_j}$==*active*, $state_{d_{j+1}}$==*idle* **do**

        $d_k = d_{j+1}$
        d_{j+1} is selected to excute the request
        break
        case $state_{d_j}$==*standby*, $state_{d_{j+1}}$==*idle* **do**

        $d_k = d_{j+1}$
        d_{j+1} is selected to excute the request
        break
        case $stated_j$==*standby*, $stated_{j+1}$==*active* **do** $d_k = d_{j+1}$
        d_{j+1} is selected to execute the request
        break

    end
end

using various policies such as remaining idle time and waiting time. Remaining idle time would be applied when all the searched disks are idle state. Selecting the disk with the maximum remaining idle time RI_i would provide advantage to the disk which is nearer to the standby state. Here the idle time is compared with the threshold which is calculated by averaging the idle values from the trace (Refer Eq 3.2). Further, in case the searched diks are in active state then the disk with the minimum waiting time W_t would accelerate the request execution time. As shown in Algorithm 4, whenever the user submits the request, all the disks containing the data associated with the request are searched using l. Then the best and energy efficient disk is selected to execute the request. All the $Dj's$ in standby state should be given least priority to execute the request that would save the disk spins.

$$\begin{aligned} D(r_i, j, n, k) = D_k \ s.t \ r_i \rightarrow D_j \in (P \cup Q) \\ \cap \, minW_t \ \{|D_j|\}_{j=1}^{n} \\ \cap \, maxI_t\{|D_j|\}_{j=1}^{n}, \\ where \ \{j, \ldots, m\}; n \geq m \end{aligned} \tag{4.5}$$

Thus, it helps us in keeping the maximum disks in the standby mode. Also, the disk with minimum remaining idle time can go to the standby state and wait time for the current request can be reduced that results in energy savings as modeled below.

4.3 Mathematical Formulation for Measuring Reduced Energy Consumption

In general, the user submits the request which is modeled as req= $\{r_i, r_j\}$ where r_j is the successor request of the current request r_i. After receiving requests, the appropriate disks that contain the request are searched and selected using scheduling algorithm. As soon as a scheduling decision is taken its energy consumption is noted. The energy consumption is represented as an input $E(i, j, D())$ refer (Eq 4.6), where i and j are the current and successor request from the set of req which can belong to Q_h or Q_c. $D()$ is the final selected disk that would execute the request. The total energy consumed is calculated using equation 4.6. Here the first part is the primary energy consumed while placing E_{place} the data to the multiple disks which is one-time energy consumption. The δ energy consumed during searching E_{Search} and selecting E_{Select} the most appropriate disk D_k would be added additionally in each iteration whenever the request is encountered as shown in Eq 4.7. The energy consumed by the selected disk D_k is measured in all the states (active, idle or standby) and spins as seen in Eq 4.9. Eq 4.10 shows the power consumed by these states as P_{active}, P_{idle}, $P_{standby}$. The disk can stay in these states for the duration t_m, t_j, t_k in $n1^{'}$, $n2^{'}$ and $n3^{'}$ number of times respectively. Eq 4.11 shows the energy consumed while the D_k is in active disk where tbr_i is the burst time of the current request. While the disk is idle, ta_l shows the arrival time of current request and te_{l-1} depicts the end time of previous request. Further, the disk energy is consumed while spinning up and down. The power to spin either from idle to standby or from standby to active is represented in Eq 4.12 as P_{is} in time t_{jk} and P_{sa} in time t_{ki} respectively. Finally, the disk after completing the execution is placed in standby mode. Idle time productivity is considered while putting a disk into a standby state. If it has a request in nearby, then it is remained in idle mode rather than spinning up and down to execute the consecutive requests. Algorithm 5 sets the idle time threshold TI_i such that disk exceeding it would be allowed to spin down to avoid random and frequent spins.

$$E(i, j, D()) = E_{place} + E_{Search} + E_{Select} + E_{Dk} \tag{4.6}$$

$$E_{Search} + E_{Select} = \sum_{t=1}^{n} E_i = \delta \tag{4.7}$$

$$= P * (\sum_{t=1}^{n} t_{Search} + \sum_{t=1}^{n} t_{Select}) \tag{4.8}$$

$$E_{Dk} = E_{State} + E_{Spin} \tag{4.9}$$

$$Where,\ E_{state} = \sum_{m=1}^{n1} P_{active} t_m + \sum_{l=1}^{n2} P_{idle} t_j + \sum_{k=1}^{n3} P_{standby} t_k \tag{4.10}$$

$$\begin{gathered} E_{state} = P_{active} \sum_{l=1}^{n1} tbr_i * f_{size} + \\ P_{idle} \sum_{l=1}^{n2} |te_{l-1} - ta_e| + P_{standby} \sum_{k=1}^{n3} t_k \end{gathered} \tag{4.11}$$

$$and\ E_{Spin} = \sum_{lk=1}^{N1} P_{is} t_{jk} + \sum_{km=1}^{N2} P_{sa} t_{ki} \tag{4.12}$$

$$E_{Spin} = P_{is} \sum_{jk=1}^{N1} t_{jk} + P_{sa} \sum_{ki=1}^{N2} t_{ki}$$

4.4 Evaluation Metrics: Intelligent Energy Aware Approach

This section, outlines the performance metrics for evaluating the proposed energy aware approach. The performance metrics includes response time and energy consumption as explained below:

Algorithm 5: Energy Efficiency Algorithm

Result: Reduced E_{total}
Input : $D_I \leftarrow$ Disk idle time
$T_I \leftarrow$ Idle time threshold
Output: E_{Total}

```
1 Execute D_place and calculate E_Search and E_Select from Eq 4.8
2 Execute r_i from d_k and calculate E(D()) from Eq 4.9
3 Calculate E_Total using Eq 4.6
4 if D_I ≥ TI_i then
5 |  Spin the disk down
6 end
7 If r_j arrives go to step 2
8 Repeat
```

- Response time

It is a execution time taken to accomplish requests submitted by the users. It is defined as t_{exe} in Eq 4.13. The average response time for executing the request from the disk should be least.

$$t_{exe} = t_{search} + t_{select} + t_{wt} + t_{ActiveEndT} \tag{4.13}$$

$$t_{ActiveEndT} = t_{active} + t_{idle} + t_{spin} \tag{4.14}$$

$$t_{wt} = t_{ActiveEndT} - t_{currtime} \tag{4.15}$$

Here a t_{wt} is the waiting time and $t_{ActiveEndT}$ is the time when disks are executing the requests as seen in Eq 4.14 and Eq 4.15. Data replication has been used to reduce this response time. Availability of data in multiple disks significantly decreases the response time that greatly improves overall system performance.

- Energy consumption

  The energy consumed by the disk while serving the requests is defined as the product of power P_{exe} with time t_{exe}. Total energy consumed while executing the request in stated in Eq 4.16.

$$E_{Exe} = [(t_{exe} * F_{size}] * [(P_{exe})] where \tag{4.16}$$

These performance matrices are measured on examining the impacts of four important workload parameters which include disk ratio, request size, and replica factor as described below.

- Request Size

  The request size may be the size of the query data in bytes. The increase in the size may increase the execution time, hence energy consumption.

- Replica factor

  It is defined the number of copies of particular objects which are saved in multiple disks at a time. An increase in replica may reduce the disks delay due to the easy availability of data. It also reduces the number of disk spins. But, the greater replication factor leads to disk pollution and redundancy.

- Disk ratio

  It is the ratio of number of hot and cold disks in the set of disk. It is calculated in Eq 4.3, where, γ has great influence on the disk energy consumption and response time. The value of γ may vary with the application. Therefore, it should be adjusted to maintain a balance between energy saving and response time.

4.5 Experimental Results and Analysis

An intelligent energy aware approach has been implemented in a real-time environment. The configuration of the server includes Intel Xeon e5 2650 2.6 GHz and RAM -DDR3 8GB 1600 MHz. Partial RAID 1 [97] has been configured on KINGSTON SA400S37120G, SATA 500 GB with 7200 RPM and 250 GB Seagate Barracuda 7200.12, with the specifications as shown in Tables 4.1 and 4.2, respectively. The system has been installed with server 2008r2. SQL Server profiler has been used to generate the traces from the SQL queries fired on the SCAT data streams of Dublin city, in Ireland. These traces have been further customized using the SQL server analyzer. Disk states and spins have been productively automated using Smartctl commands along with shell scripting. Fluke 325, 115 True Root Mean Square (RMS) Clamp Meter and Multimeter have been employed to calculate the power using the current and voltage readings, as shown in Figure 4.2. The results have been evaluated as per the phases employed in the proposed framework. The data placement policy has been validated based on the energy consumed, time factor and cache hits. The stability in disk idle time has been analyzed using the disk idle time threshold. It is followed by a energy and time measurement while executing the requests.

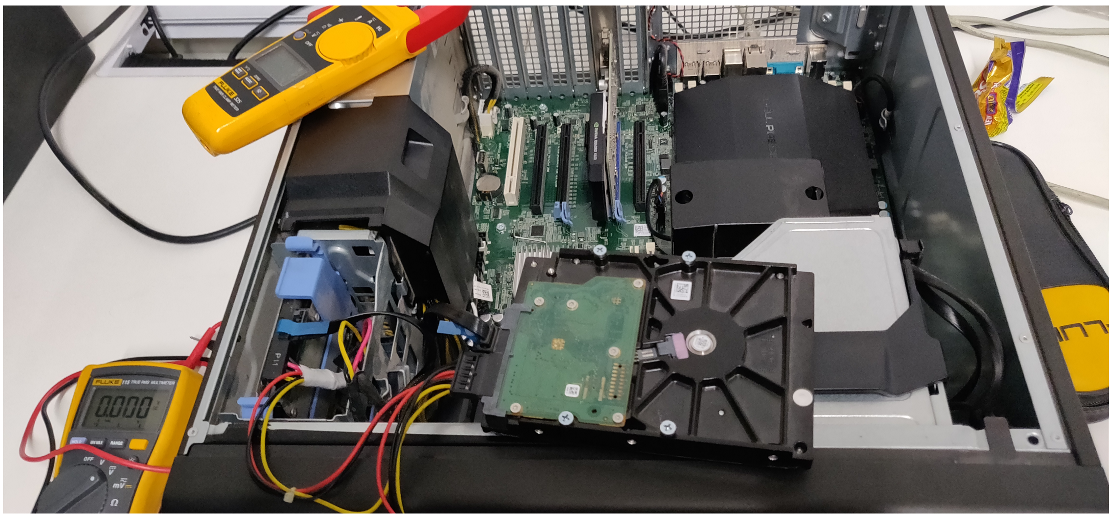


Figure 4.2: Experimental testbed.

Table 4.1: Disks configuration

Features	**Disk 1**	**Other Disks**	**SSD**
Interface	SATA 6Gb/s	SATA 3Gb/s NCQ	S-ATA Gen3, 6Gbps
Hard Disk Model ID	ST500DM002-1BD142	ST3250318AS	KINGSTON SA400S37120G
Firmware Revision	RSM7	X4120006	SBFKB1D1
Hard Disk Serial Number	ZDE5KRT8	174179804281	50026B76831AB6E8

Table 4.2: Disk states and power consumption

<table>
<tr><th>Description</th><th>Value in W(HDD)</th><th>Value in Watt(SSD)</th></tr>
<tr><td>Idle</td><td>≤ 5 W</td><td rowspan="5">0.195 idle/
0.27 w Avg/0.64(max) read
/1.35w (max)write</td></tr>
<tr><td>Active</td><td>≤ 8 W</td></tr>
<tr><td>Standby Power</td><td>0.4 W</td></tr>
<tr><td>Spin-up-Power</td><td>14.63 W</td></tr>
<tr><td>Spin Down</td><td>1.83W</td></tr>
</table>

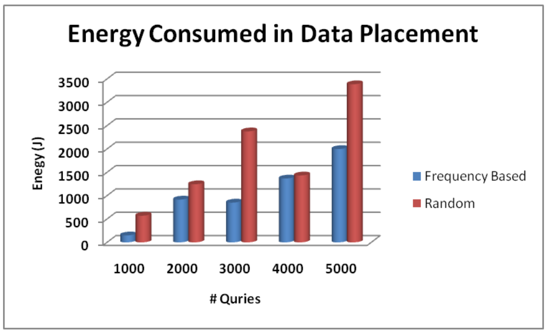
The Experimental setup is categorized into the following phases.

- Data Placement Validation
- Disk Idle State Validation
- Energy Reduction Validation for Request Execution
- Comparisons

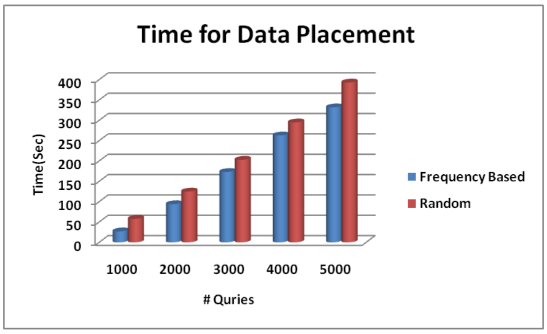
4.5.1 Validation of Proposed Approach: Data Placement Technique

The main motive of the data placement approach is to place the maximum used predicted data in the hot disks(SSD cache) and the remaining in HDD. Figure 4.3 illustrates energy and time consumed while allocating the data to the multiple disks. These parametric values are determined by changing the number of user's queries, which ranges between 1000 and 5000.

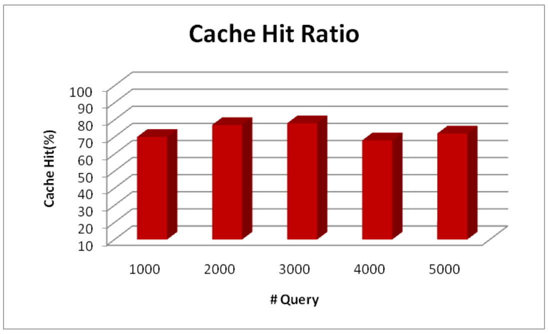
The energy consumption during data allocation is validated in Figure 4.3a [107]. The total energy consumption is significantly decreased on applying the proposed data placement algorithm. The frequency-based data distribution takes a maximum of 1999 J for executing 5000 queries, which is increased to 3387 J on randomly distributing the data.



(a) Energy consumption



(b) Query execution time



(c) Cache hits

Figure 4.3: Data placement validations

The time taken for buffering of the frequently used predicted data in the hot disk is shown in Figure 4.3b. Although, the execution time escalates with an increase in the number of queries, still it takes only an average of 177 sec to execute the queries on using the proposed placement algorithm. This is much lower when compared with the 214 sec consumed when the data is distributed randomly.

Furthermore, the data placement results are also validated through the high cache hits. Most of the data accesses are satisfied by the hot disks that provides 78.5% average cache hit ratio, as shown in Figure 4.3c. It confirms that only 20% of the data is utilized most of the time by the user. Only, a new query fired by the user causes a cache miss as notified in the case of 4000 queries, yet lead to 70% of cache hits. Hence, an efficient data placement approach enables the other HDDs to remain stable in low power state for a much longer duration [123]. It elongates the idle period of the disk inorder to reduce the frequent disk spins. The following sections verifies the stability in the state of disk idle time monitoring.

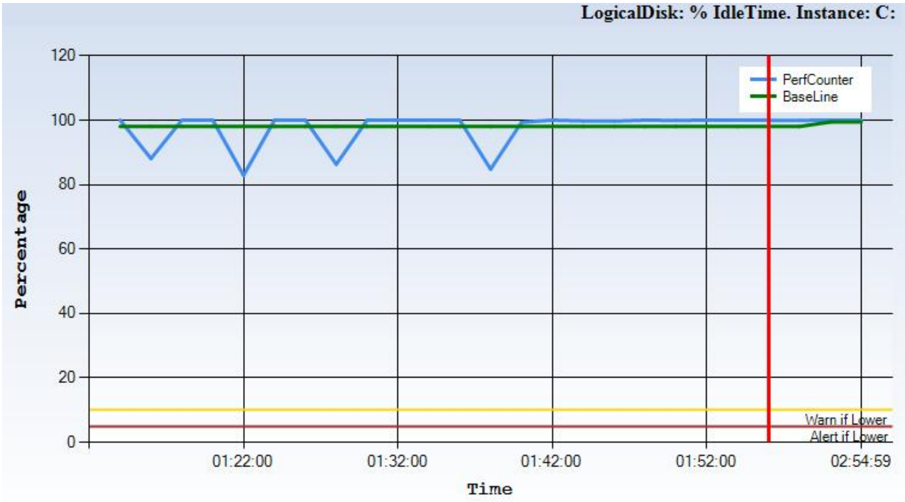


Figure 4.4: Disk idle state analysis using analyser monitor

4.5.2 Validation of Proposed Approach: Disk Idle State Monitoring

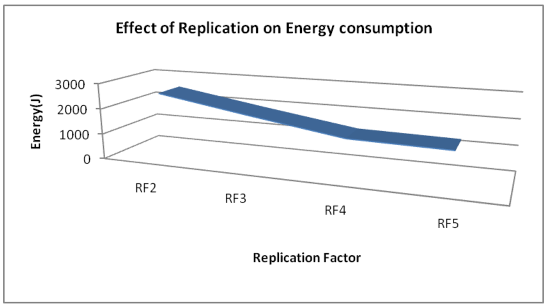
The efficient data placement also elongates the idle state of the disk by concentrating the request on the set of hot disks. Thus, reduces the shorter stay of the disk in standby mode even in case of frequent requests. To check the stability in the state, the predefined idle time threshold is fed to the SQL check-up monitor that displays a record of a disk in an idle state and keeps the disk in idle mode until it crosses the that threshold using SmartCtl commands. Figure 4.4 shows the percentage of idle state encountered encountered by the disk with respect to time. Initially, the disk is continuously spinning from 1:22:00 to 01:42:00 as the energy-efficient operation is yet to be applied. Now on applying efficient placement approach, the disk stays in an idle state with out spins as shown in the time slot 01:42:00-01:52:00. Such long duration of the idle state in case of frequent requests decreases the spinning rate, which reduces the sudden power shoots, hence results in maximum energy savings.

4.5.3 Validation of Proposed Approach: Energy Reduction

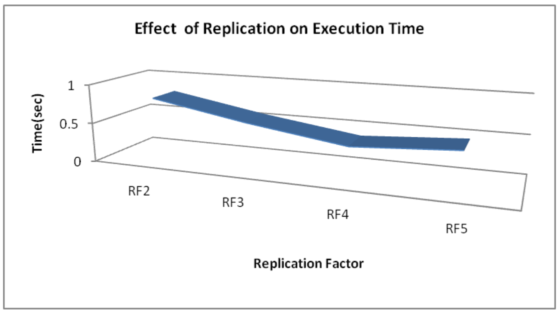
The user inputs request to the disk which successfully executes the query. Energy consumption and execution time have been measured while completing the whole process. The impact of file size, replication number, data size and disk ratio are examined on energy and time reduction.

(a) **Impact of Data Replication**

Energy and time with respect to the data replication has been measured. Data associated with the popular queries are replicated in hot and cold disks that results in reduced energy consumption. As reported in Figure 4.5a, the replicated frequently accessed data reduces energy from 2541 J at replica 2 to 1331 J on reaching replica 5. The decrease in reduction is seen until the replication reaches the saturation level. The energy consumed in replica 4 and 5 do not show much difference by consuming 1397.12 J and 1331.12 J respectively.



(a) Energy consumption w.r.t to replication



(b) Execution time w.r.t to replication

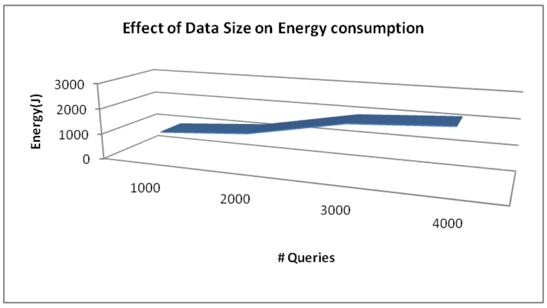
Figure 4.5: Impact of data replication

Another important advantage for data replication is the reduced response time for data access. Allocating and maintaining the predicted popular data in hot disks significantly decrease access time. As seen in Figure 4.5b, the time taken to execute the queries reduces from 0.8 sec to 0.4 sec as the replica factor is increased from

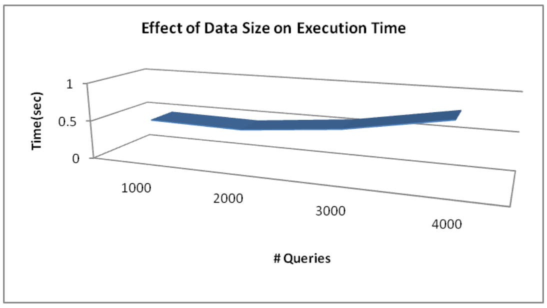
2 to replica 4. However, the replication factor should be selected carefully as the excessive replication may increase the associated costs. There is no reduction in energy and time when the replication factor is increased to 5. Hence, the obtained results indicate that the proposed replication always keeps the energy and response time within the boundaries, provided the replica factor should be well optimized. Therefore only the frequently accessed data objects should be replicated to reduces the total number of replicas. The following subsection reports the effect of data size on energy and time inorder to execute the request.

(b) **Impact of Data Size**

This section represents the energy consumption when the number of queries are varied from 1000 to 4000. Figure 4.6a reports the little increase in energy consumption when 1000 and 2000 queries are fired by consuming 968J and 1245 J respectively. This is possible due to repeated usage of same queries along with an easy availability of data locations caused by the efficient data placement using replication. Howsoever, the energy is linearly scaled and is hiked to 2175 J incase of 4000 queries.



(a) Energy consumption w.r.t to request size



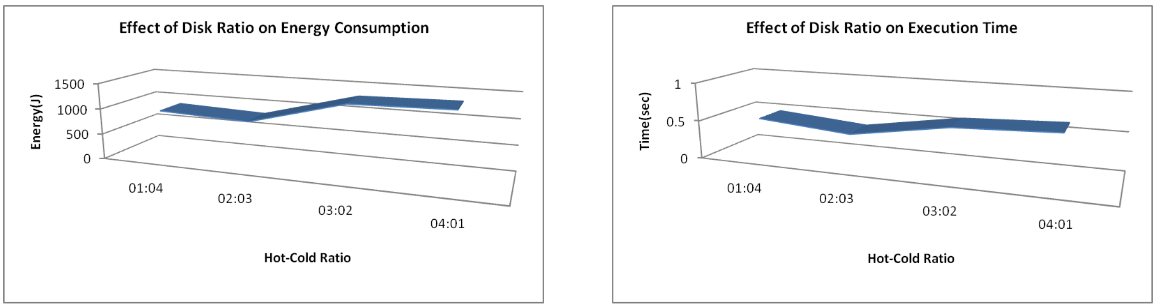
(b) Execution time w.r.t to the request size

Figure 4.6: Impact of request size

Further, the delay in the execution is also seen with an increase in the number of queries (see Figure 4.6b). It consumes 0.48 sec to execute 1000 queries, which is scaled to 0.8 sec while executing 5000 queries. However, executing 2000 queries does not show large difference due to less delay caused by the least frequent spins encountered. This is possible due to timely buffering of popular predicted data in SSD cache.

(c) **Impact of Disk Ratio**

Unlike in the case of data replication, the energy consumption with the different disk ratio is less sensitive. It is mainly due to the fact that hot disk mostly remains active while cold stays in standby state for maximum time. To better understand the impact of varying disk ratio on energy and time consumption, the number of queries are set to 3000. Figure 4.7a reports the obtained energy consumption levels with the disk ratio varying from 1:4 and vice versa. 1:4 consumes 917 J because less number of hot disk causes cold disk to spin frequently, whereas 2:3 maintains the equilibrium between the disks and reduces energy consumption to 852 J. The similar scenario is seen for the execution time (Refer Figure 4.7b). It is decreased to 0.4 seconds at 2:3 and again incremented to 0.65 second in case of 4:1 disks due to an increment in the number of hot disks.



(a) Energy consumption w.r.t to the disk ratio (b) Execution time w.r.t to the disk ratio

Figure 4.7: Impact of γ

So, devising to keep the maximum disks in standby mode allows the substantial energy and time savings. Further,the superiority of the proposed approach has been tested on comparing and contrasting with the other approach as presented below

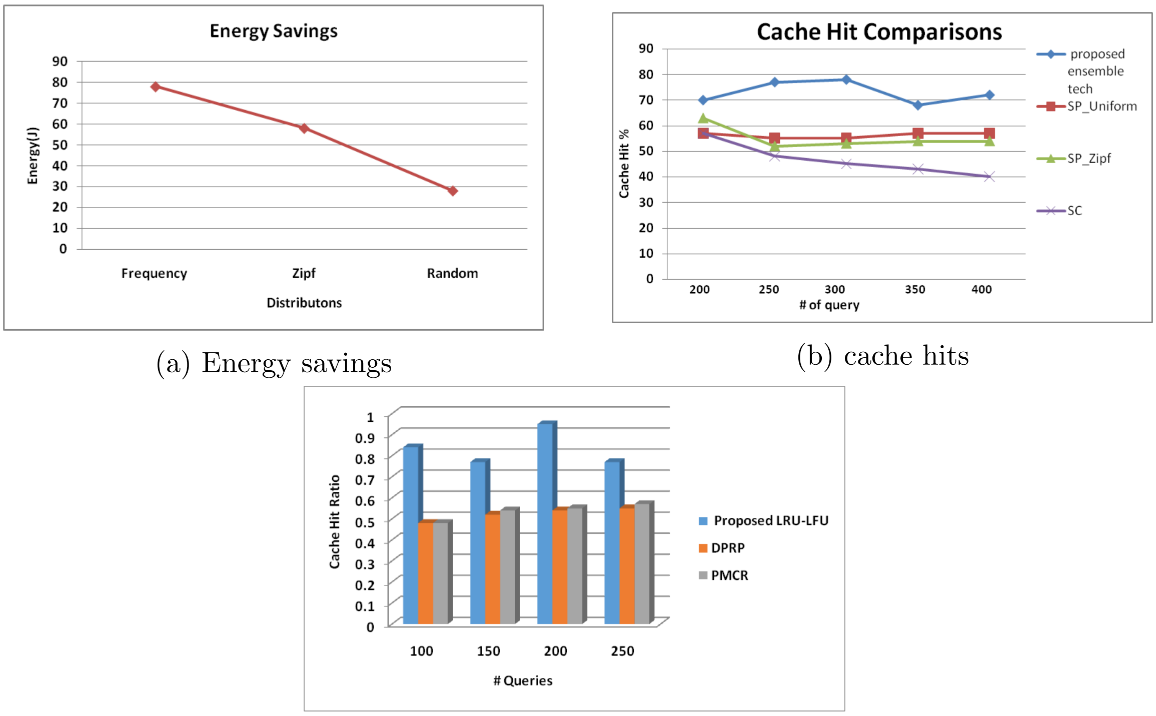
4.6 Comparative Analysis: Proposed Approach With Existing Approaches

The results of the proposed approach have been compared with the reference benchmarks as well as the existing literature.

4.6.1 Comparative Analysis: Data Placement Approach

Efficiently placed data in accordance with its usage frames the base to obtain maximum energy savings. The proposed approach takes into account the data placement based on the frequency. This is the reason for our energy-efficient technique to save maximum energy. The energy-saving results are compared with the zipf and random distribution used in [98]. According to Figure 4.8a, the results obtained through frequency-based distribution saves the maximum energy with an average of 78 %, whereas Zipf and random distribution could save 55 % and 25 % respectively. This confirms that efficiently replicating and segregating both the data and disks can reduce energy consumption as well as improves the performance.

Data placement is also compared through the cache hit ratio, i.e. the number of times the data hits the cache disk. Figure 4.8b graphically compares the average cache hit ratio using various schemes available in literature by taking 400 queries at random. When the data is prefetched using uniform distribution and Zipf distribution (SP_Uniform, Sp_Zipf, and Sc)[124],[125], the average cache hits are calculated as 56.2 %, 55% and 46.6%, respectively. Our present technique outperforms the others by providing an average cache hit ratio of 78.5%. Further, Figure 4.8c shows the cache hit comparison with various replacement policies used in the literature such as Distance-based Predicted Region Policy (DPRP) and Predictable Markov based Cache Replacement (PMCR). Our hybrid LRU-LFU provides the highest cache hit ratio for all range of queries as compared to other techniques. It provides the best value of 0.95 for 200 queries and yields a 0.77 hit ratio



(a) Energy savings

(b) cache hits

(c) Cache hits wrt replacement techniques

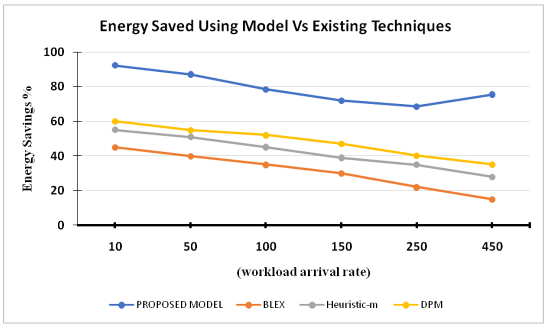
Figure 4.8: Data placement comparisons with existing literature

for 150 and 250 queries. While PMCR and DPRP provide 0.48 cache hits for 100 queries. PMRC could perform a little bit better as compared to DPRP when the number of queries are raised. But with the highest hit ratio of 0.57, it could not overcome our proposed technique.

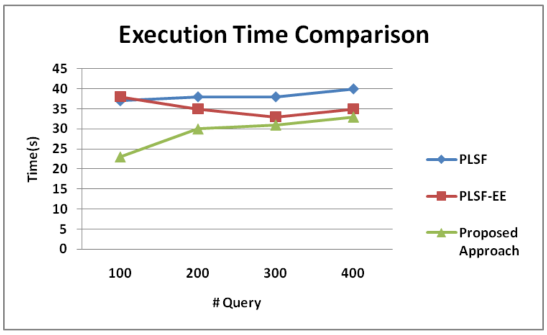
4.6.2 Comparative Analysis: Energy Reduction

The trendline in Figure 4.9a depicts the energy savings achieved with different workloads under various schemes. Our proposed technique could save up to 90% energy under the least workload and 75% at the highest given workload when compared with existing techniques such as Dynamic Power Management (DPM), Heuristic-m, and Block Exchange (BlEX) [103]. Passive disk spins are the main causes for the maximum energy savings. The existing techniques such as DPM, BLEX, and Heuristic-m could only save the average energy up to 65%, 33% and 38.6%, respectively, due to the active disk spin-ups.

Further, the response time penalty has also been reduced due to efficient buffering of the



(a) Energy savings



(b) Execution time

Figure 4.9: Energy and time comparisons with the existing techniques

predicted popular data in hot disk that remains ready to execute the query. It saves the time taken to spin up the disk inorder to execute the request, Figure 4.9b compares the execution time taken by our intelligent approach with the Reliable Energy-Efficient Storage System (RESS) and Modified Parallel Log-Structured File System (PLFS) proposed in [126]. The results demonstrate that the current approach performs much better for 100 and 200 queries by taking 23 and 30 sec, respectively. PLSF takes the maximum execution time of 40 sec. REES (PLSF-EE) performs equally well in executing 300 and 400 concurrent outputs, which takes approximately 30 sec to complete. The results specify that an intelligent energy aware approach can significantly decrease the number of disk accesses, that saves the energy and improves the performance.

4.7 Conclusion

In this chapter an intelligent energy-aware approach has been proposed that allocates the data among the disks according to its predicted frequency. Besides the prediction and placement, it also combines the merits of intelligent scheduling resulting in the reduced energy and time consumption. On the arrival of the request, the most suitable disks are searched and selected using disk state monitoring such that the disk in the standby state would not be used. It follows least waiting time and maximum remaining idle time selection criteria incase all the searched disks are in active and idle state respectively. The proposed approach has been validated in real time environment resulting in maximum energy time reduction.

Chapter 5

An Intelligent Energy Aware Approach for Cloud Data Centers: A Case Study on OLTP Applications

This chapter verifies the effective working of the whole framework using OLTP commercial applications benchmarked with financial and websearch traces. The entire framework is implemented in the cloud environment by integrating storage capabilities into it. The results of both prediction as well as energy aware approach are evaluated using respective evaluation metrics. The working of the proposed approach along with the dataset is described well. Moreover, the implementation of the method is highlighted in two stages.

Firstly, the storage prediction model using classified ensembled approach is evaluated by analyzing the traces of an applications and predicting the frequency of data files. The files are tagged as popular and unpopular using predicted threshold frequency. The prediction results are validated using accuracy, error rate, precision, recall, and F-measure. Secondly, the predicted frequencies of files are used in an intelligent energy aware approach to place them in disk according to its predicted popularity. It correspondingly schedules the request to disk that would consume least energy. The results are evaluated using energy efficiency and execution time in the cloud environment. The optimal replica factor and disk ratio are well recognised for each application.

5.1 OnLine Transaction Processing - Case Study as Big Data Applications

OLTP, is a class of information systems that manages the transaction oriented applications, specifically data entry or retrieval transaction processing. OLTP application involves good amount of both reads and writes. Such applications are composed of a series of files or a single large file with a specific format stored in a disk. The broader analysis and prediction along with energy efficient techniques are needed to be employed to process such applications in cloud data centers.

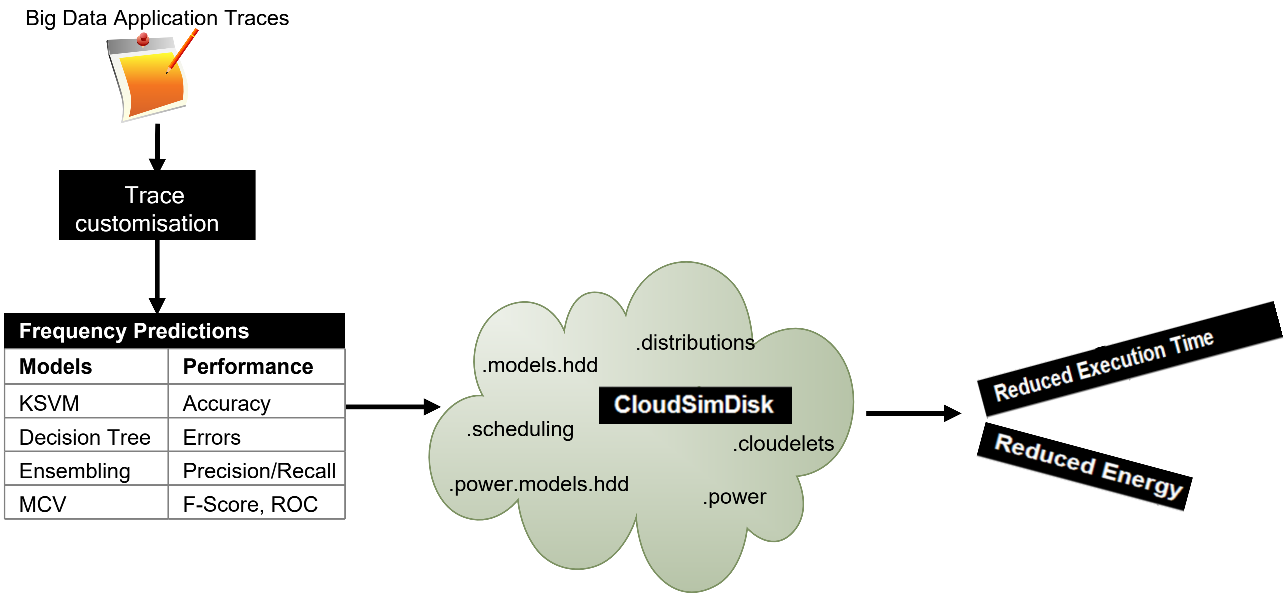


Figure 5.1: Energy aware approach in cloud environment

An seen in Figure 5.1, the traces of a few OLTP commercial applications such as financial1, financial2, websearch1, and websearch2 are collected. The characteristics of the trace include I/O commands that include Logical Block Address (LBA), Application-Specific Unit (ASU), size, opcode, timestamp. Files in a trace are associated with a limited number of ASU. Therefore, the frequency of each ASU value is calculated to predict and classify the popular and unpopular files using storage prediction model. The best prediction model is selected using RMSE, accuracy, precision, recall, F-score and ROC. Further intelligent energy aware approach is executed where, the files associated with a predicted ASU frequency greater than threshold frequency are replicated in hot and cold disks and rest are placed only in the cold disks. On the arrival of the requests, the disks are searched and selected in accordance with their states using idle and waiting time threshold. Accordingly, the energy and time consumption is calculated by selecting the most suitable disk that can succeed to save power. The whole framework in validated in cloud environment by integrating storage capabilities into the cloud platform.

5.1.1 Traces of OLTP Applications

The trace used in a current work, keeps the record of the files associated with financial and websearch applications [1]. Traces along with their characteristics are shown below

- Financial Traces

  Financial traces are extracted from the OLTP applications running at two financial institutions as available by the Storage Performance Council(SPC). Records in the trace file represent I/O commands like LBA, ASU, size, opcode, timestamp refer (Table.5.1).

- Websearch Traces

  Websearch 1, websearch 2 and websearch 3 traces are recorded from a famous search engine. Websearch bears the same characteristics of financial traces.

5.2 Experimental Results and Analysis

This section focuses on the experiments and simulated results in a cloud environment. The simulations are implemented on an Intel(R) Core(TM)i7-8550U CPU 1.80GHZ 2.00 GHz processor with 8 GB RAM on the 64-bit operating system. CloudSimDisk that extends

[1] `http://traces.cs.umass.edu/index.php/storage/storage/`

Table 5.1: Characteristics of OLTP traces

Characteristics	**Abbrevation**	**Definition**
Application specific unit	ASU	The ASU is a positive integer associated with an application
Logical block address	LBA	The LBA describes the ASU block offset of the data transfer in interger format
Size	S	The size describes the number of bytes of the file.
Opcode	o	Opcode can Read or Write
Timestamp	t	The timestamp represents teh time in seconds

Table 5.2: Disk state and power consumption

<table>
<tr><th>Description</th><th>Value in W(HDD)</th><th>Value in W(SSD)</th></tr>
<tr><td>Idle</td><td>3 W</td><td rowspan="5">.06 w idle/
2.5 w active</td></tr>
<tr><td>Active</td><td>5.8 W</td></tr>
<tr><td>Standby Power</td><td>0.4 W</td></tr>
<tr><td>Spin-up-Power</td><td>4.63 W</td></tr>
<tr><td>Spin Down</td><td>1.83W</td></tr>
<tr><td>Manufacturer</td><td>HGST Western Digital</td><td>Intel X18-M G1</td></tr>
<tr><td>Model Number</td><td>HUC109090CSS600</td><td>SSDSA1MH080G1</td></tr>
<tr><td>capacity (MB)</td><td>2100000000</td><td>2100000000</td></tr>
<tr><td>Average Rotation Latency (s)</td><td>0.003</td><td>0.0003</td></tr>
<tr><td>Average Seek Time (s)</td><td>0.004</td><td>0.0004</td></tr>
<tr><td>Maximum Internal Data Transfer Rate (MB/s)</td><td>198.0</td><td>198.0</td></tr>
</table>

CloudSim Toolkit 4.0 integrates the HDD model and HDD power model using DiskSim 4.0 [108]. Intel X18-M G1 SSD and HGST Western Digital HDD are configured to store 22 GB of OLTP applications whose configurations are presented in Table 5.2.

The experimental results of the proposed work has been verified and categorized into two sections:

- Data Frequency Based Storage Prediction Model
- An Intelligent Energy Aware Approach Using Storage Prediction Model

  Further, the impact of disk ratio, replica factor, request size on energy consumption, and time has been investigated. The most optimized replica, disk ratio for each application has been selected as explained below.

5.2.1 Results: Data Frequency Based Storage Prediction Model

In this segment, an assessment of the proposed storage prediction approach is represented for "OLTP" applications. The performance of machine learning classification models is evaluated using accuracy, errors, precision, recall, and F-score where an ensemble model is selected.

5.2.1.1 Validation and Selection of the Best Prediction Model

The performance of the ensemble model is seen to be better than the individual models as seen in Figure 5.2. These individual models are used as a baseline scheme for performance evaluation.

- Accuracy

  An ensembled model leads to the best prediction accuracy of 0.89. It is followed by a decision tree that provides an accuracy of 0.82. The accuracy generated by KSVM reaches to 0.71, whereas the linear model could yield only 0.66.

- Error Rate

  Ensemble model successfully reduces the error rate by up to 0.13, which is impossible for the individual model to attain. The decision tree generates 0.23 error and KSVM yields 0.28, whereas 0.34 is seen as the highest error rate by linear model

- Precision and Recall

  The proposed model leads all other models with 0.9 and 0.87 precision and recall. Decision tree contributed its best to generate an accurate ensembled model by providing 0.8 precision and ROC. KSVM generates equivalent range for precision and recall.

- Fscore and ROC

  F-score of the proposed ensembled model reaches to 0.9 with 0.85 as a best ROC value among all other models that were ensembled. Least value of 0.67 and 0.63 is noticed in case of linear model. Decison tree and SVM provided almost equivalent values but decision tree could overcome SVM but failed to perform well when compared with the proposed model. Hence ensemble model again proved to best for multiple cloud applications, followed by the decision tree, and linear model.

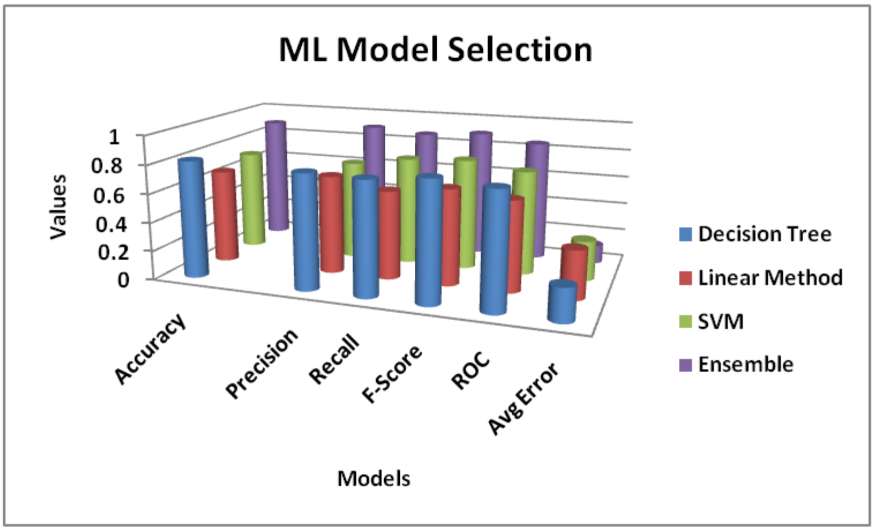
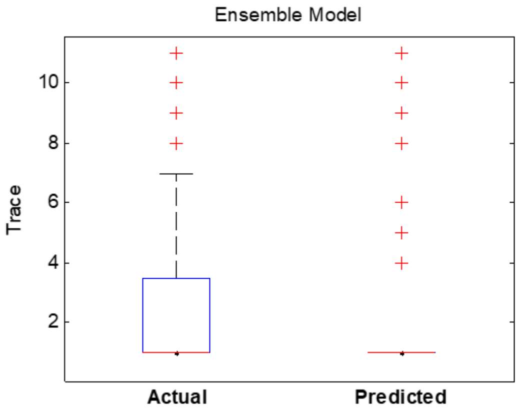


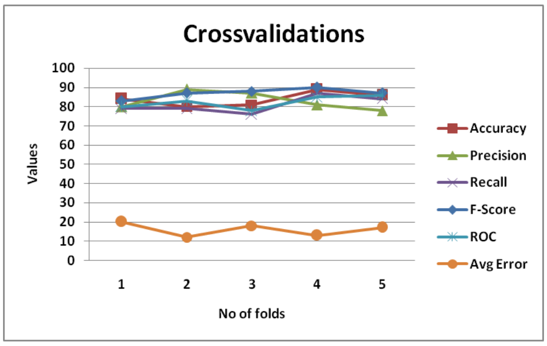
Figure 5.2: ML model selection using performance parameters:

5.2.1.2 Performance Validation: Selected Ensembled Model

The actual and predicted results of an ensembled model are confronted using boxplot. The narrow box plot for the predicted values in Figure 5.3a reveals the matching values between the actual and predicted data. Moreover, the reliability of the model is verified using K fold cross-validation for accuracy, error rate, precision, recall, ROC. Figure 5.3b depicts the 5-fold CV, where average accuracy is noticed to be 0.87 and with .15 errors. Similarly, the cross-validation of precision reflects 0.89 as the maximum value. It scored 0.82 as an average ROC.



(a) Actual-predicted data frequency



(b) Repeated K-fold cross validation

Figure 5.3: Validation for selected ensembled model

5.2.2 Results: An Intelligent Energy Aware Approach

Further, the results are recorded to investigate the relationship between application type with energy and time consumption. According to the mathematical formulation presented in the section 4.3, the total energy is consumed in data placement, disk searching and selecting which is followed by the energy consumed by the number of spins made to reach the states for some duration. The obtained results help us to understand the advantage of data predicting and tagging in energy aware approach.

5.2.2.1 Data Placement approach

The energy consumed while allocating the files to hot and cold disk is stated in Figure 5.4a using replication 2. Financial 1 consumes 7150 J to get placed in a disk. It is seen that financial 2 consumes minimum energy about 3470 J to get allocated. While allocation, the SQL proves to be the second minimum energy consumer after financial 2. It consumes 5710 J to get placed such that the requests can be executed in a more energy-efficient way. Maximum energy of 9900 J is consumed by websearch 2 whereas, websearch1 consumed 7800 J. This is a one-time allocation process that would be followed by the multiple iterations of the request execution which follows intelligent scheduling.

Figure 5.4b compares the time consumed by each application to get placed in the disks.

As shown in the figure, financial2 takes a shorter time of 3000 sec to allocate. While, webserach3 consumes 5000 sec, whereas financial 1 and websearch 1 do not show much difference while getting allocated. The allocation time for websearch 2 is maximum among all other applications which may be due to the variation in the workload. Although our proposed allocation algorithm consumed significant energy and time but it takes advantage of data locality while executing the request. Also, the data replication would reduce the probability of suffering disk spin-up delays resulting in least energy and time consumption as shown in the following subsections.

(a) Energy consumption

(b) Execution time

Figure 5.4: Energy efficient data allocation

5.2.2.2 Energy Reduction

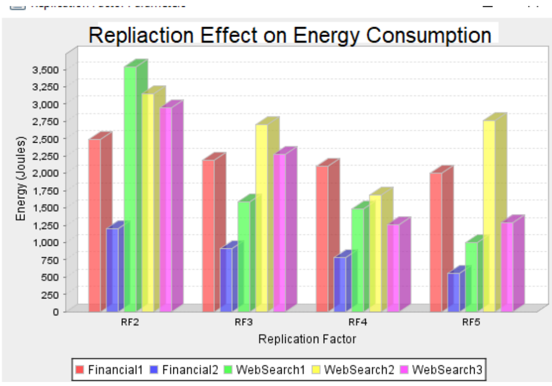
To ensure the energy efficiency and the performance of the applications in cloud data centers, the energy and time consumption have been measured when the user inputs the request to the disk. Every access to file is validated in terms of replication factor, file size and disk ratio. It also unveiled the best replica factor along with the best hot and cold disk ratio for each data application.

(a) **Energy Reduction: Replication Factor**

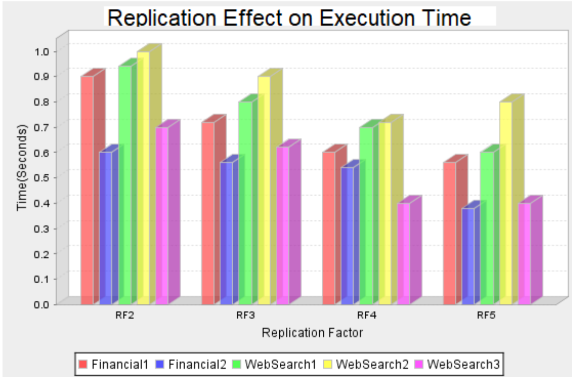
Energy consumed on executing the applications is validated on increasing the data replication factor from 1 to 5. The increases in the replication factor provides a more choice to assign a request to the most available disk such that other disks can remain in standby mode for the longer duration. Thus frequent spins can be reduced. As shown in Figure 5.5a, the energy consumption decreases with an increase in the replication factor. In case of financial 1 and financial 2, the energy consumption gently falls with an increase in the replication factor. On the other hand, energy consumed by websearch 1 drops drastically to 1000 J at replica 5. About 71.59% less energy is consumed by increasing the replication factor. Energy for websearch 2 drops from 3100 J to 1730 J on reaching replication 4. The energy consumption again increases to 2750 J as it approaches replica factor 5. A similar trend is seen for SQL trace with the least difference in energy consumption at factor 5.

The proposed algorithm also saves time with an increase in replication due to the decrease in the delays caused by disk spins. The availability of more multiple locations to execute the request is the foremost reason to reduce the execution time. The other reason can be the reduction in the additional queuing delay. Figure 5.5b compares request response time for various workloads. Financial 1 gradually reduces its time from 0.9 sec to 0.56 sec with an increases in the replication factor from 1 to 5 . Financial 2 shows slow reduction in time respectively. Websearch1 reduces the maximum execution time to 0.6 sec when replication factor is incremented to 5. This happens due to the less probability of suffering from the disk spin-up delay. Time for executing websearch2 increases from 0.72 to 0.8 sec as it approaches to replica 4 to replica 5, whereas time does not show significant effect while SQL trace is executed.

Hence, from the results, it is clear that energy consumption effects only in case of an optimized replication factor. Once that optimized factor is increased, no reduction in energy and time would be seen for any application. Also, the number of replicas should be selected wisely because excessive replication may increases the cost and traffic load within the data center.



(a) Energy consumption w.r.t replication



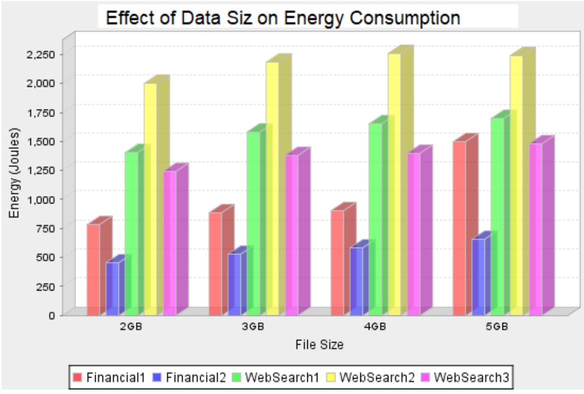
(b) Execution time w.r.t replication

Figure 5.5: Impact of data replication factor

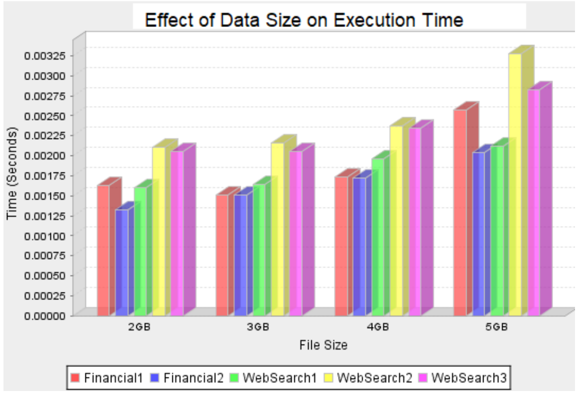
(b) **Energy Reduction: Request Size**

The impact of the average request size on energy and time consumption has been examined in Fig. 5.6. Figure 5.6a shows the energy consumed in executing the request of 2 GB, 3 GB, 4GB, 5 GB for various applications. It takes 782.228 J, 833.21 J, 900 J to run 2GB,3GB, 4 GB files of financial 1 respectively. The energy increase to 1500 J when 5 GB of file is executed. Financial 2 consumes minimum energy for executing files of all the sizes. The energy consumption ranges from 480-600 J while running 2GB to 5 GB files. Websearch 1 consumes 1400 J for 2 GB which is increased upto 1700 J to run 5 GB of file. Webserch 2 consumes maximum energy among all the traces. 2000 J and 2200 J is consumed for running 2 and 3 GB file whereas it increases upto 2250 J while 4-5 GB files are executed. SQL takes 1250 J which is increased upto 1500 J as the size increases from 1 to 5 GB.

The similar scenario is seen for the execution time. Time consumed to finish the request also increases gently with an increase in the file size (see Figure 5.6b). Response time of both financial 1 and financial 2 gently increases when the request size is prolonged from 2 to 5 GB. Similar scenario is seen for websearch 1 by consuming minimum 0.00157 sec to maximum of 0.0020 sec. WebSearch 2 again takes the maximum time to execute the file of all the size. It takes 0.00325 seconds to execute the 5GB of file.



(a) Energy consumption w.r.t data size



(b) Execution time w.r.t data size

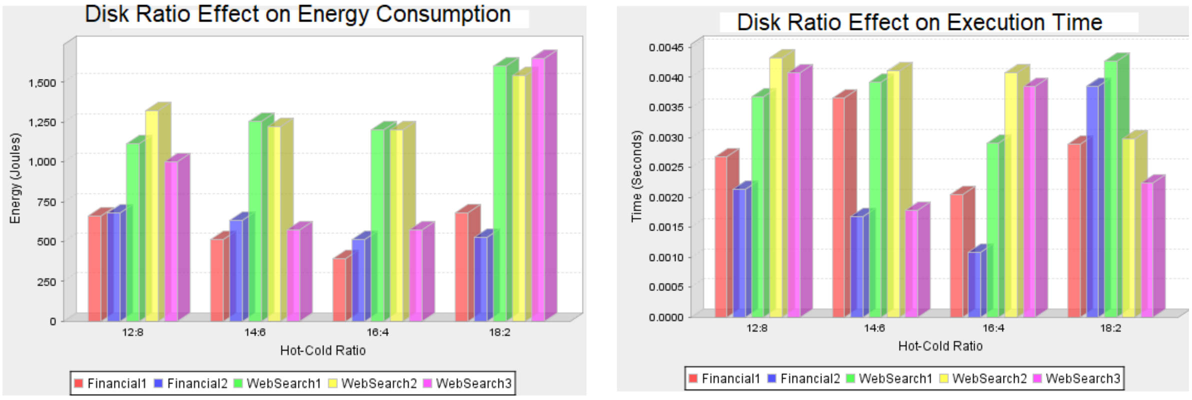
Figure 5.6: Impact of request size

(c) **Energy Reduction: Impact of Disks Ratio**

This section shows the impact of the ratio of a hot and cold disk on energy and time using disk scheduling algorithm. Various files of same size are executed using replica 3, where the range of hot and cold disks varies from 12:8 to 18:2. Figure 5.7 plots the performance patterns that are similar to those in previous sections. As shown in Figure 5.7a and 5.7b, when γ is configured from 12:8 to 18: 2, the energy and time consumption first decreases and then increases in case of financial 1. It shows the best result with 16 hot disks and 4 cold disks. The energy and time consumption decrease when the number of hot disks are brought to 16 for financial2. The best choice for financial 2 is 16: 4 which consumes about 513.177 J of energy with the lowest response time of 0.0010 sec, may be due to the repetitive use of the large amount of same data. A similar trend is seen for web search 1. Energy and time taken by 14:6 and 16:4 do not show much difference for webSearch 2. Both the ratios consumed 1222 and 1202 J energy with 0.0041, 0.0040 sec of time respectively, whereas, the same amount of energy about 573 J is consumed with 14:6 and 16:4 while processing SQL trace. It is also noticed that for all the traces, energy and time increases when the number of hot disks are brought to 18. Hence, the number of hot disk should be chosen well such that it consumes the minimum energy.

. It can be concluded that the best value of γ depends upon the type of cloud application.

Hence for all the experimental results, it can be summarized that the access delay becomes smaller with the data replication. Also, for all the replication scenarios, an increase in the size of data objects increases data access delay but with the little extent.



(a) Energy consumption w.r.t disk ratio

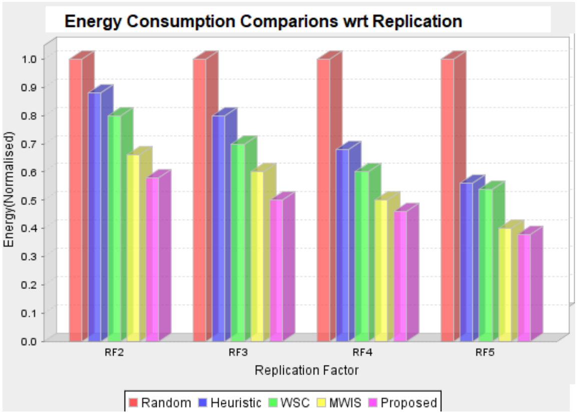
(b) Execution time w.r.t disk ratio

Figure 5.7: Impact of γ

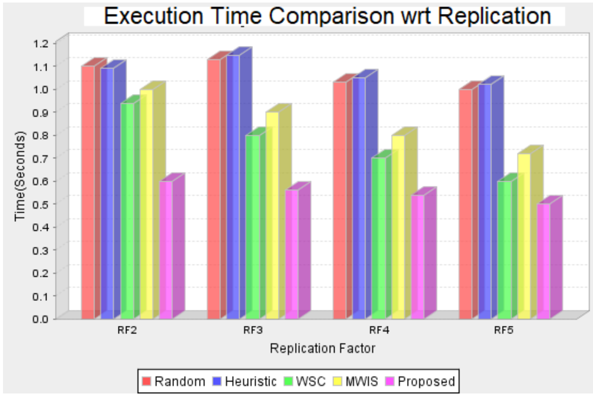
5.2.3 Comparative Analysis: Data Replication

Energy consumption among various scheduling algorithms [32] using data replication from 1 to 5 has been compared in Figure 5.8a. The energy consumed in the case of random distribution remains constant as 1 J for all the replication factors. This is because it randomly assigns the request to all the locations. Heuristic decreases the energy consumption from 0.88 to 0.8, 0.68, 0.56 J with an increase in the replication from 1 to 5. Weighted Set Cover (WSC) saves more energy then heuristic due to the priori knowledge about requests while later only has the single request information at a time. Further, Maximum Weighted Independent Set (MWIS) uses set cover and independent set algorithms to reduce energy. Since, the data replication in our experiment is selected using the frequency usage, so, the hot disks constantly receive the requests such that the cold is longer spun down. Hence, the proposed approach proves to be the lowest energy consumer among all the replica factors tested here. It successfully reduces the energy consumption from 0.58

at replica 2, 0.5 in replica 3, 0.46 at replica 4 and finally, it reduced it to 0.38 at replica 5. This leads to 6% of less energy consumption on comparing it with all the existing techniques.



(a) Energy consumption



(b) Execution time

Figure 5.8: Energy and time comparisons for data replication with existing literature

Figure 5.8b compares the response time of existing approach with the proposed when the data replication is increased to 5. Random and heuristic did not show much improvement in time. In case of random, the requests suffers additional queuing delay. The decrease in the time is seen for WSC and MWS, where WSC achieves shorter request response time than WMS. However, our proposed technique significantly reduces the time execution from 0.6 to 0.5 while approaching replica 5. Hence the response time is reduced by 18.26% due to fewer disk spin-up/down operations.

5.3 Conclusion

The performance of an intelligent energy aware approach has been validated on OLTP traces by integrating storage capabilities in the cloud environment. The storage prediction model used in a approach possess the least error rate, highest accuracy, precision, recall, and F-score on predicting file frequency. Further, these prediction results have been used to place the data among the disks. The requests are scheduled such that least energy and time is consumed while executing it. The proposed framework could save data pollution by replicating the data that would be accessed frequently. It could recognize the optimal replica factor for each application. It also shows an increase in the energy consumption and delay with an increase in the data size. It could reveal the best ratio of a hot and cold disk based on the application. The next chapter concludes all the chapters and presents future directions.

Chapter 6

Conclusion and Future Scope

In the world of overflowing data, cloud storage has become a prevalent technology for saving information-driven data. Cloud data centers provide on-demand computing storage to users by storing and processing IO intensive applications. However, the cloud deployment of these applications lead to the various challenges including energy efficiency. Energy consumption in data centers is still an important concern that requires attention. Storage power consumption comprising of disks is maximum among all the other entities present in data centers.

This thesis addresses the need for a prediction based energy efficient approach for the cloud environment. The work proposes storage prediction model that implements classified ensemble approach for predicting the frequency of data to analyze data usage patterns. Further, based on the prediction results of storage prediction model, it implements an intelligent energy-aware approach which places the predicted popular requests in hot disk and schedules the request to the most available disk. The proposed framework is also validated on SQL and OLTP traces

Finally, the whole thesis has been summarized in this chapter. The progress made in attaining the objectives of the research work in terms of prediction and energy efficiency in the cloud environment has been well portrayed. The outcome of each chapter, along with the contributions of the research work, are also specified briefly. Furthermore, this chapter outlines a number of directions for future research work.

6.1 Conclusion

The motive of this research work has been to provide an intelligent energy aware approach for big data storage in cloud data centers. The whole process is validated using SQL trace of real-world Dublin use cases in real world invironmet. Further The performance has also been evaluated on OLTP (e.g., e-commerce) applications benchmarked with financial and websearch I/O traces in cloud environment. The key findings of this research work are summarized below:

The Chapter 1 begins with an overview on big data and its applications. It clearly demonstrates the relationship between big data and cloud computing by exploring cloud data centers. Further, it discusses various big data challenges where energy efficiency has been recognised as a major challenge. It reflects the taxonomy of energy efficiency techniques where challenge efficient storage presents the motivation behind this research work. Thus primary contributions has been framed and organization of the rest of this thesis has been laid.

Chapter 2 reveals the empirical research on big data and its challenges, where the storage energy efficiency has been recognised as a major challenge which is in infancy. So, the various ways to resolve this challenge has been well explored. It surveys existing data prediction approaches followed by existing energy efficient data placement techniques. Additionally, an extensive survey of the existing energy efficient disk scheduling that includes various threshold techniques are explored. This chapter also reveals various tools and platforms used by various authors to implement storage energy-efficient techniques. Further, the limitations and gaps pointed out in the existing research help us to bridge the research gaps. Based on these gaps, the problem formulation and objectives of this thesis are outlined.

Chapter 3, reflects the need to get familiar with the data usage trend in order to reduce storage energy consumption. The prediction can be used in efficient data placement among disks that would lead to the efficient disk scheduling, hence save energy and time. Prediction results also help in identifying which data objects need to be replicated and where. This chapter proposes storage prediction model that firstly, generates the SQL trace of SCATS data of Dublin city, Ireland. Some addition and essential features such as frequency and idle time are extracted using tools and arithmetic caculations. Machine learning-based ensembled approach is applied to predict the frequently used query using frequency as a target variable. Furthermore, the predicted results are tagged as popular and unpopular data based on threshold frequency. The performance of the proposed approach is validated with the 87.5% accuracy, 11% error rates, 89% precision, 87% recall, and 88% F-score.The Experimental results proved to be improved on comparing it with the existing models.

Chapter 4 reveals the next step to prediction that needs to be taken in order to obtain energy efficient data centers. Prediction results are used in efficient data allocations followed by the intelligent scheduling. This chapter showcases an intelligent energy aware approach where the predicted popular data is replicated in hot and as well as cold disks while unpopular data is only placed in cold disks. When user inputs the request, an intelligent scheduling technique is applied that searches and selects the most available disk that will execute the request in minimum energy and time consumption. The standby disk will not be disturbed in any case thus maximising the energy savings. The whole framework is validated in a real-world environment, where our proposed technique could save up to75% energy with 9.7% decreased execution time when compared with existing techniques.

Chapter 5 explores, validates, tests and compares the whole intelligent energy aware approach on OLTP applications in cloud environment. Various traces of OLTP applications such as financial, financial2, websearch1, websearch2, and webserach3 contains common features. Among all the features, the ASU is selected whose frequency has been calculated, predicted and tagged as the popular and unpopular files using predefined threshold. The proposed technique is implemented in the cloud environment using CloudSimDisk that models the storage component. The effect of 39% energy and 18.26% time savings are noticed on varying the replication factor, file size, and disk ratio for each trace. The optimized replica, best disk ratio for each application is noticed that can be used to make energy-efficient data centers. The following outlines the key contributions of the thesis:

- Various characteristics and challenges of big data applications in cloud data centers have been explored. Subsequently, the relationship between big data and cloud computing has been studied. The existing energy efficiency challenges pertaining to big data storage in cloud data centers have been recognized.

- Pertaining to the energy-efficient storage techniques, various research challenges have been identified followed by its solution that includes prediction approaches, data placement and disk scheduling techniques

- Storage prediction model has been proposed that predicts data frequency using classified ensembled approach. Based on the prediction results, it categorizes the popular and unpopular data using a predefined threshold frequency.

- An intelligent energy-aware approach has been framed that takes predicted popular and unpopular requests as an input to distribute them in hot and cold disks respectively. It further schedules the request to the best-identified disk that can save maximum disk spins and consume the least energy and time. Execution time, energy savings are validated on the real-world use case of Dublin City, in a real-time environment.

- The results of the proposed intelligent energy-aware approach are also validated in cloud environment using various big data commercial applications such as OLTP. Besides the best replica and disk ratio for each application has been recognised based on the performed experiments.

- Besides the experimental results are compared with the existing approaches that exhibit the efficacy of the proposed approach with 87.5% accuracy, 87% recall, 89% precision, 11% error rate and 88% F-score for prediction results. Also, the energy savings for 39% along with the 18.26% time reduction are received for energy aware approach along with the little overhead cost.

6.2 Future Scope

This thesis elevates the knowledge of intelligent energy aware approach for cloud data centers. It has uncovered new research areas in cloud computing that requires further research. Some of the future research directions are mentioned below:

- Various cache levels and disk blocks can be brought in to the picture. Data migrations and write offloading can substantially be used in a future to enhance the productivity of the framework.

- The future plan also includes the involvement of hadoop technology in the proposed system. Hadoop Distributed File System (HDFS) data node replication policy would be applied on multiple disks which can result in faster data processing with more energy savings.

References

[1] Rachad Atat, Lingjia Liu, Jinsong Wu, Guangyu Li, Chunxuan Ye, and Yi Yang. Big data meet cyber-physical systems: A panoramic survey. *IEEE Access*, 6:73603–73636, 2018.

[2] M Shamim Hossain, Mohamad Hoda, Ghulam Muhammad, Ahmad Almogren, and Atif Alamri. Cloud-supported framework for patients in post-stroke disability rehabilitation. *Telematics and Informatics*, 35(4):826–836, 2018.

[3] Isabel de la Torre Díez, Héctor Merino Cosgaya, Begoña Garcia-Zapirain, and Miguel López-Coronado. Big data in health: a literature review from the year 2005. *Journal of medical systems*, 40(9):209, 2016.

[4] Muhammad Fahim and Thar Baker. Knowledge-based decision support systems for personalized u-lifecare big data services. In *Current Trends on Knowledge-Based Systems*, pages 187–203. Springer, 2017.

[5] Sang-Hoon Kim, Dawoon Jung, Jin-Soo Kim, and Seungryoul Maeng. Heterodrive: Re-shaping the storage access pattern of oltp workload using ssd. In *Proceedings of 4th International Workshop on Software Support for Portable Storage (IWSSPS 2009)*, pages 13–17, 2009.

[6] Jinsong Wu, Song Guo, Jie Li, and Deze Zeng. Big data meet green challenges: Big data toward green applications. *IEEE Systems Journal*, 10(3):888–900, 2016.

[7] Nouf Alkhater, Robert Walters, and Gary Wills. An empirical study of factors influencing cloud adoption among private sector organisations. *Telematics and Informatics*, 35(1):38–54, 2018.

[8] Md Gofran Faroqi, Noore Alam Siddiquee, and Shahid Ullah. Sustainability of telecentres in developing countries: lessons from union digital centre in bangladesh. *Telematics and Informatics*, 37:113–127, 2019.

[9] Kenndy Njenga, Lalit Garg, Amit Kumar Bhardwaj, Vijay Prakash, and Seema Bawa. The cloud computing adoption in higher learning institutions in kenya: hindering factors and recommendations for the way forward. *Telematics and Informatics*, 38:225–246, 2018.

[10] Wikipedia. Exabyte. `https://en.wikipedia.org/wiki/Exabyte`, 2019. [Online; accessed 19-Jan-2019].

[11] Tony Redmond. Unpacking office 365: Exploring microsoft's data centers and regions, 2019. https://www.avepoint.com/blog/office-365/office-365-data-centers/, [Online; accessed 1-Dec-2019].

[12] N. Sönnichsen. Energy consumption of Google from 2011 to 2018 . `https://www.statista.com/statistics/788540/energy-consumption-of-google/`, 2019. [Online; accessed 1-Dec-2019].

[13] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of "big data" on cloud computing: Review and open research issues. *Information systems*, 47:98–115, 2015.

[14] Brijesh B Mehta and Udai Pratap Rao. Privacy preserving unstructured big data analytics: Issues and challenges. *Procedia Computer Science*, 78:120–124, 2016.

[15] Rong Ge, Xizhou Feng, and Kirk W Cameron. Performance-constrained distributed dvs scheduling for scientific applications on power-aware clusters. In *SC'05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, pages 34–34. IEEE, 2005.

[16] Jong Hyuk Park. Advanced it-based future sustainable computing (2017–2018), 2019.

[17] Maria Deutscher. siliconAngle. `https://siliconangle.com/2020/02/26/google-says-will-spend-10b-data-centers-offices-2020/`, 2019. [Online; accessed 26-Feb-2020].

[18] Suzanne Rivoire, Mehul A Shah, Parthasarathy Ranganathan, and Christos

Kozyrakis. Joulesort: a balanced energy-efficiency benchmark. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 365–376. ACM, 2007.

[19] Dolores Añón Higón, Roya Gholami, and Farid Shirazi. Ict and environmental sustainability: A global perspective. *Telematics and Informatics*, 34(4):85–95, 2017.

[20] Mark Golden. Data centers can slash co2 emissions 88% or more. `https://energy.stanford.edu/news/data-centers-can-slash-co2-emissions-88-or-more`, 2013. [Online; accessed 18-Jan-2013].

[21] Mark Golden. Efficiency. `https://www.google.com/about/datacenters/efficiency/`. [Online; accessed 18-Jan-2020].

[22] Huigui Rong, Haomin Zhang, Sheng Xiao, Canbing Li, and Chunhua Hu. Optimizing energy consumption for data centers. *Renewable and Sustainable Energy Reviews*, 58:674–691, 2016.

[23] Nidhi Jain Kansal and Inderveer Chana. An empirical evaluation of energy-aware load balancing technique for cloud data center. *Cluster Computing*, 21(2):1311–1329, 2018.

[24] Thar Baker, Jose M García-Campos, Daniel Gutiérrez Reina, Sergio Toral, Hissam Tawfik, Dhiya Al-Jumeily, and Abir Hussain. Greeaodv: an energy efficient routing protocol for vehicular ad hoc networks. In *International Conference on Intelligent Computing*, pages 670–681. Springer, 2018.

[25] Xiaomin Zhu, Laurence T Yang, Huangke Chen, Ji Wang, Shu Yin, and Xiaocheng Liu. Real-time tasks oriented energy-aware scheduling in virtualized clouds. *IEEE Transactions on Cloud Computing*, 2(2):168–180, 2014.

[26] Yizhou Yan and Wenjun Wu. Analysis of energy consumption of deduplication in storage systems. In *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 295–301. IEEE, 2015.

[27] Seokhei Cho, Changhyun Park, Youjip Won, Sooyong Kang, Jaehyuk Cha, Sungroh Yoon, and Jongmoo Choi. Design tradeoffs of ssds: From energy consumptions

perspective. *ACM Transactions on Storage (TOS)*, 11(2):8, 2015.

[28] Sumedha Arora and Anju Bala. A survey: Ict enabled energy efficiency techniques for big data applications. *Cluster Computing*, pages 1–22, 2019.

[29] Anthony Hylick, Ripduman Sohan, Andrew Rice, and Brian Jones. An analysis of hard drive energy consumption. In *Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008. IEEE International Symposium on*, pages 1–10. IEEE, 2008.

[30] A Galicia, R Talavera-Llames, A Troncoso, I Koprinska, and F Martínez-Álvarez. Multi-step forecasting for big data time series based on ensemble learning. *Knowledge-Based Systems*, 163:830–841, 2019.

[31] Dejene Boru, Dzmitry Kliazovich, Fabrizio Granelli, Pascal Bouvry, and Albert Y Zomaya. Energy-efficient data replication in cloud computing datacenters. *Cluster computing*, 18(1):385–402, 2015.

[32] Jerry Chou, Jinoh Kim, and Doron Rotem. Energy-aware scheduling in disk storage systems. In *2011 31st International Conference on Distributed Computing Systems*, pages 423–433. IEEE, 2011.

[33] Manu Vardhan, Dharmendra K Yadav, and Dharmender Singh Kushwaha. A transparent service replication mechanism for clouds. In *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems*, pages 389–394. IEEE, 2012.

[34] Tao Xie. Sea: A striping-based energy-aware strategy for data placement in raid-structured storage systems. *IEEE Transactions on Computers*, 57(6):748–761, 2008.

[35] Jinsong Wu, Song Guo, Jie Li, and Deze Zeng. Big data meet green challenges: Greening big data. *IEEE Systems Journal*, 10(3):873–887, 2016.

[36] Thar Baker, Bandar Al-Dawsari, Hissam Tawfik, D Reid, and Yanik Ngoko. Greedi: An energy efficient routing algorithm for big data on cloud. *Ad Hoc Networks*, 35:83–96, 2015.

[37] Thar Baker, Yanik Ngoko, Rafael Tolosana-Calasanz, Omer F Rana, and Mar-

tin Randles. Energy efficient cloud computing environment via autonomic meta-director framework. In *2013 Sixth International Conference on Developments in eSystems Engineering*, pages 198–203. IEEE, 2013.

[38] Ibrahim Moghram and Saifur Rahman. Analysis and evaluation of five short-term load forecasting techniques. *IEEE Transactions on power systems*, 4(4):1484–1491, 1989.

[39] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, 26(2):1–26, 2008.

[40] Katarina Grolinger, Alexandra L'Heureux, Miriam AM Capretz, and Luke Seewald. Energy forecasting for event venues: Big data and prediction accuracy. *Energy and Buildings*, 112:222–233, 2016.

[41] Mohit Dayal and Nanhay Singh. Indian health care analysis using big data programming tool. *Procedia Computer Science*, 89:521–527, 2016.

[42] M Haridas. Redefining military intelligence using big data analytics. pages 72–78, 2015.

[43] Gema Bello-Orgaz, Jason J Jung, and David Camacho. Social big data: Recent achievements and new challenges. *Information Fusion*, 28:45–59, 2016.

[44] Audrey W Zhu and Halton Pi. A method for improving the accuracy of weather forecasts based on a comprehensive statistical analysis of historical data for the contiguous united states. *Journal of Climatology & Weather Forecasting*, 2(1):1–4, 2014.

[45] Dany Vohl, Christopher J Fluke, and Georgios Vernardos. Data compression in the petascale astronomy era: A gerlumph case study. *Astronomy and Computing*, 12:200–211, 2015.

[46] James M Tien. Big data: Unleashing information. *Journal of Systems Science and Systems Engineering*, 22(2):127–151, 2013.

[47] Lin Dai, Xin Gao, Yan Guo, Jingfa Xiao, and Zhang Zhang. Bioinformatics clouds for big data manipulation. *Biology direct*, 7(1):43, 2012.

[48] Islam Atta, Pinar Tözün, Xin Tong, Anastasia Ailamaki, and Andreas Moshovos. Strex: boosting instruction cache reuse in oltp workloads through stratified transaction execution. In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, pages 273–284, 2013.

[49] Ricardo Koller, Akshat Verma, and Anindya Neogi. Wattapp: an application aware power meter for shared data centers. In *Proceedings of the 7th international conference on Autonomic computing*, pages 31–40. ACM, 2010.

[50] Chao Li, Wangyuan Zhang, Chang-Burm Cho, and Tao Li. Solarcore: Solar energy driven multi-core architecture power management. In *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, pages 205–216. IEEE, 2011.

[51] Ali Hammadi and Lotfi Mhamdi. A survey on architectures and energy efficiency in data center networks. *Computer Communications*, 40:1–21, 2014.

[52] Qingbo Zhu, Francis M David, Christo Frank Devaraj, Zhenmin Li, Yuanyuan Zhou, and Pei Cao. Reducing energy consumption of disk storage using power-aware cache management. In *Software, IEEE Proceedings-*, pages 118–129. IEEE, 2004.

[53] Saiqin Long, Yuelong Zhao, and Wei Chen. A three-phase energy-saving strategy for cloud storage systems. *Journal of Systems and Software*, 87:38–47, 2014.

[54] Akshat Verma, Ricardo Koller, Luis Useche, and Raju Rangaswami. Energy proportional storage using dynamic consolidation. In *In Proceedings of the File and Storage Systems*, pages 23–26. Citeseer, 2010.

[55] Farhad Mehdipour, Hamid Noori, and Bahman Javadi. Chapter two-energy-efficient big data analytics in datacenters. *Advances in Computers*, 100:59–101, 2016.

[56] Shadi Ibrahim, Tien-Dat Phan, Alexandra Carpen-Amarie, Houssem-Eddine Chihoub, Diana Moise, and Gabriel Antoniu. Governing energy consumption in hadoop

through cpu frequency scaling: An analysis. *Future Generation Computer Systems*, 54:219–232, 2016.

[57] Donato Barbagallo, Elisabetta Di Nitto, Daniel J Dubois, and Raffaela Mirandola. A bio-inspired algorithm for energy optimization in a self-organizing data center. In *Self-Organizing Architectures*, pages 127–151. Springer, 2010.

[58] Tadashi Nakano. Biologically inspired network systems: A review and future prospects. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(5):630–643, 2011.

[59] Ashok Gautham, Kunal Korgaonkar, Patanjali Slpsk, Shankar Balachandran, and Kamakoti Veezhinathan. The implications of shared data synchronization techniques on multi-core energy efficiency. In *HotPower*, pages 1–5.

[60] Arthur Francisco Lorenzon, Márcia Cristina Cera, and Antonio Carlos Schneider Beck. Investigating different general-purpose and embedded multicores to achieve optimal trade-offs between performance and energy. *Journal of Parallel and Distributed Computing*, 95:107–123, 2016.

[61] Alex X Zhang, Fereydoon Safai, Dirk M Beyer, Jerome Rolia, and Marie-Jo L Fremont. Performance-data based server consolidation, August 28 2012. US Patent 8,255,516.

[62] Jian-Sheng Liao, Chi-Chung Chang, Yao-Lun Hsu, Xiao-Wei Zhang, Kuan-Chou Lai, and Ching-Hsien Hsu. Energy-efficient resource provisioning with sla consideration on cloud computing. In *Parallel Processing Workshops (ICPPW), 2012 41st International Conference on*, pages 206–211. IEEE, 2012.

[63] Eugen Feller, Lavanya Ramakrishnan, and Christine Morin. Performance and energy efficiency of big data applications in cloud environments: A hadoop case study. *Journal of Parallel and Distributed Computing*, 79:80–89, 2015.

[64] Fábio D Rossi, Miguel G Xavier, César AF De Rose, Rodrigo N Calheiros, and Rajkumar Buyya. E-eco: Performance-aware energy-efficient cloud data center orchestration. *Journal of Network and Computer Applications*, 78:83–96, 2017.

[65] Christoforos Kachris, Georgios Ch Sirakoulis, and Dimitrios Soudris. A mapreduce scratchpad memory for multi-core cloud computing applications. *Microprocessors and Microsystems*, 39(8):599–608, 2015.

[66] Pedro HP Castro, Vívian L Barreto, Sand Luz Corrêa, Lisandro Zambenedetti Granville, and Kleber Vieira Cardoso. A joint cpu-ram energy efficient and sla-compliant approach for cloud data centers. *Computer Networks*, 94:1–13, 2016.

[67] Anton Beloglazov, Rajkumar Buyya, Young Choon Lee, Albert Zomaya, et al. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in computers*, 82(2):47–111, 2011.

[68] Thar Baker, Muhammad Asim, Hissam Tawfik, Bandar Aldawsari, and Rajkumar Buyya. An energy-aware service composition algorithm for multiple cloud-based iot applications. *Journal of Network and Computer Applications*, 89:96–108, 2017.

[69] Min Sang Yoon, Ahmed E Kamal, and Zhengyuan Zhu. Adaptive data center activation with user request prediction. *Computer Networks*, 122:191–204, 2017.

[70] Wei Wu, Wenjia Li, Deify Law, and Woonki Na. Improving data center energy efficiency using a cyber-physical systems approach: integration of building information modeling and wireless sensor networks. *Procedia Engineering*, 118:1266–1273, 2015.

[71] Amit Munjal, Rajiv Kr Tripathi, and Yatindra Nath Singh. Balancing energy consumption using cluster based approach in wireless sensor network. In *2014 Twentieth National Conference on Communications (NCC)*, pages 1–5. IEEE, 2014.

[72] HS Sun and SE Lee. Case study of data centers' energy performance. *Energy and Buildings*, 38(5):522–533, 2006.

[73] Jorge Guerra, Wendy Belluomini, Joseph Glider, Karan Gupta, and Himabindu Pucha. Energy proportionality for storage: Impact and feasibility. *ACM SIGOPS Operating Systems Review*, 44(1):35–39, 2010.

[74] Zhu Xiao, Shuangchun Li, Xiaochun Chen, Dong Wang, and Wenjie Chen. A load-balancing energy consumption minimization scheme in 5g heterogeneous small cell wireless networks under coverage probability analysis. *International Journal of*

Pattern Recognition and Artificial Intelligence, 31(07):1759013, 2017.

[75] Murray G Patterson. What is energy efficiency?: Concepts, indicators and methodological issues. *Energy policy*, 24(5):377–390, 1996.

[76] Richard Hankins, Trung Diep, Murali Annavaram, Brian Hirano, Harald Eri, Hubert Nueckel, and John Paul Shen. Scaling and characterizing database workloads: Bridging the gap between research and practice. In *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36.*, pages 151–162. IEEE, 2003.

[77] Axel Busch, Qais Noorshams, Samuel Kounev, Anne Koziolek, Ralf Reussner, and Erich Amrehn. Automated workload characterization for i/o performance analysis in virtualized environments. In *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, pages 265–276. ACM, 2015.

[78] Jaehyung Kim, Jinuk Park, and Sanghyun Park. Neural network for saturation prediction of solid state drives.

[79] Salvatore Capra. Cloud computing trace characterization and synthetic workload generation. Technical report, Air force inst of tech wright-patterson afb oh graduate school of engineering and management, 2013.

[80] Hui Kang and Jennifer L Wong. vcsimx86: a cache simulation framework for x86 virtualization hosts. *Stony Brook University*.

[81] Tomislav Janjusic and Krishna Kavi. Gleipnir: A memory profiling and tracing tool. *ACM SIGARCH Computer Architecture News*, 41(4):8–12, 2013.

[82] Waheed Iqbal, Abdelkarim Erradi, and Arif Mahmood. Dynamic workload patterns prediction for proactive auto-scaling of web applications. *Journal of Network and Computer Applications*, 124:94–107, 2018.

[83] Matthieu Dorier, Shadi Ibrahim, Gabriel Antoniu, and Rob Ross. *On the Use of Formal Grammars to Predict HPC I/O Behaviors*. PhD thesis, ENS Rennes; Inria Rennes Bretagne Atlantique; Argonne National Laboratory; INRIA, 2015.

[84] Qais Noorshams, Kiana Rostami, Samuel Kounev, Petr Tuma, and Ralf Reussner.

I/o performance modeling of virtualized storage systems. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2013 IEEE 21st International Symposium on*, pages 121–130. IEEE, 2013.

[85] Sohail Sarwar, Zia Ul-Qayyum, and Owais Ahmed Malik. A hybrid intelligent system to improve predictive accuracy for cache prefetching. *Expert Systems with Applications*, 39(2):1626–1636, 2012.

[86] James Oly and Daniel A Reed. Markov model prediction of i/o requests for scientific applications. In *Proceedings of the 16th international conference on Supercomputing*, pages 147–155. ACM, 2002.

[87] Miyuru Dayarathna, Yonggang Wen, and Rui Fan. Data center energy consumption modeling: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):732–794, 2015.

[88] Dennis Colarelli and Dirk Grunwald. Massive arrays of idle disks for storage archives. In *Supercomputing, ACM/IEEE 2002 Conference*, pages 47–47. IEEE, 2002.

[89] Trishul M Chilimbi. Efficient representations and abstractions for quantifying and exploiting data reference locality. In *ACM SIGPLAN Notices*, volume 36, pages 191–202. ACM, 2001.

[90] Avani Wildani and Ethan L Miller. Can we group storage? statistical techniques to identify predictive groupings in storage system accesses. *ACM Transactions on Storage (TOS)*, 12(2):7, 2016.

[91] Xiaodong Meng, Chentao Wu, Minyi Guo, Long Zheng, and Jingyu Zhang. Pam: an efficient power-aware multilevel cache policy to reduce energy consumption of storage systems. *Frontiers of Computer Science*, 13(4):850–863, 2019.

[92] Xiaoyu Yao and Jun Wang. Rimac: a novel redundancy-based hierarchical cache architecture for energy efficient, high performance storage systems. In *ACM SIGOPS Operating Systems Review*, volume 40, pages 249–262. ACM, 2006.

[93] Zichao Yuan, Ce Yu, Jizhou Sun, Jian Xiao, Jianmei Wang, Zhaohui Shang, and

Yi Hu. An energy efficient storage system for astronomical observation data on dome a. In *International Conference on Algorithms and Architectures for Parallel Processing*, pages 33–46. Springer, 2015.

[94] Tao Xie and Yao Sun. Dynamic data reallocation in hybrid disk arrays. *IEEE Transactions on Parallel and Distributed Systems*, 21(9):1330–1341, 2010.

[95] Xiaodong Li, Zhenmin Li, Yuanyuan Zhou, and Sarita Adve. Performance directed energy management for main memory and disks. *ACM Transactions on Storage (TOS)*, 1(3):346–380, 2005.

[96] Richard Golding, Peter Bosch, John Wilkes, et al. Idleness is not sloth. In *USENIX*, pages 201–212, 1995.

[97] Dong Li and Jun Wang. Eeraid: energy efficient redundant and inexpensive disk array. In *Proceedings of the 11th workshop on ACM SIGOPS European workshop*, page 29. ACM, 2004.

[98] Lingwei Zhang, Yuhui Deng, Weiheng Zhu, Jipeng Zhou, and Frank Wang. Skewly replicating hot data to construct a power-efficient storage cluster. *Journal of Network and Computer Applications*, 50:168–179, 2015.

[99] Tobias Sturm, Foued Jrad, and Achim Streit. Storage cloudsim. In *Proceedings of the 4th International Conference on Cloud Computing and Services Science*, pages 186–192. SCITEPRESS-Science and Technology Publications, Lda, 2014.

[100] Saiqin Long and Yuelong Zhao. A toolkit for modeling and simulating cloud data storage: An extension to cloudsim. In *Control Engineering and Communication Technology (ICCECT), 2012 International Conference on*, pages 597–600. IEEE, 2012.

[101] Sergio Vázquez Pérez, José Rodríguez, Marco Rivera, Leopoldo García Franquelo, and Margarita Norambuena. Model predictive control for power converters and drives: Advances and trends. *IEEE Transactions on Industrial Electronics, 64 (2), 935-947.*, 2017.

[102] Mousumi Paul and Srivatsa Krishnaswamy. System and method to enable

prediction-based power management, September 19 2019. US Patent App. 15/978,176.

[103] Peyman Behzadnia, Yi-Cheng Tu, Bo Zeng, and Wei Yuan. Energy-aware disk storage management: Online approach with application in dbms. *arXiv preprint arXiv:1703.02591*, 2017.

[104] David P Helmbold, Darrell DE Long, and Bruce Sherrod. A dynamic disk spin-down technique for mobile computing. In *Proceedings of the 2nd annual international conference on Mobile computing and networking*, pages 130–142. ACM, 1996.

[105] Yongqiang Gao, Haibing Guan, Zhengwei Qi, Bin Wang, and Liang Liu. Quality of service aware power management for virtualized data centers. *Journal of Systems Architecture*, 59(4-5):245–259, 2013.

[106] Shu Yin, Xuewu Li, Kenli Li, Jianzhong Huang, Xiaojun Ruan, Xiaomin Zhu, Wei Cao, and Xiao Qin. Reed: A reliable energy-efficient raid. In *2015 44th International Conference on Parallel Processing*, pages 649–658. IEEE, 2015.

[107] Divya Mahajan, Cody Blakeney, and Ziliang Zong. Improving the energy efficiency of relational and nosql databases via query optimizations. *Sustainable Computing: Informatics and Systems*, 22:120–133, 2019.

[108] Baptiste Louis, Karan Mitra, Saguna Saguna, and Christer Åhlund. Cloudsimdisk: Energy-aware storage simulation in cloudsim. In *Utility and Cloud Computing (UCC), 2015 IEEE/ACM 8th International Conference on*, pages 11–15. IEEE, 2015.

[109] Adrian M Caulfield, Laura M Grupp, and Steven Swanson. Gordon: using flash memory to build fast, power-efficient clusters for data-intensive applications. *ACM Sigplan Notices*, 44(3):217–228, 2009.

[110] Nikolay Grozev and Rajkumar Buyya. Performance modelling and simulation of three-tier applications in cloud and multi-cloud environments. *The Computer Journal*, 58(1):1–22, 2013.

[111] Junichi Tatemura, Oliver Po, and Hakan Hacgümüş. Microsharding: a declarative

approach to support elastic oltp workloads. *ACM SIGOPS Operating Systems Review*, 46(1):4–11, 2012.

[112] Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Mezerhane, Andrew Pavlo, and Geoffrey J Gordon. Query-based workload forecasting for self-driving database management systems. In *Proceedings of the 2018 International Conference on Management of Data*, pages 631–645. ACM, 2018.

[113] Zhen Sun, Geoff Kuenning, Sonam Mandal, Philip Shilane, Vasily Tarasov, Nong Xiao, Erez Zadok, et al. Cluster and single-node analysis of long-term deduplication patterns. *ACM Transactions on Storage (TOS)*, 14(2):13, 2018.

[114] Lizhe Wang, Gregor von Laszewski, Fang Huang, Jai Dayal, Tom Frulani, and Geoffrey Fox. Task scheduling with ann-based temperature prediction in a data center: a simulation-based study. *Engineering with Computers*, 27(4):381–391, 2011.

[115] Sukhwinder Singh, Vinod Kumar, HK Verma, and Dilbag Singh. Svm based system for classification of microcalcifications in digital mammograms. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4747–4750. IEEE, 2006.

[116] Manoj Khandelwal. Blast-induced ground vibration prediction using support vector machine. *Engineering with Computers*, 27(3):193–200, 2011.

[117] Oscar Cordón, Przemysław Kazienko, and Bogdan Trawiński. Special issue on hybrid and ensemble methods in machine learning. *New Generation Computing*, 29(3):241, 2011.

[118] Thomas Liebig, Nico Piatkowski, Christian Bockermann, and Katharina Morik. Dynamic route planning with real-time traffic predictions. *Information Systems*, 64:258–265, 2017.

[119] Junhai Zhai, Liguang Zang, and Zhaoyi Zhou. Ensemble dropout extreme learning machine via fuzzy integral for data classification. *Neurocomputing*, 275:1043–1052, 2018.

[120] Wen-Tung Chang and Long-Iong Wu. Computerized tolerance analysis of disk cam

mechanisms with a roller follower. *Engineering with Computers*, 25(3):247–260, 2009.

[121] Severin Sadjina and Eilif Pedersen. Energy conservation and coupling error reduction in non-iterative co-simulations. *Engineering with Computers*, pages 1–9, 2016.

[122] Srikanth Vasamsetti, Neerja Mittal, Bala Chakravarthy Neelapu, and Harish Kumar Sardana. 3d local spatio-temporal ternary patterns for moving object detection in complex scenes. *Cognitive Computation*, 11(1):18–30, 2019.

[123] Yuhui Deng, Frank Wang, and Na Helian. Eed: energy efficient disk drive architecture. *Information Sciences*, 178(22):4403–4417, 2008.

[124] Sang-Won Kang, MoonBae Song, KwangJin Park, and Chong-Sun Hwang. Towards an efficient semantic prefetching for location based services. In *iiWAS*, 2004.

[125] Haoyuan Wang and Zhiwei Luo. Data cache prefetching with perceptron learning. *arXiv preprint arXiv:1712.00905*, 2017.

[126] Shu Yin, Zhaoyu Xiao, Kenli Li, Jianzhong Huang, Xiaojun Ruan, Xiaomin Zhu, and Xiao Qin. Ress: A reliable energy-efficient storage system. In *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pages 1193–1198. IEEE, 2016.