Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2022 SCPE. Volume 23, Issues 1, pp. 35–50, DOI 10.12694/scpe.v23i1.1971

A COMPREHENSIVE SURVEY ON ENERGY CONSUMPTION ANALYSIS FOR NOSQL

MONIKA SHAH[*], AMIT KOTHARI[†] AND SAMIR PATEL[‡]

Abstract. During the last few years, we are witnessing increasing development in the Internet of Things (IoT) and big data. To address increasing workload complexity with better performance and to handle scalability issues of such applications, non-relational (NoSQL) has started taking the place of relational databases. With increasing load, it is challenging to maintain NoSQL's performance, scalability, and availability without expanding the capacity of hosts and power budget of computing resources [57]. Future scaling of data center capabilities depends on the improvement of server power efficiency [22, 33]. Considering the rise of energy costs and environmental sustainability, we can not ignore this high energy consumption caused by NoSQL. Despite the increasing popularity and share of NoSQL in the software market, little is still known about its energy footprint. To the best of our knowledge, there are no comprehensive studies that analyze the energy consumption by various modules of NoSQL. This article, therefore, conducts a comprehensive survey on the energy consumption analysis of NoSQL. There are limited proposals to reduce the energy consumption of NoSQL. This paper also provides a brief description of these little efforts on reducing the energy consumption of NoSQL. Based on the review, this paper discusses the research scope and opportunities for researchers to improve the energy conservation of NoSQL systems.

Key words: NoSQL, Energy Consumption, Efficiency Analysis, Energy Conservation, Power Management, Proportionality, Trade-off Analysis, power distribution

AMS subject classifications. 68M20, 97P30

Acronyms.

ACID	Atomicity, Consistency, Isolation, and Durability
BASE	Basically available, soft-state, and Eventually consistent
CPU	Central Processing Unit
DRAM	Dynamic Random Access Memory
DVFS	Dynamic Voltage and Frequency Scaling
EC	Energy Consumption
EE	Energy Efficiency
IoT	Internet of Things
LCS	Leveled Compaction Strategy
NoSQL	Not only SQL (Non-relational Database)
RAPL	Running Average Power Limit
RDBMS	Relational database management system
SLO	Service Level Objective
STCS	Size Tiered Compaction Strategy
TPC-H	Transaction Processing performance Council
WEC	Waiting Energy Consumption
YCSB	Yahoo! Cloud Serving Benchmark

1. Introduction. It is not easy to imagine human life without the internet, computers, and mobile applications in this modern era. Online services like e-commerce, online banking, and social networking have become part of our daily routine. Easy access and reducing the cost of internet access have attracted developers to

[*] Computer Science and Engineering Department, Nirma University, India (monika.shah@nirmauni.ac.in).
[†] Gujarat Technological University, India (amitdkothari@gmail.com)
[‡] Computer Science and Engineering Department, Pandit Deendayal Petroleum University, India (samir.patel@sot.pdpu.ac.in)

expand these services using cloud computing and IoT. With the expansion of such applications wondering across the world today, the complexity and dimensions of data transmitted are growing exponentially with each passing year. For 30 years, database software has advanced to deal with these challenges. NoSQL database covers the shortage of traditional databases [35] while widely used as Big Data storage recently [39]. A wide class of NoSQL databases is available to meet different applications' requirements. Researchers of NoSQL databases are still struggling to optimize their performance with the increased scalability and complexity of data. Generally, the NoSQL database executes over a distributed cluster system to support horizontal scalability, where the NoSQL database schedules jobs scheduled to different nodes of a given cluster. The designing perspectives of NoSQL databases and relational databases are different. Relational databases are popular for transactional applications, where updates and Delete are the most frequent operations. At the same time, NoSQL databases' main perspective is handling massive data records and availability. In NoSQL databases, Create and Read operations are most popular, and operations like Update and Delete are replaced through ″timestamp″or ″data version″.

A study presented in [2] shows past and projected world energy consumption, which highlights a continuous rise in energy consumption. Database, analytics, and IoT will be the fastest-growing applications [1]. Statistics given in [22] tells 71% of data centers are occupied for big data processing. Data centers are known to be energy-hungry infrastructure running internet-based services [12, 64]. The work depicted in the study [26] has also warned about the need to reduce energy consumption at data centers. As a result, there are continuous efforts instituted on hardware-level and operating system-level energy management of data centers [12, 64]. Simple models that work well for hardware may not work well for software [12]. Due to a lack of application information (like resource consumption, data access pattern, etc. [62], the operating system also becomes inadequate to provide a pro-active energy-aware solution. Database workload is different than other workload [44]. In addition, the database is such an exceptional application, which can expose alternate execution plans in advance. It is also reported that power consumption by back-end database services is higher than front-end web services [49]. Power consumption in the database system was evidenced well at all stages of development [3]. Therefore, modern databases are posing high requirements on energy efficiency in addition to other metrics [36]. Many researchers have applied energy-aware processing of relational database systems [29, 19, 51, 23, 32, 31]. Energy consumption of NoSQL database is the average electricity consumed by computing nodes of NoSQL clusters for executing some tasks. Despite the increasing share of NoSQL in the database world, a lack of energy consumption model has been observed. With the growing load on the data center, it is becoming difficult to maintain the NoSQL application's performance without both increasing the processing capacity of hosts and increasing the power budget of computing resources [57]. Increasing power consumption by NoSQLs and the trend of power economic development shows research direction to analyze and reduce power consumption by NoSQL.Energy consumption an

The major contribution of this comprehensive survey can be summarized as follow:

- Power distribution among various components of NoSQL servers.
- Classifying ECA of NoSQL databases to its modules like query processing, query optimization, data modeling, and configuration of cache structure, cloud patterns, consistency levels, and latency.
- Study of energy proportionality to understand energy efficiency scope and trade-off requirement.
- Summary of proposed energy conservation techniques.
- Analyze research scope to optimize the energy consumption of NoSQL systems.

The rest of the paper is organized as follows. Section 2 shows power distribution among various components like CPU, and memory. It also provides a summary of power monitoring tools used to analyze the energy consumption of NoSQL databases. Section 3 is the heart of the article, which provides a comprehensive survey on energy consumption analysis of various NoSQL functional modules. Section 4 discusses the proportionality study of energy with workload, performance, and latency. It helps to identify trade-off requirements between energy conservation and other metrics. Minimal efforts are found to reduce the NoSQL system's energy consumption. Section 5 describes these efforts. Based on the review, gaps, challenges, and some directions for research in energy conservation of NoSQL are discussed in section 6, and conclusion in Sect. 7. Fig. 1.1 demonstrates an overview of Energy Consumption Analysis of NoSQL.

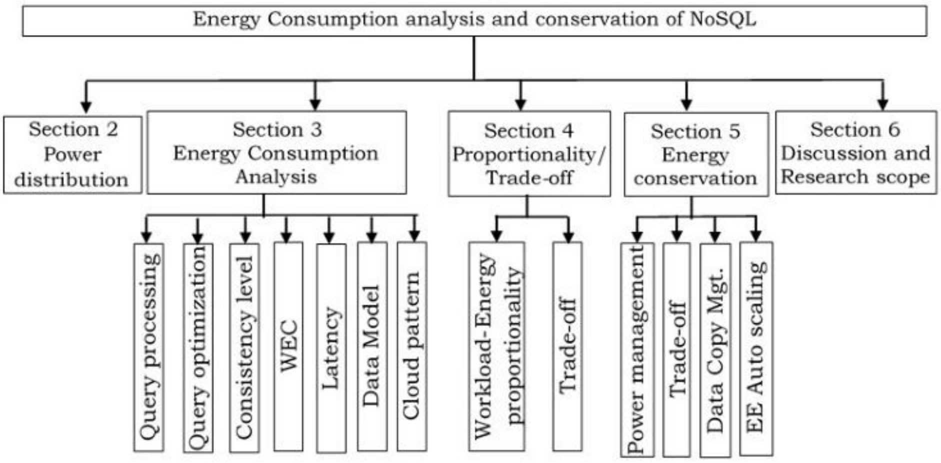
Energy Consumption analysis and conservation of NoSQL
Section 2 Power distribution
Section 3 Energy Consumption Analysis
Section 4 Proportionality/ Trade-off
Section 5 Energy conservation
Section 6 Discussion and Research scope
Query processing
Query optimization
Consistency level
WEC
Latency
Data Model
Cloud pattern
Workload-Energy proportionality
Trade-off
Power management
Trade-off
Data Copy Mgt.
EE Auto scaling

Fig. 1.1: An overview of Energy Consumption Analysis of NoSQL

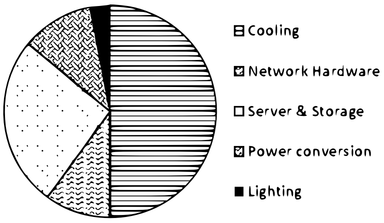

Fig. 2.1: Data Centre Power distribution

2. Power Distribution. Reliable measurement for each component of the system is an essential step toward sustainable energy consumption [11]. It helps to identify where power goes.It helps identify where power goes, and it can be helpful to researchers to identify components with unnecessary power supply and excessive power consumption. Therefore, this section provides a comprehensive survey on power distribution. Energy consumption can be presented as the product of power and execution time.

Fig. 2.1 shows energy consumption by different components of a data center, where the cooling system consumes more energy [12]. In a typical data center server, storage and network devices consume around 40%, 37%, and 23% of the total IT power, respectively [39]. 1 Watt of IT power saving can reduce 2.5 Watt in total power [39]. Data provided by Intel labs in [40], and the energy consumption survey of the data center presented in [12] reveal that a significant fraction of power consumed by a server is accounted for by the CPU, followed by the memory. NoSQL databases primarily use cluster setup. The component-level power distribution of the Cassandra cluster for the read-only and update-only workload is presented in [55, 54], which demonstrates two critical observations: i) In the Idle state, the highest power consumption is of components other than CPU and Memory. ii) The processor package adds the highest power consumption in the Active state. Static and dynamic power consumption analysis of in-memory databases presented in [28] also shows that dynamic power consumption of CPU and memory increases from 18% to 82%, and concludes that processor and memory consume significant energy during execution. Therefore, most NoSQL energy consumption analysis focuses on power consumed by CPU and Memory. Table 2.1 summarizes the components for which energy consumption is measured by literature work related to the energy consumption analysis of NoSQL.

Energy measurement can be done in three ways:i) Hadware-based, ii)Software-based, and iii) Hybrid. Hardware-based energy measurement has disadvantage of expensive and complex setup [11]. Reducing the

Table 2.1: Power consumption monitoring for components

Reference	CPU/ Processing Unit	DRAM /Memory	Entire System
[6]	✓	✓	
[14]	✓	✓	
[18]			✓
[34]			✓
[36]	✓	✓	
[37]	✓	✓	
[52]	✓	✓	
[53]	✓	✓	
[54]	✓		
[55]	✓		
[57]			✓

Table 2.2: Power monitoring tools

Reference	PowerAPI	jRAPL	Intel's RAPL	Power-meter	Other
[6]	✓				
[14]		✓			
[17]					API
[18]					Emeter software and EVM430-F6736 hardware
[34]				✓	
[36]					Log_power_to_file API
[37]					Log_power_to_file API
[52]		✓			
[53]	✓				
[54]			✓		
[55]			✓		
[57]				✓	

energy consumption of the NoSQL database system is a challenging task. The first challenge is to analyze energy consumption by different functional modules of NoSQL systems. Identifying suitable tools to measure the energy consumption of the NoSQL cluster is another challenge. Table 2.2 presents different approaches adopted by researchers to measure the energy consumption of NoSQLs.

Intel's Running Average Power Limit (RAPL) is a powerful tool that uses a software power model to estimate the energy consumption with the help of hardware performance counters and I/O models. It works for Intel processor architectures of Skylake, Haswell, Sandy Bridge, and Ivy Bridge. It provides accurate energy reading for CPU and RAM [25]. jRAPL is an API in java to monitor the energy consumption using RAPL. PowerAPI uses RAPL counters to provide power consumption information of each socket of the monitored machine [13]. In our literature survey, most articles use RAPL directly or through PowerAPI to measure the energy consumption of CPU and RAM components. To measure the energy consumed by the entire system, different meters like power-meter and Emeter are used in some research. Mahajan et al. have introduced a new API Log_power_to_file, which can measure the power consumed by major components like CPU, Disk, RAM, GPU, and Xeon Phi [36].

3. Energy Consumption Analysis of NoSQL functional modules.

It is well said that if you have a hammer, you tend to see every problem as a nail. If one is planning to

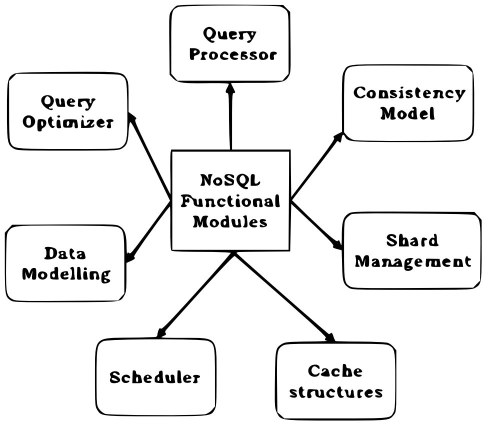

Fig. 3.1: Basic functional modules of NoSQLs

Table 3.1: Energy Consumption Analysis of NoSQL functional modules

Reference	Query optimization	Query Processing	Consistency	WEC	Latency	Data Modelling	Cloud pattern
[6]							✓
[14]	✓	✓					
[17]			✓				
[18]		✓					
[34]				✓			
[36]	✓						
[37]	✓						
[52]		✓					
[53]		✓				✓	
[54]		✓			✓		
[55]		✓			✓		
[57]					✓		

optimize the energy consumption of NoSQL, he tends to analyze the energy consumption of any functional module of NoSQL. Every NoSQL system has a different architecture but has a common subset of functional modules. Fig 3.1 shows the essential standard modules of NoSQL, where Data Modeling is the user interface module, and the rest are functional modules. This section presents the energy consumption analysis of various NoSQL modules. Despite being considered as a 'green system', NoSQL databases still lack mature solutions to evaluate and reduce Energy Consumption [34]. Only a few studies have analyzed the energy consumption of NoSQL. Table 3.1 presents a summary of our literature survey on energy consumption analysis of various functional modules of NoSQL databases.

There are more than 225 NoSQL databases introduced till now [45]. The webpage describes data model category-wise NoSQL databases. Major categories of NoSQL databases include Key-Value store, Wide column store, Document store, and Graph-based database. Every NoSQL has different architecture and functional modules. But, EC analysis is done on a limited set of NoSQL systems.

Table 3.2 shows a list of NoSQL for which energy consumption analysis is found in our literature survey. MongoDB is an example of a document store, Cassandra and HBase are column-store, Radis and Memcached are Key-Value stores, and Neo4j is a graph-based NoSQL. MongoDB and Cassandra are popular NoSQL for energy consumption analysis. These days many NoSQL databases are in-memory databases. So, this article also discusses the Energy consumption study of in-memory databases. The majority of the database operations

Table 3.2: Energy Consumption Analysis of NoSQL databases

Reference	MongoDB	Cassandra	HBase	Hive	Neo4j	Redis	Memcached
[6]	✓						
[14]	✓						
[17]			✓				
[18]	✓	✓				✓	
[34]		✓	✓	✓			
[36]	✓	✓					
[37]	✓	✓					
[52]					✓		
[53]	✓					✓	
[54]		✓					
[55]		✓					
[57]							✓

Table 3.3: Energy Consumption Analysis for Query operations

Reference	NoSQL	A	J	S	I	U	D	R	P	MR
[14]	MongoDB	✓	✓		✓					✓
[18]	MongoDB				✓	✓				
	Cassandra				✓	✓				
	Redis				✓	✓				
[34]	HBase	✓	✓		✓			✓	✓	
	Hive	✓	✓		✓			✓	✓	
	Cassandra	✓	✓		✓			✓	✓	
[37, 36]	Cassandra				✓	✓	✓			
	MongoDB				✓	✓	✓			
[52]	Neo4j	✓	✓	✓						
[53]	MongoDB	✓	✓		✓			✓		

A:Aggregate, J:Join, S:Sort, I:Insert, U:Update, D:Delete,
R:Range query, P:Pattern matching, MR:Map-Reduce

are memory-bound and waste computation power [44]. The impact of power management during various functions of the in-memory database on energy efficiency is presented in [28, 44], which is also included in this paper. Detailed energy consumption analysis of various NoSQL functional modules and resource management by in-memory databases are discussed in sub-sections.

3.1. Energy consumption analysis for Query Processing. Query processing is the heart of all database systems. It plays a key role in data analytics required for scientific or business intelligence. Therefore, the majority of energy consumption analysis of NoSQL systems is around query processing, which is presented in Table 3.3. It is infeasible to analyze the processing of each query of different applications. Instead, analysis of the type of queries and query operations would be a better choice. Although, at the initial stage of NoSQL Energy consumption analysis, researchers have started analyzing EC of a list of queries. Analysis of the energy consumption during query processing is presented at [53, 52, 14] for their query list, where [14] monitors EC of queries using aggregate and same queries using map-reduce operations. Similarly, the work represented at [37, 36] shows EC during query processing of some queries with an alternate query. The Energy consumption of Insert, Read, and Update operations for the increasing workload is presented at [18, 55]. Data access with patter analyzes energy consumption of Insert, Read, and Update operations. The energy consumption of Operations required for query access(Grep, Select), and for analytical access(Aggregate, Join) have been compared for different NoSQLs at [34].

Table 3.4: Expensive Query operations in terms of energy

Reference	MongoDB	Cassandra	Redis	Neo4j	HBase	Hive
[14]	Map-Reduce					
[18]	Insert, Update	Read	Insert			
[34]		Aggregate, Join, Grep			Aggregate, Join, Range Query, Reduce side Join	Aggregate, Join
[37, 36]	Update, Aggregate	Insert, Update, Aggregate, Search				
[52]				Join, Aggregate		

This paper has mapped query operations with test queries used in experiments conducted at various research papers. Table 3.3 describes a list of query operations analyzed and the corresponding NoSQL used for analysis. This summary helps researchers to identify further research scope in energy-efficient query processing. It shows the popularity of Insert, Aggregate, and Join operations. MongoDB is one of the widely used NoSQL, which is ensured from usage-based database ranking [24]. It might be an attraction point for more researchers to analyze energy consumption on MongoDB queries. Aggregate, Join, Insert, and Range queries are most popular. There are two groups of the survey found in this domain. One group has compared the impact of query operations on energy consumption by selected NoSQL with a relational database. At the same time, other groups have compared EC of query operations among various NoSQLs only. Both aspects will help find the scope of optimizing query processing of NoSQL in the context of energy consumption. Other perspectives may include a selection of the most energy-efficient NoSQL matching the application need and choosing alternate operations to get the query to execute.

Neo4j(NoSQL) consumes more energy than a relational database, especially for aggregate and join operations [52]. The work presented in [37, 36] reports the impact of query operation on energy consumption for simple dataset YCSB and complex dataset of Twitter. Both NoSQL (Cassandra and MongoDB) are found energy economical in comparison to MySQL for all test queries on YCSB. In contrast, their experiment result on Twitter data varies for different query types. For Insert and Aggregate query on Twitter data, MySQL consumes less energy consumption than both Cassandra and MongoDB. For Update, Delete, and simple Search queries on Twitter data, MongoDB consumes less energy than MySQL. Cassandra is found expensive in terms of energy spent for all types of queries except update queries on Twitter. The work also compares energy consumptions by query processing among different NoSQL databases (MongoDB and Cassandra). The result reveals that MongoDB is economical compared to Cassandra in the context of energy consumption by query processing.

The energy consumption by basic query commands (Insert, Read, Delete) on a different category of NoSQL have been compared in [18]. Cassandra is chosen as a column-oriented data store, MongoDB is taken as a document-oriented data store, and Redis is taken as Key/Value pair. Their result shows Cassandra consumes much more energy for the read operation, while Redis is more expensive at Insert queries. MongoDB consumes more power for the workload (1000 to 10000 operations) of Insert and Update than Redis and Cassandra. With increasing workload, energy consumption by MongoDB is reduced. In MongoDB, aggregate pipeline usage is found economic in both terms of energy and performance in comparison to map-reduce operations [14]. Table 3.4 summarizes the energy consumption analysis of query operations. It highlights query operations for each testbed NoSQL that consume more energy.

3.2. Energy consumption analysis for Query Optimization. Query Optimization is a pivotal component for any efficient database design. A huge amount of queries are executed daily. So, optimizing each

Table 3.5: Greenup Scenarios of optimization

Category	GreenUp	PowerUp	SpeedUp
1	> 1	1 (Power saving)	1 (Improve performance)
2	> 1	$= 1$ (No change in Power)	1 (Improve performance)
3	> 1	1 (Increase Power)	1 (Improve performance)
4	> 1	1 (Power saving)	1 (Degrade Performance)

query a little may help to improve the throughput and energy efficiency of a system respectively to a great extent. Unlike other software, a database system is the only software that can explore alternative ways of executing the query. The cost can also be query response time, latency, and energy consumption. Unlike NoSQL, a lot of research is done on optimizing queries to improve the energy efficiency of relational databases. In our information, [4] was the first to propose energy-aware query optimization. Later much other work on energy-aware Query Optimization was discussed for relational databases. [62, 61, 60, 58, 64, 22, 43, 20, 23].

Despite the continuous increasing usage of NoSQL databases, only [37] and its extended article [36] are found in our literature review that studies the impact of query optimization on the energy efficiency of NoSQL. The work depicted in [37, 36] illustrates some well-known basic query optimizations techniques for Cassandra and MongoDB. It also analyzes the impact of query optimization on power, performance, and energy efficiency. They believe that result of query optimization can be any of three or a combination of them: improve performance (Speedup >1), Power saving (PowerUp < 1), or improve energy efficiency (GreepUp > 1). They have tried to analyze four scenarios of GreenUp (Energy efficiency) as described in Table 3.5.

Category 1 is an ideal scenario, where improvement in both performance and power consumption results in energy saving. On another side, due to lack of energy consumption awareness, the 4th category is rarely observed. For MongoDB, their results say that using index fields on predicate for delete and covered query and project phase in aggregate query can help to achieve speedup with power saving. In contrast, sharding on multiple servers can improve performance at the cost of higher power consumption. MongoDB provides options to perform the write operation in bulk, and it can implement the bulk Insert operation either ordered or unordered. The general understanding is that the system implements unordered bulk write in parallel fashion and the ordered write operation in serial. Surprisingly counter-intuitive results that unordered write degrades performance and energy efficiency.

For Cassandra, row caching, LCS for read-heavy queries, and STCS optimizations for Insert-heavy queries also improved both performance and energy efficiency. They conclude that energy efficiency can be improved significantly on both MongoDB and Cassandra without degrading performance, but the improvement rate of energy efficiency is not linearly proportional to the speed of performance improvement. There are also scenarios where query optimization techniques may not be helpful for either performance improvement or energy efficiency improvement. Finally, they conclude three points: i) Query optimization can achieve energy efficiency for Cassandra and MongoDB without compromising performance. ii) Energy optimization is not always linearly proportional to performance optimization. And iii) query optimization technique may not optimize performance and decrease power all the time.

There is no other work in the literature that explicitly studies the impact of query optimization on energy consumption. Some work compares the energy consumption of queries using alternate ways of processing. Its result analysis can also help to redesign query optimization algorithms of NoSQLs. The work represented in [53], compares the energy consumption of sample queries on MongoDB with and without index. They conclude that the use of indexes helps to reduce energy consumption in most cases. Their other conclusion is application-level joins consume less energy than NoSQL-level joins. The work depicted in [14] describes energy consumption analysis for Insert and TPC-H (1,5,10,15,20) queries with and without index, with map-reduce, and with aggregate pipeline functions on MongoDB document store. Their result shows that the use of the Aggregate pipeline is more effective than the complex map-reduce process, and the use of the index is more effective for most queries with few exceptions like TPC-H query 1.

3.3. Energy consumption analysis of Consistency levels. None other than [17] article from our literature survey analyzing the impact of strong and eventual consistency on energy consumption and concurrency. The experiment comprises of 3 workloads(1. Write intensive(80% write), Read intensive(80% read), and mixed (50%read 50%write) that fully stress memory and exercise hard-disks on a columnar store HBase by applying the semantics of YCSB benchmark. They simulated these three types of workload over two configurations: i) Buffer based to simulate eventual consistency, which is the default configuration in HBase ii) Without Buffer to simulate eventual consistency.

The result reveals that strong consistency costs more in terms of energy on write-intensive workload, and eventual consistency costs more for the read-intensive and mixed workload. Finally, they think that change of request patterns by avoiding requests to unused disks and using caching to save energy consumption.

3.4. Waiting Energy Consumption Analysis . There exists much work analyzing the energy consumption of different software. A novel approach of reducing WEC to reduce energy wastage of NoSQL is proposed in [34]. WEC is one of the factors causing energy wastage due to computer idleness. It defines WEC as the energy wasted when some nodes are in a ”passive idle” or ”busy idle” state due to waiting for other resources.

The work chose four NoSQL databases (HBase, Cassandra, HadoopDB, and Hive), five types of queries (loading, and 4 query operations like fuzzy search, range search, aggregate, and join) to analyze WEC. The experiment result shows that NoSQL databases using a non-relational data model (like HBase and Cassandra) have high local and network I/O operations. NoSQL databases are I/O intensive, and the performance of the CPU is much higher than I/O operations done by the disk and network card. Hence, the CPU needs to wait for longer and more waiting energy consumption is produced. The paper describes that many NoSQL databases use the Map-Reduce model for query operations (like Selection, Aggregation, and Fuzzy Selection) and explicit use of Map Reduce. The result of the paper reveals that inappropriate Map Reduce can cause poor parallelism and poor synchronization, which generate waiting energy reduction.

One of the solutions to reduce WEC is to shut off idle systems to reduce energy wastage by idle nodes of a database cluster. But, this solution may not work for a NoSQL-like distributed system. Nodes cannot be shut off when they are temporarily idle but waiting for job scheduling, I/O operations, or computational results from other nodes. Hence, they suggest reducing waiting for energy consumption by lowering network I/O, synchronizing CPU and I/O operations, and proper Map-reduce framework selection considering data features.

3.5. Energy consumption analysis of Latency levels. Increasing reliance on the cloud has led to scale-out workloads, which are latency-sensitive. NoSQL servers need massive infrastructures to satisfy latency constraints, which consume more energy [6, 55, 57]. The work represented in [6, 55] describes the power consumption of the Cassandra cluster at 95th and 99th percentile latency for the read-only and write-only workload. Their result shows that read-only workloads need more energy consumption to maintain the 99th percentile compared to the 95th percentile latency, while update-only workloads do not need more energy to satisfy 95th to 99th percentile latency. They have also shown the impact of power provisioning and resource provisioning on energy saving, and their results say that resource provisioning can save more energy than power provisioning.

3.6. Energy consumption analysis of Data Modeling. Data Model provides a database user with a conceptual framework in which developers can specify the database requirements and structure to satisfy these requirements. The Document store NoSQL is one of the most popular NoSQL. It provides flexibility to choose data structures to represent data and their relationships. Parent Embedding, Child Embedding, Parent Referencing, Child Referencing, Two-way Embedding, Two-way Referencing, Bucketing, and De-normalization are primitive data models to specify any database. Data modeling influences query performance, consistency, and maintainability. Despite that, no work analyzes the impact of data modeling and schema design on energy consumption to our best knowledge. Only one work [53] has initiated investigating the effect of adopting these data models to design schema and energy consumption by executing queries on these schemas. Their result conveys some messages: i) There is no schema uniformly performing best for all queries, ii) No schema nor data models are consuming less energy for all queries. They suggest choosing a suitable data model based on required queries.

3.7. Energy consumption analysis of Cloud patterns for Database. With the increasing trend of Internet and Cloud computing, the inclination of many companies is toward cloud-based applications. Relational databases and NoSQL are two well-known database families used as the backbone of these cloud-based applications. Developers prefer to use cloud patterns to configure database systems to benefit from best practices [16]. Despite the wide adaptability of cloud patterns, only one work found in the literature studies the energy consumption of NoSQL while adopting cloud patterns. Therefore, this section discusses analysis done to study the impact of cloud patterns of NoSQL database systems on energy consumption.

The work depicted in [6] presents the impact of energy consumption of three cloud patterns [16]: Local Sharding Based Router, Local Database Proxy, and Priority Message Queue, with three databases: two popular relational databases (PostgreSQL and MySQL), and one NoSQL (MongoDB). In Local Database proxy, data is replication among a master node and the slave nodes, a proxy component route read requests and write requests to appropriate master and slave nodes. NoSQL database system is an excellent example of a distributed system. The majority of the NoSQL database uses the master/slave replication model, where every data chunk has one master copy, and other copies spread to other nodes are known as a slave. When the client requests for reading/write operations, the proxy's responsibility is to route all write operations to master and read procedures to slaves. With increasing data volume, NoSQL facilitates by splitting the database into multiple databases called shards. There are two well-known methods of sharding applied on shard keys: range-based sharding and hashing-based sharding. The local router routes a request to access the data in the Local Sharding-Based router. Priority Message Queue pattern is known for allowing asynchronous communication between components. It helps to improve the scalability of applications by supporting loosely coupled design. Priority Message Queue generally deals with different types of messages. They report the contrasting result. MongoDB executes faster than MySQL and consumes more energy than MySQL in cloud-based applications designed without adopting cloud patterns. Their results show that different cloud patterns impact relational and NoSQL database systems. Local Database Proxy improves the significant energy efficiency of MySQL while consuming more energy consumption on MongoDB. In adopting the Local Sharding Based Router pattern, the Modulo strategy strongly affects MongoDB, but a small for MySQL. Consistent and Lookup Strategy of Local Sharding Based router.

4. Proportionality and Trade-off.

4. Proportionality and Trade-off. Most research works aim to make execution more and more fast. The fastest response is never the target of any application. Instead, desired performance expectations are specified using response time, latency, or throughput. On another side, Energy conservation is one of the significant research focuses these days. Energy consumption E of a NoSQL cluster of N nodes for the T period can be defined as shown in Eq. 4.1, where $P_i(t)$ is the power consumption of node i at time t.

$$E\left(T\right) = \sum_{i=1}^{N} \int_{0}^{T} P_i\left(t\right) dt \tag{4.1}$$

$$E\left(T\right) = P_{avg}\ X\ T \tag{4.2}$$

The cloud pattern analysis work depicted in [27] describes some essential points: i) Lookup and consistent hashing can improve energy consumption and performance. ii) Modulo algorithm does not improve performance or energy efficiency. iii) The pattern 'Local Database Proxy' can significantly improve the energy efficiency of cloud applications, while the pattern 'Local Sharding- based Router' combined with 'Local Database Proxy' can also improve response time without compromising energy efficiency.

There are two conflicting views showing the relation between energy and performance. One class of researchers believe that energy optimization comes byproduct of performance optimization, and another type of researcher believes that optimizing energy and performance are two conflicting targets. The first case can be verified by analyzing power consumption proportionality with performance measured. If energy is proportional to performance, there are two options to reduce energy consumption: i) Energy optimization as a result of performance optimization e.g. The application should take shortest execution time to achieve highest energy efficiency [11], ii) Select alternate operations that consume less power. Otherwise, a trade-off decision is required. Therefore, this section discusses the energy-performance proportionality study and trade-off analysis on NoSQL systems.

Table 4.1: Summary of Survey on Trade-offs in NoSQL

Reference	NoSQL	Metrics Analyzed for Trade-off
[37]	MongoDB, Cassandra	Performance (Query Response Time) and Energy Consumption
[36]	MongoDB, Cassandra	Performance(Query Response Time) and Energy Consumption
[55, 54]	Cassandra	Performance(Latency) and Energy Consumption
[17]	HBase	Consistency Level and Energy Consumption

4.1. Proportionality. Energy efficiency and Energy proportionality are major concerns today. As a result of researchers' effort, idle power consumption of database servers is reduced from 50% (in 2010) to 20% (today) of peak power consumption, which shows the trend of energy proportionality [28]. The energy-workload proportionality of Cassandra cluster for read-only workload and update-only workload is presented in [55, 54]. They exhibit poor energy proportionality in both workload types for all components except processor - CPU. CPU component is more energy proportional in read-only workload than update-only workload. The power consumption range for CPU and processor packages is from 30-100% and 55-100% in read-only workload and 78-100% and 82-100% in an update-only workload.

Dynamic power provisioning and resource provisioning are well-known techniques to control power consumption and resource allocation when the system is underloaded. The work also exhibits the impact of power provisioning and resource provisioning on the energy proportionality of the Cassandra cluster, where resource provisioning is much more effective. They have proposed hybrid provisioning (power provisioning and resource provisioning) technique to improve the energy proportionality to the next level.

Heterogeneous consumption of disks and memory instability usually causes power dis-proportionality of storage systems [17]. A comparison of energy consumption and execution time of various query operations is described in [14, 34]. It has observed almost the exact relationship between energy consumption and execution time. For example, HBase and Cassandra consume more energy as well as more execution time than Hive and HadoopDB for Loading, Grep, Selection, Aggregate, and Join operations [34]. Map-Reduce operation consumes more energy as well as execution time than Aggregation [14].

4.2. Trade-off. This section provide comprehensive survey on Trade-off in NoSQL databases. Fig. 4.1 summarize summary of Trade-off analysis done over different NoSQLs. Energy consumption cannot be considered independently of performance delivered by the system as they directly relate to each other [34, 48]. One view is that high performance costs high energy. So, it may be required to compromise performance for reducing energy consumption. Developing software techniques to achieve energy and performance trade-offs is one of the current research trend [8]. SLOs may specify performance requirements to decide the trade-off between performance and energy.

Most database researchers believe that a trade-off between power and performance is inevitable. There is a belief that many scenarios need to be analyzed for NoSQLs to understand the trade-off between energy optimization and performance optimization. The work shown in [36] is only work in our opinion analyzing the trade-off requirement between energy optimization and query optimization for the NoSQL database. They demonstrated a few well-known query optimization approaches on NoSQL databases (Cassandra and MongoDB) to analyze the impact of query optimization on energy optimization and performance optimization. For this, they have evaluated Powerup, SpeedUp, and GreenUp. If all these three conditions satisfy every time, energy optimization comes along with performance optimization. Conversely, if GreenUp and SpeedUp conditions are not satisfied for all scenarios without degrading PowerUp, then the performance optimization and energy optimizations seem two different goals. They showed that energy optimization is neither byproduct nor a conflicting goal of performance optimization in some situations and concluded that energy efficiency does not always scale linearly with performance. Hence, it shows research scope to analyze in detail trade-offs.

Latency SLOs are common these days. Performance targets (Latency SLOs) are typically based on 99th or 95th percentile in place average latency. This type of SLOs provides us an opportunity to trade performance for power [55]. They have also demonstrated that compromising latency from 99th%-ile to 95th%-ile can also reduce the energy consumption of the Cassandra cluster. This work shows research avenues to make energy proportionality systems as wastage of energy cannot be ignored today, where the use of the energy-hungry technical device is continuously increasing.

Trade-offs between consistency level and energy for HBase NoSQL are analyzed in [17]. Strong consistency has better throughput-energy proportionality for all types of workload. On another side, eventual consistency shows better throughput-energy proportionality for the read-intensive and balanced workload. But, it is not proportional at under-loaded (low throughput). It also offers a trade-off scenario at write workload, where strong(high) consistency results at the cost of spending more energy.

5. Energy Conservation in NoSQL. Reducing the energy consumption of NoSQL or improving the energy efficiency of NoSQL requires proper knowledge of energy consumption by NoSQL. section 3 describes a little effort on the energy consumption analysis of NoSQL. During the literature survey, only four papers were found that propose techniques to reduce the energy consumption of NoSQL. All these papers touch on different functional modules of NoSQL. It includes optimizing energy proportionality [55], the trade-off between memory performance and power [57], low-power database server for IoT [47], and auto-scaling of virtual data centers [9].

The total system energy of the Cassandra cluster is poorly proportional to workload, especially when the system is underloaded [55]. The work presented in [55] investigates the effect of power management techniques on energy proportionality, where resource provisioning results better compared to power provisioning techniques. To improve energy proportionality to the next level, they have proposed hybrid (resource + power) provisioning and trade latency by considering the difference between measured latency and SLO. The proposed hybrid provisioning with 95%-ile latency delivers the most power-saving (up to 55%.)

A hardware-software unified solution offers a trade-off memory performance with power. Lake [57] : a Low Latency, power-efficient Key-value store design to improve power efficiency. They present a multi-level multi-core cache design after exploring the trade-off between performance and power by leveraging different types of on-chip and onboard memories. It claims low latency (1.1μs on hit) and better throughput (13.1Mqps) at the cost of 10W additional power.

The work depicted in [47] proposes a methodology to construct a low-power database server for IoT middleware. The work uses Raspberry Pi, and MongoDB. It presents two designs: i) Non-data Copy Oriented method: The client sends a request to the master node. The Master node finds a data node matching the client request and sends it an activating signal. Then, the client node will send data to the master node. ii) Data copy oriented from data node to the master, where each data node periodically transfers data to the master node. Here, the master node should have ample storage.

6. Discussion and Future Directions. NoSQL systems are used widely for the development of web applications, data analytics, and IoT systems. NoSQLs have started taking place of RDBMSs to serve better scalability, availability, performance, and variety of data handling. This section describes gaps or limitations observed in related work done, future research directions, and challenges to do energy consumption research on NoSQL.

Benchmark consumption is the biggest issue for research in NoSQL. Despite the wide use of NoSQL, researchers could not find a suitable benchmark for NoSQL except YCSB [41]. YCSB framework provides a set of test cases combined by Insert, Read, Update and Scan operations [36], which could be adopted to measure the performance of NoSQL databases [10]. Still, these test cases only involve too simple database operations [34].

Table 6.1 describes benchmarks used to test the energy consumption of NoSQL. Most research on NoSQL uses YCSB and TPC benchmarks. TPC benchmarks are designed to test relational databases' performance and are far away from NoSQL databases. For example, i) TPC-benchmark supports ACID, and NoSQL supports the BASE. ii) TPC-H contains complicated queries containing Join, Group, and Aggregate operations. While, many NoSQLs do not support explicit interfaces for Join, Group-by, and Aggregate operations. SSB is available to test the performance of the data warehouse, and SSB design is based on TPC-H only. Looking toward the distinct feature of NoSQL and relational databases, it is not preferred to apply any benchmark of the relational database to non-relational databases. In addition, every NoSQL also has different characteristics. Another problem is that most benchmarks available for databases are to test performance. TPC-Energy is a benchmark for a relational database, which allows examining the energy consumption of the servers [27]. Hence, it is

Table 6.1: Benchmarks

	YCSB	Twitter	TPC-H	Other
[6]				DVD store [15], JPteSTore [30]
[9]	✓			
[14]			✓	
[17]	✓			
[18]	✓			
[28]				TATP(OLTP) [21], SSB(OLAP) [46]
[34]				Own
[36]	✓	✓		
[37]	✓			
[52]				Adventure Data Warehouse [38]
[54]				Cloudsuit [50]
[55]	✓			

required to design benchmarks to test NoSQL servers' performance and energy consumption. Table 6.2 briefly describes other gaps or limitations observed in the literature survey and shows some future directions.

7. Conclusion. Despite the wide use of NoSQL and knowing NoSQL consumes more energy consumption, NoSQL is lagging for energy conserving optimization. Therefore, this article presents a comprehensive survey on energy consumption analysis for NoSQL databases. This paper classifies the analysis work as per NoSQL functional modules. This work collects results from different papers and generates various summaries in tables. It includes components consuming significant power, a list of NoSQL analyzed, a list of NoSQL functional modules analyzed for its EC, a list of query operations monitored, and query operations consuming high energy. Little effort is made to reduce the energy consumption of NoSQL, which is also presented here. Finally, the paper discusses gaps in the articles surveyed, summarizes recommendations from the articles, and directs future research scope toward conserving energy consumption of NoSQL systems.

REFERENCES

[1] *Cisco Global Cloud Index: Forecast and Methodology, 2016–2021 White Paper.* `https://techie.buzz/2019/03/19/cisco-global-cloud-index-forecast-and-methodology-2016-2021-white-paper/`, 2019. Accessed:2019-03-29.

[2] *Work, energy, and energy resources.* `https://opentextbc.ca/openstaxcollegephysics/chapter/world-energy-use/`. Accessed:2022-03-29.

[3] R. Agrawal, A. Ailamaki, P. A. Bernstein, E. A. Brewer, M. J. Carey, S. Chaudhuri, A. Doan, D. Florescu, M. J. Franklin, H. Garcia-Molina, et al., *The claremont report on database research*, ACM Sigmod Record, 37 (2008), pp. 9–19.

[4] R. Alonso and S. Ganguly, *Energy efficient query optimization*, in Matsushita Info Tech Lab, Citeseer, 1992.

[5] B. Bani, *Understanding the impact of databases on the energy efficiency of cloud applications*, PhD thesis, Ecole Polytechnique, Montreal (Canada), 2016. Thesis.

[6] B. Bani, F. Khomh, and Y.-G. Guéhéneuc, *A study of the energy consumption of databases and cloud patterns*, in International Conference on Service-Oriented Computing, Springer, 2016, pp. 606–614.

[7] R. Bolla, R. Bruschi, and P. Lago, *The hidden cost of network low power idle*, in 2013 IEEE International Conference on Communications (ICC), IEEE, 2013, pp. 4148–4153.

[8] B. Bylina, J. Potiopa, M. Klisowski, and J. Bylina, *The impact of vectorization and parallelization of the slope algorithm on performance and energy efficiency on multi-core architecture*, in 2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS), IEEE, 2021, pp. 283–290.

[9] E. Casalicchio, L. Lundberg, and S. Shirinbab, *Energy-aware auto-scaling algorithms for cassandra virtual data centers*, Cluster Computing, 20 (2017), pp. 2065–2082.

[10] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, *Benchmarking cloud serving systems with ycsb*, in Proceedings of the 1st ACM symposium on Cloud computing, 2010, pp. 143–154.

[11] D. Davidović, M. Depolli, T. Lipić, K. Skala, and R. Trobec, *Energy efficiency of parallel multicore programs*, Scalable Computing: Practice and Experience, 16(4) (2015), pp. 437–448.

[12] M. Dayarathna, Y. Wen, and R. Fan, *Data center energy consumption modeling: A survey*, IEEE Communications Surveys & Tutorials, 18 (2015), pp. 732–794.

Table 6.2: Gap Analysis and Future Direction

Key element	Gap analysis and future directions
RDBMS and NoSQL	• Energy and Performance difference analysis between NoSQL and RDBMS is available for a limited category of NoSQL only. Researchers can explore similar research for different types of NoSQL(like Document Oriented, Graph-based, Key-Value Pair, Column Family, etc.). • NoSQL systems consume more energy than relational databases [6, 52, 37, 36], where they also use similar deployment setup for both NoSQL and RDBMS. But, Which component or function of NoSQL consumes more energy and Why? are still in the research scope. • An empirical study in [6] describes that despite a similar setup (deployment to a distributed cluster) in RDBMS and MongoDB, query response time and energy consumption patterns conflict with each other. Therefore, It is suggested to explore a relationship between energy consumption and utilization of each component of each cluster node in active and idle states both.
Power monitoring	• Usually, it is a practice to use a distributed cluster setup for NoSQL deployment. But, section 2 does not show power consumption analysis of network resources nor specify a reason to ignore it. • Although the introduction to Waiting Energy Consumption in [34], a method of monitoring waiting energy is still unknown.
Power Wastage	• Energy-workload proportionality and WEC study seem promising techniques to identify power wastages. Power management is suggested for the underloaded situation [55]. A comprehensive survey on power management techniques are presented in [56]. • Traditional resource provisioning algorithms may not be directly applicable to NoSQL. Is it possible to design resource provisioning algorithm for a distributed NoSQL, which do not deteriorate other quality of service like scalability, availability, etc • Waiting energy consumption analysis presented in [34] recommends reducing CPU waiting time and WEC by optimizing data format, storage and selecting appropriate scheduling algorithms. • Analysis done to improve power efficiency and latency in [57] strongly suggest to reduce speed gap between network I/O and Computation.
Query Optimization	• The work presented in [37, 36] have explored the impact of query optimization on energy efficiency for very limited query operations and with a single alternate query plan of MongoDB and Cassandra. The work can be extended for various types of query operations along with promising plans.
Data Modeling	• There are many factors like query structure, data model, index usage, and query optimization in document stores that affect query performance [36]. Data modeling is a process to define and structure data elements in the context of the relevant application. Only one article draws our attention to data structuring [53]. Proposing data structures considering their impact on energy consumption will be a great contribution to the energy conservation of NoSQL. • [36] have observed the different impacts of query processing on different datasets, where datasets were models using two different data models i.e. Denormalized, Child Embedding. It shows need of analyzing energy consumption impact of different query patterns on possible set of data models.
Tools	• Like performance profiler, energy profiler tools can be designed to generate the energy consumption profile of a distributed NoSQL and highlight energy hotspot modules.
Energy Conservation	• Energy conservation of NoSQLs is still in future scope as proper energy consumption analysis is fundamental for it.

[13] A. d'Azemar, ldesauw, B. Jordan, G. Fieni, and Guillaume, *Powerapi-ng/rapl-formula.*

[14] D. Duarte and O. Belo, *Evaluating query energy consumption in document stores*, in International Conference on Emerging Technologies for Developing Countries, Springer, 2017, pp. 79–88.

[15] *Dell dvd store database test suite.* `http://linux.dell.com/dvdstore/`, 2011. Accessed:2022-03-29.

[16] C. Fehling, F. Leymann, R. Retter, D. Schumm, and W. Schupeck, *An architectural pattern language of cloud-based applications*, in Proceedings of the 18th Conference on Pattern Languages of Programs, 2011, pp. 1–11.

[17] Á. García-Recuero, *On the energy efficiency of client-centric data consistency management under random read/write access to big data with apache hbase*, arXiv preprint arXiv:1509.02640, (2015).

[18] C. Gomes, E. Tavares, and M. N. d. O. Junior, *Energy consumption evaluation of nosql dbmss*, in Anais do XV Workshop em Desempenho de Sistemas Computacionais e de Comunicação, SBC, 2016, pp. 71–81.

[19] R. Gonçalves, J. Saraiva, and O. Belo, *Defining energy consumption plans for data querying processes*, in 2014 IEEE Fourth International Conference on Big Data and Cloud Computing, IEEE, 2014, pp. 641–647.

[20] G. Graefe, *Database servers tailored to improve energy efficiency*, in Proceedings of the 2008 EDBT workshop on Software engineering for tailor-made data management, 2008, pp. 24–28.

[21] I. S. Group, *Ibm software group information management, telecom application transaction processing (tatp) benchmark description.* `http://tatpbenchmark.sourceforge.net/TATP_Description.pdf`. Modified Date: 2009-03-27 Accessed:2022-03-29.

[22] B. Guo, J. Yu, B. Liao, D. Yang, and L. Lu, *A green framework for dbms based on energy-aware query optimization and energy-efficient query processing*, Journal of Network and Computer Applications, 84 (2017), pp. 118–130.

[23] S. Harizopoulos, M. Shah, J. Meza, and P. Ranganathan, *Energy efficiency: The new holy grail of data management systems research*, arXiv preprint arXiv:0909.1784, (2009).

[24] S. IT, *Db-engines ranking.* `https://db-engines.com/en/ranking`. Accessed:2022-03-29.

[25] K. N. Khan, M. Hirki, T. Niemi, J. K. Nurminen, and Z. Ou, *Rapl in action: Experiences in using rapl for power measurements*, ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS), 3 (2018), pp. 1–26.

[26] B. Khargharia, S. Hariri, and M. S. Yousif, *Autonomic power and performance management for computing systems*, Cluster computing, 11 (2008), pp. 167–181.

[27] F. Khomh and S. A. Abtahizadeh, *Understanding the impact of cloud patterns on performance and energy consumption*, Journal of Systems and Software, 141 (2018), pp. 151–170.

[28] T. Kissinger, D. Habich, and W. Lehner, *Adaptive energy-control for in-memory database systems*, in Proceedings of the 2018 International Conference on Management of Data, 2018, pp. 351–364.

[29] M. Kunjir, P. K. Birwa, and J. R. Haritsa, *Peak power plays in database engines*, in Proceedings of the 15th International Conference on Extending Database Technology, 2012, pp. 444–455.

[30] J. Landis, K. Shimizu, E. Macarron, I. AVE, G. Gotimer, R. Bala, I. Baiborodine, K. Zero, H. Boutemy, and S. Tripodi, *Mybatis jpetstore.* `https://github.com/mybatis/jpetstore-6`. Accessed:2022-03-29.

[31] W. Lang, R. Kandhan, and J. M. Patel, *Rethinking query processing for energy efficiency: Slowing down to win the race.*, IEEE Data Eng. Bull., 34 (2011), pp. 12–23.

[32] W. Lang and J. Patel, *Towards eco-friendly database management systems*, arXiv preprint arXiv:0909.1767, (2009).

[33] J. B. Leverich, *Future scaling of datacenter power-efficiency*, Stanford University, 2014.

[34] T. Li, G. Yu, X. Liu, and J. Song, *Analyzing the waiting energy consumption of nosql databases*, in 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, IEEE, 2014, pp. 277–282.

[35] Y. Li and S. Manoharan, *A performance comparison of sql and nosql databases*, in 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), IEEE, 2013, pp. 15–19.

[36] D. Mahajan, C. Blakeney, and Z. Zong, *Improving the energy efficiency of relational and nosql databases via query optimizations*, Sustainable Computing: Informatics and Systems, 22 (2019), pp. 120–133.

[37] D. Mahajan and Z. Zong, *Energy efficiency analysis of query optimizations on mongodb and cassandra*, in 2017 Eighth International Green and Sustainable Computing Conference (IGSC), IEEE, 2017, pp. 1–6.

[38] MashaMSFT, M. Ghanayem, JulieMSFT, D. Coulter, KatsutoshiOtogawa, J. Parente, M. Ray, T. Petersen, J. Wells, and J. Rytlewski, *Adventureworks sample databases.* `https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15&tabs=ssms`. Accessed:2022-03-29.

[39] F. Mehdipour, H. Noori, and B. Javadi, *Energy-efficient big data analytics in datacenters*, in Advances in Computers, vol. 100, Elsevier, 2016, pp. 59–101.

[40] L. Minas and B. Ellison, *Energy efficiency for information technology: How to reduce power consumption in servers and data centers*, Intel press, 2009.

[41] M. J. Mior, K. Salem, A. Aboulnaga, and R. Liu, *Nose: Schema design for nosql applications*, IEEE Transactions on Knowledge and Data Engineering, 29 (2017), pp. 2275–2289.

[42] S. Mittal, *Power management techniques for data centers: A survey*, arXiv preprint arXiv:1404.6681, (2014).

[43] R. Niemann, *Towards the prediction of the performance and energy efficiency of distributed data management systems*, in Companion Publication for ACM/SPEC on International Conference on Performance Engineering, 2016, pp. 23–28.

[44] S. Noll, H. Funke, and J. Teubner, *Energy efficiency in main-memory databases*, Datenbank-Spektrum, 17 (2017), pp. 223–232.

[45] *List of nosql database management systems.* `https://hostingdata.co.uk/nosql-database/`. Accessed:2022-03-29.

[46] P. O'Neil, B. O'Neil, and X. Chen, *Star schema benchmark.* `https://www.cs.umb.edu/~poneil/StarSchemaB.PDF`. Accessed:2022-03-29.

[47] P. Paethong, M. Sato, and M. Namiki, *Low-power distributed nosql database for iot middleware*, in 2016 Fifth ICT

international student project conference (ICT-ISPC), IEEE, 2016, pp. 158–161.

[48] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, *Load balancing and unbalancing for power and performance in cluster-based systems*, (2001).

[49] M. Poess and R. O. Nambiar, *Energy cost, the key challenge of today's data centers: a power consumption analysis of tpc-c results*, Proceedings of the VLDB Endowment, 1 (2008), pp. 1229–1240.

[50] P. Pourhabibi, Javier, M. Otto, D. Ustiugov, ivonindza, C. Lin, and aledaglis, *Cloudsuite*. `https://github.com/parsa-epfl/cloudsuite`. Accessed:2022-03-29.

[51] N. Rasmussen, *Determining total cost of ownership for data center and network room infrastructure*, Relatório técnico, Schneider Electric, Paris, 8 (2011).

[52] J. Saraiva, M. Guimarales, and O. Belo, *An economic energy approach for queries on data centers*, (2017).

[53] M. Shah, A. Kothari, and S. Patel, *Influence of schema design in nosql document stores*, in Mobile Computing and Sustainable Informatics, Springer, 2022, pp. 435–452.

[54] B. Subramaniam and W. Feng, *On the energy proportionality of scale-out workloads*, arXiv preprint arXiv:1501.02729, (2015).

[55] B. Subramaniam and W.-c. Feng, *On the energy proportionality of distributed nosql data stores*, in International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems, Springer, 2014, pp. 264–274.

[56] A. Thakkar, K. Chaudhari, and M. Shah, *A comprehensive survey on energy-efficient power management techniques*, Procedia Computer Science, 167 (2020), pp. 1189–1199.

[57] Y. Tokusashi, H. Matsutani, and N. Zilberman, *Lake: An energy efficient, low latency, accelerated key-value store*, arXiv preprint arXiv:1805.11344, (2018).

[58] Y.-C. Tu, X. Wang, B. Zeng, and Z. Xu, *A system for energy-efficient data management*, ACM SIGMOD Record, 43 (2014), pp. 21–26.

[59] J. Wang, L. Feng, W. Xue, and Z. Song, *A survey on energy-efficient data management*, ACM SIGMOD Record, 40 (2011), pp. 17–23.

[60] Y. Xu, Zichen and Tu and X. Wang, *Dynamic energy estimation of query plans in database systems*, in 2013 IEEE 33rd International Conference on Distributed Computing Systems, IEEE, 2013, pp. 83–92.

[61] Z. Xu, Y. Tu, and X. Wang, *Pet: reducing database energy cost via query optimization*, Proceedings of the VLDB Endowment, 5 (2012), pp. 1954–1957.

[62] Z. Xu, Y.-C. Tu, and X. Wang, *Exploring power-performance tradeoffs in database systems*, in 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), IEEE, 2010, pp. 485–496.

[63] Z. Xu, X. Wang, and Y. Tu, *Power-aware throughput control for database management systems*, in 10th International Conference on Autonomic Computing ({ICAC} 13), 2013, pp. 315–324.

[64] Z. . Xu, Y. Tu, and X. Wang, *Online energy estimation of relational operations in database systems*, IEEE transactions on computers, 64 (2015), pp. 3223–3236.

Edited by: Fabrizio Marozzo
Received: Jan 21, 2021
Accepted: Apr 3, 2022