MEASURING THE ENERGY EFFICIENCY OF RDF QUERY PROCESSING

Marko Niinimaki[1], Kitichai Chanyalikit[2] and Atamfon Udofia[3]

[1]Lecturer, Chulalongkorn School of Integrated Innovation, Chulalongkorn University, 254 Phayathai Road, Pathumwan, Bangkok 10330, marko.n@chula.ac.th

[2]Assistant, Chulalongkorn School of Integrated Innovation, Chulalongkorn University, 254 Phayathai Road, Pathumwan, Bangkok 10330, kitichai.c@chula.ac.th

[3]Student, Chulalongkorn School of Integrated Innovation, Chulalongkorn University, 254 Phayathai Road, Pathumwan, Bangkok 10330, 6558104256@student.chula.ac.th

ABSTRACT

The cost of electric power consumed by a server computer is a significant component of its total cost of ownership. Since database servers are essential in the era of Big Data, we studied the performance and energy consumption of a small server. To achieve this, we stored a large set of RDF (Resource Description Framework) data in a database (RDF4J) running on consumer-grade hardware. Using realistic SPARQL language queries from Wiki data and a low-cost power/energy meter, we measured the energy consumption of RDF query processing. Our database management system responded to queries over a network connection, demonstrating that the network processing overhead in query processing was quite low (about 2 to 4%). We found that the most energy-efficient processing (queries per Watt) could be achieved with a slightly larger degree of parallelism than the best throughput (queries per hour). Moreover, we discovered that using a stripped-down version of the operating system on which the database ran did not affect the energy consumption of the query processing.

KEYWORDS: database, energy, RDF, RDF4J

1. Introduction

According to an estimate by the International Energy Agency (IEA) [1], the ITC ecosystem in 2019 consumed about 8.9% of world electricity supplied for all purposes. Power and cooling were between 25% to 33% of the total cost of ownership (TCO) of an enterprise data center [2, 3]. Analysts estimate that in 2024, 147 Zettabytes of data will be generated, with the amount increasing each year [4]. New methods will be needed to manage the data, store the energy needed and process the data. Indeed, energy-efficiency is one of the key drivers in the UN Sustainable Development Goals 7 (Affordable and Clean Energy) and 13 (Climate Action).

Big Data has been loosely defined as “the information asset characterized by high Volume, Velocity and Variety to require specific technology and analytical methods for its transformation into value” [5]. The value can naturally be financial potential or other benefits if the data can be analyzed. The challenge in managing big data lies in the ability to efficiently locate, access, store, and integrate vast volumes of rapidly evolving data that exist in diverse formats.

Here, we would like to promote an assumption by Martinez-Prieto et al: In the presence of heterogeneous data, the more data integrated and managed under a common model, the more interesting the knowledge generated may be, increasing the resulting dataset value [6]. As a common model, the Resource Description Framework (RDF, see [7]) appears to be a usable tool for Big Data integration. As discussed by Cruz and Xiao [8], data sources can be heterogeneous in syntax, schema, or semantics. Syntactic heterogeneity is the result of using different presentation formats for different data sources. Schematic heterogeneity results from structural differences in the data sources and can mean naming concepts differently in them. Ontologies, expressed in RDF and RDFS (RDF Schema) can alleviate these problems. We can define an RDF statement as a pair of resources (nodes) connected by a property (edge) [8]: a city (node) Rome is connected to its population (node) by a property (edge) “has population”. Due to this format, RDF statements are often called “triples”. RDFS is used to define the semantic relationships between properties and resources by using namespaces (or prefixes). For example, the prefix http://www.wikidata.org/entity/ defines the semantics of entities that can appear in statements about them in the context of Wiki data (see below).

In this paper, we present a method for measuring energy efficiency of RDF queries applied to non-trivial amounts of data. Our dataset is based on Wiki data – the data resource behind Wikipedia’s articles. Wiki data contains data items described through property-value pairs; for example, the item for "Rome" might have a property "population" with the value "2,777,979" [9]. These statements can further be refined by “qualifiers” like the statement above qualified as being an estimate by The Italian National Institute of Statistics in 2010.

Furthermore, items and properties each possess their own dedicated Wikipedia pages (such as "Rome" and "Population") and are also associated with unique resource identifiers (URIs) [9]. For example, the URI of Rome, Italy is http://www.wikidata.org/entity/Q220 and the URI for population is http://www.wikidata.org/entity/Q1613416. Our RDF export of the data is an RDF file of Wiki data, available at archive.org (https://archive.org/details/ wikidata-json-20150706). This file contains more than 1 billion statements. The queries used in testing are a set of 30 Wiki data sample queries from https://query.wikidata.org. The query language used is SPARQL [10].

Though the terms energy and power are often used synonymously, power is measured by Watts and energy by "Watts over time" like kilowatt hour (kWh).

As discussed by O'Neil [11], there are several types of database performance: throughput, response time, and the appropriate cost per user. By convention, we use "queries per hour" (QPH) as a measure for throughput and "average query processing time" as a measure for response time. "Watts per query" (WPQ) or "energy efficiency" is used in this paper for cost evaluation.

Our research objectives are:

- To evaluate the database performance of an RDF database management system (DBMS) operating on a commodity hardware PC.
- To identify the energy-consuming software components in a database system and determine their proportional power consumption.
- To analyze the differences in energy consumption when queries are executed in a DBMS running under different operating systems.

The rest of the paper is organized as follows. In Section 2, we give a review of related research. In Section 3, we describe the data, queries, and environment used for testing. In section 4, we detail the actual tests and their results. Section 5 contains a summary of the research and its benefits, a conclusion and directions for future research.

2. Related Work

In related research, Pickavet et al [12] gave a summary of power consumption with different kinds of information and communications technology (ICT) devices. Their survey includes energy consumption figures of TV sets, and video players but for the purpose of this research, the figures listed in Table 1 are the most relevant. Table 2 contains the International Energy Agency's estimate of Worldwide ICT Electricity Consumption by equipment type [1].

Table 1 ICT equipment and their power consumption

Type of equipment	Power consumption when active
Desktop PC with an LCD display	100 W
Laptop PC	30 W
Volume server	220 W
Mid-range server	700 W
High-end server	10 000 W

Table 2 Worldwide electricity consumption of ICT equipment, IEA estimate 2019

Type of equipment	Electricity consumption
Data centers	200 TWH
End users	550 TWH
Network equipment	250 TWH
ICT manufacturing	1000 TWH

Following Bianchini and Rajamony's [3] example, a 200W server running for one year would consume 1750 kWh. Based on the current U.S. electricity prices (17 cents/kWh) in 2024, the cost of running a 200W server for one year would be approximately $300. However, Bianchini and Rajamony calculated that cooling the server for a year would consume an additional 498 kWh.

The energy efficiency of PC hardware has improved since Pickavet's study. Park et al [13] stated that an ENERGY STAR registered LCD monitor consumes 10 to 25 Watts in on-mode. The ENERGY STAR Program [14] specifies requirements for typical annual energy consumption (TEC) in kWh. Some energy-efficient PCs listed at the program's web site have TECs as low as 5.4. During our measurements, we recorded power consumptions between

23.7 to 93.3 W. A PC hobbyist web page [15] gave the following measures for the components of an office PC like the one we have used in our tests:

CPU Intel i5: 73 to 95 W,

Motherboard: 25 to 40 W,

RAM DDR3: 2 to 3 W,

Hard drive 3.3”: 6.5 to 9 W,

80 mm case fan: 0.6 to 1.8 W.

Barroso and Holzle [16] have promoted energy proportional computing: computers should consume energy proportionally to the tasks they are performing. However (as of 2007), even an energy-efficient server still consumed about half its full power when doing virtually no work. Lang et al [17] studied if processing (optimizing and executing) queries as fast as possible is the most energy-efficient way to operate a database management system (DBMS). They found that this is not the case due to low server utilization and the hardware’s power/performance configurations.

Economou et al [18] presented a study of energy consumption of a server computer (“blade”) and its hardware and software components. Further, they developed the Mantis model to estimate a server’s energy consumption.

Arenas et al [19] discussed the logical model of RDF databases and the SPARQL query language but did not mention any actual RDF DBMS software implementations. Wylot et al [20], on the other hand, described multiple methods of implementing RDF storage and queries and listed DBMS’s that utilize them. Bizer and Schultz [21] introduced an RDF benchmark and tested four different RDF storages. The Java-based Sesame [22] is a popular RDF query processor and was discussed in their study. Sesame is the basis of the RDF4J framework (including a DBMS) and the commercial GraphDB product. Nacional et al [23] compared the performance of several RDF storage solutions (including RDF4J) with medical publication data and further compared their performance with a MongoDB storage of the same data. However, energy efficiency was not discussed in their study. Hernandez et al [24], likewise, studied RDF retrieval using five different RDF DBMS’s (including GraphDB) but did not study their energy efficiency. Notably, they used Wikidata in their study.

Shakhovska et al [25] provided a framework for Big Data processing based on a formal model of Big Data. Big Data technologies like Hadoop [26] combine distributed processing and storage methods.

Queries with RDF data in Hadoop environments have been discussed by Schätzle et al [27], Kawises and Vatanawood [28] and Husain et al [29]. Husain et al tested a limited set of queries using a 10-node Hadoop cluster and RDF data with up to 1.1 billion triples. Niinimaki et al [30] tested RDF query processing with many different node configurations in a Hadoop cluster and improved the processing speed (compared to a single node solution) by 85%.

Concerning database query energy efficiency, the Transaction Processing Council (TPC) releases benchmarks and tools aimed at measuring the “Watts per operation” characteristics of database products [31]. Tsirogiannis et al [32], using the TPC benchmark data, studied both hardware components and relational DBMS software methods in their database energy efficiency tests. Niinimaki et al [33] measured the energy consumption of a web-based database application and presented methods to improve its efficiency. Some studies have addressed the energy efficiency perspective of using RDF data on the Internet of Things context [34]. However, to our knowledge, there have not been studies measuring the energy efficiency of RDF databases and comparing our results with others is difficult.

3. The Test Data, Queries and Environment

For our tests, we have used a publicly available large data set from the Wiki data [9] project. This data set contains the statements of Wiki data, archived in June 2015 (https://archive.org/details/wikidata-json-20150706) in the Terse RDF Triple Language [35] format. The data set had some mistakes like encoding of the percent sign in the URLs, but after correcting those, the set contained 1 051 431 958 statements. The data size of the storage for these statements was about 96 gigabytes. An example of the data is shown below; the first item is about a volleyball player. Property P21 states her gender (Q6581072 is female), P31 – Q5 states she is human and P569 states her date of birth.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
..
@prefix wikibase: <http://wikiba.se/ontology-beta#> .
..
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix schema: <http://schema.org/> .
@prefix wd: <http://www.wikidata.org/entity/> .
..
```

```
@prefix wdt: <http://www.wikidata.org/prop/direct/> .
..
wd:Q702872 a wikibase:Item ;
..
  rdfs:label "Zhao Ruirui"@en ;
  skos:prefLabel "Zhao Ruirui"@en ;
  schema:name "Zhao Ruirui"@en ;
..
  schema:description "pallavolista cinese"@it,
        "Chinese volleyball players"@en,
..
  wdt:P21 wd:Q6581072 ;
  wdt:P31 wd:Q5 ;
..
  wdt:P569 "1981-10-08T00:00:00Z" ^^xsd:dateTime ;
```

Our thirty SPARQL queries are adopted from the sample queries of Wiki data. The queries include “Cats in Wiki data”, “Number of humans in Wiki data”, “Chemistry Nobel Prizes”, and “Persons whose gender is not known” (shown below).

```
SELECT ?human WHERE { ?human wdt:P21 ?gender FILTER isBLANK(?gender) . }
```

At the time of the writing, there were 326 sample queries in Wiki data. We have omitted queries that did not produce an answer with the 2015 data that we used. For queries that produced thousands of results, we have limited the number of results to 50. Some queries were omitted or modified because they returned multimedia content that our primitive test environments could not display. Three queries ran correctly but their running time was so long (more than 1 hour in our environment) that they would have dominated the test. The hardware and software environments for our test setup are shown below.

CPU: Intel I5-4590 @ 3.30 GHz

RAM: 8GB DIMM DDR3 Synchronous 1600 MHz

Motherboard (with an integrated video adapter): Dell 02YYK5

Hard drive: ST 500D M002-1DD142, 7200 RPM, average seek time 8.5 ms.

Operating system: Ubuntu Linux 18.04 kernel 4.18.0-15

Java: OpenJDK 11.0.7 2020-04-14

RDF4J: 3.1.3

Web Server: Tomcat 8.5.39

The energy consumption was measured using a low-cost consumer grade power/energy meter TS-863A.

To compare our test setup with a more professional server, we have used a higher-end Linux system kindly provided by the Ludwig Maximillian University, Munich. This computer was previously used in our database performance testing [23]. The software is similar to our test setup and the hardware characteristics are shown below.

CPU: 24-core Xeon (E5-2620 v2 @ 2.10 GHz),

RAM: 32 GB DDR3.

We measured the query performance of this system and compared it with our test system but unfortunately, we were not able to measure the energy consumption of the server due to data center policies.

4. Test Results

We executed the queries using two methods: (i) a completely linear execution where the RDF4J server is running in a “console” mode as a single process and (ii) a parallel method where RDF4J is running as a server.

The linear method gave us an overall view of the queries, while the server method corresponded to a more realistic database server environment.

The running times of the queries, using the linear method, are shown in the first column of Table 3. The test was repeated only three times because there was very little variance (0.1% of the combined time) in the execution times. The average combined execution time of the queries was 6530 seconds (about 1 hour 49 minutes) and thus, the average execution time for a query was 218 seconds. The longest running query (“largest cities”) took ca. 22 minutes and the shortest ones ca. 1/10 of a second.

For the parallel method, we used the RDF4J web applications (server and workbench) running under the Tomcat 8 Java servlet HTTP server. Using this technology, we created the repository as shown in Figure 1 and executed the “URL encoded” queries as shown below (the query is “People whose gender is not known”). The queries were executed on a separate computer connected to the server by a 1 Gbit Ethernet cable. The data transferred over the network by the requests and responses was only 352 kB.

curl http://192.168.10.120:8080/rdf4j-server/repositories/webwiki?query=%20%20PREFIX%20wdt%3A%20%3Chttp%3A%2F%2Fwww.wikidata.org%2Fprop%2Fdirect%2F%3E%20%20PREFIX%20wd%3A%20%3Chttp%3A%2F%2Fwww.wikidata.org%2Fentity%2F%3E%20SELECT%20%3Fhuman%20WHERE%20%7B%20%3Fhuman%20wdt%3AP21%20%3Fgender%20FILTER%20isBLANK%28%3Fgender%29%20.%20%7D

Even though using the web server allowed us to execute our queries in parallel, Table 3 column 3 contains the execution times of queries when they are run as a sequence of HTTP requests. Comparing columns 2 and 3, we can observe that the overhead of running the queries through the HTTP server is small. The difference in the average combined execution times was 282 seconds (about 4%). Both the console and web-based systems executed about 16 queries per hour. For our higher-end system this figure was about 700 queries per second.

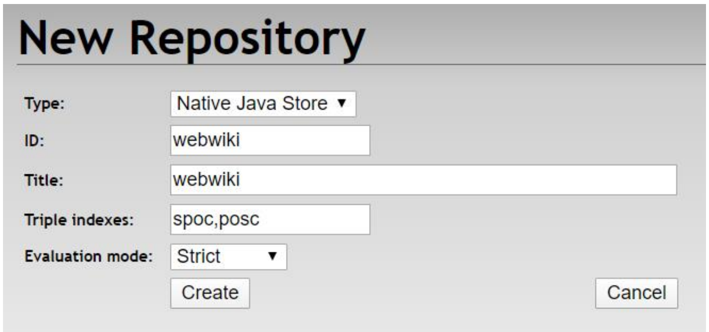

Figure 1 Creating a web-based RDF database

Our first parallel benchmarking task was to find the best performance level, which is defined as the number of parallel processes that gives the most responses per hour. For this purpose, we created N query sets, i.e. files containing the URL encoded queries in a random sequence. These files were then executed in parallel. The results with different values of N are shown in Table 4.

We can see that parallel processing with two query sets is actually less efficient than running the queries individually. However, the queries per hour figures improved as we increased the number of parallel query sets. This continued until the resources were exhausted. We first noticed this when we ran 6 query sets in parallel. When running multiple query sets in parallel, the server was quite overloaded. The average load figures were between 7.1 and 8.9 with 7 query sets. When running the linear test, however, these figures were between 0.4 and 1.2.

Having determined that the best throughput (queries per hour) was achieved when five query sets were executed in parallel, our next task was to study the energy consumption of the server hardware and software while running the queries.

The desktop PC used in our tests is a typical “small form factor” office PC with an integrated VGA and Ethernet adaptors, and a 500 GB traditional SATA hard disk (not a solid disk drive). This PC is not listed at energystar.gov, however a comparable model by the same manufacturer has TEC ratings between 77.8 and 100.8 kWh. Using our power/energy meter, the power consumption was 23.7W to 23.8W, when idle. The maximum reading of the power meter was 93.3 W. Furthermore, we measured the energy consumption of the system when it ran for one hour with and without the Tomcat web server and these tests gave a figure of 24 W each. We could not determine a definite overhead for the Tomcat process when it was not processing any queries. This is understandable since the Tomcat process consumed only about 0.3% of the CPU power and 3.3% of the memory. We shall later estimate the energy consumption of Tomcat using the proportion of CPU usage (in Figure 2).

The results of energy measurements are shown in Table 5. The linear test was executed using the “console” command of RDF4J. The parallel tests were executed by connecting a client computer directly to the server, starting the Tomcat server in the server computer and running the queries using the curl program in the client.

Table 5 indicates that the lowest Watts per query figure does not correspond with the best performance – it was reached after running six or seven query sets in parallel.

Earlier, we noted that the performance difference was approximately 4% when running a single sequence of queries using RDF4J alone compared to RDF4J in conjunction with Tomcat. Next, we estimated Tomcat’s energy consumption overhead when running queries in parallel. Figure 2 (made with the Python psrecord package) illustrates the CPU usage of the service containing both Tomcat and RDF4J. The CPU levels were recorded during the first hour of parallel testing with six query sets. The average CPU consumption was 17.9%. The role of Tomcat during the process was to pass queries to RDF4J, receive the replies and pass them to the client. We simulated the process by measuring the CPU consumption of Tomcat when retrieving 25 small static files under its control (25 since this is the QPH figure for the server). During this experiment, the Tomcat process's CPU consumption was 0.3%. Therefore, it is likely that the Tomcat component's energy consumption was about 0.3/17.9 which is 2% of the Tomcat+RDF4J combination.

Our final research question aims at finding out whether we can save energy by using a different kind of operating system. We tested the Porteus Linux distribution [36] and the preliminary findings were mixed. When the Porteus system was idle, it consumed about 20% less energy than Ubuntu Linux. However, when running a database and replying to queries, there was no difference between the energy consumption of the two operating systems.

Table 3 Queries and Their Linear Execution Times in Milliseconds

Query#	Command line	Web server
1	703	1377
2	44	74
3	455500	517986
4	520919	508632
5	4250	4637
6	12350	13403
7	704651	793857
8	1281709	1333476
9	3008	2829
10	576682	598554
11	131	162
12	239821	249979
13	42411	44843
14	13019	15444
15	1119	1348
16	987	1160
17	166	181
18	121215	135043
19	13840	15312

Table 3 Queries and Their Linear Execution Times in Milliseconds (continued)

Query#	Command line	Web server
20	45527	57866
21	5583	5625
22	14523	15079
23	783339	786481
24	662078	673351
25	670623	678277
26	4818	4690
27	313785	314466
28	37124	37362
29	89	103
30	43501	42185
SUM	6573515	6853785
H:MIN:S	1:49:33	1:54:13
AVG	219117	228459
QPH	16.4	15.75

Table 4 Parallel Processing

Num parallel tasks	Average query processing time (ms)	Average query set processing time	QPH
2	494663	4h9min	14.48
3	542872	4h31min	19.89
4	620471	5h10min	23.21
5	709122	5h54min	25.35
6	945741	7h52min	22.84
7	1115723	8h53min	23.60

Table 5 Energy Consumption During Testing

Type of test	Num queries	Time (h)	Energy consumption (kWh)	Watts	WPQ
Linear	30	1:50	0.05	27.38	0.91
Parallel-5	150	5:54	0.174	29.44	0.20
Parallel-6	180	7:52	0.218	27.66	0.15
Parallel-7	210	8:53	0.247	26.5	0.15

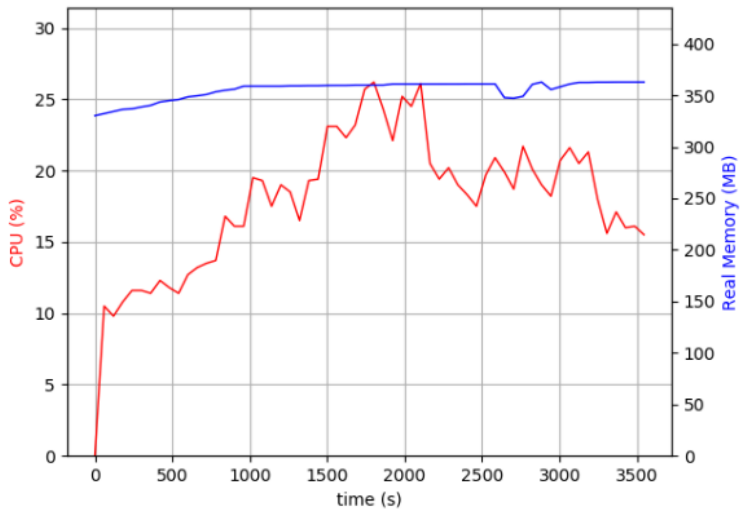

Figure 2 CPU and Memory Consumption of Tomcat + RDF4J

5. Summary and Future Research

In this paper, we answered the following questions:

- What kind of database performance can we expect from an RDF database system running in a commodity hardware PC?
- How can we identify the energy consuming software components in a database system, and their proportional power consumption?
- Are there differences in energy consumption when the queries are executed in a DBMS running under different operating systems?

The main benefits of our research are:

- Demonstrating that a commodity PC with a Linux operating system is feasible for processing RDF queries. However, it is observed that the execution of complex queries results in significant processing delays.

- Demonstrating that the overhead of network data processing is small compared with query processing.
- Demonstrating that a “lightweight” Linux distribution does not improve the energy efficiency of query processing.

For our measurements, we have used a large set (more than 1 billion statements) from Wiki data, and a set of 30 Wikidata example SPARQL queries. Our hardware environment (the database server) was a simple desktop PC with integrated graphics, 500 GB hard drive, 8 GB memory and an Intel i5 CPU. The tests were run using a client PC computer connected to the server directly by an Ethernet cable. We measured the energy consumption of the tests with a consumer grade power/energy meter connected to a wall socket.

The results, based on our measurements, can be stated as follows:

The commodity hardware PC’s query processing performance was less than 3% of that of our higher end system (24 core Xeon with 32 GB RAM).

We can measure the impact of a database management system (RDF4J) in the energy consumption when processing queries with a system that contains RDF4J under the control of a web server (Apache Tomcat). However, estimating the impact of the web server is more difficult since its overhead is small compared with RDF4J, and we estimate it to be about 2%.

Changing the operating system to a more minimalistic Linux distribution reduces the energy consumption when the system is running idle but does not have a real impact when the system is processing queries.

We have run some initial measurements on our desktop PC using a solid-state disk (SSD) and the performance improvement was notable. This is in line with Intel’s reporting quadruple database performance when using SSDs [37]. In our future research, we shall provide figures for our setup with an SSD. Other future research will concentrate on “fine tuning” the operating system and the Java environment for better performance and economy. More challenging topics will expand the scope of the study with other types of databases and retrieval systems. RDF processing can be compared with a more recent alternative: graph databases with the GraphQL query language. RDF databases have been studied in the context of Large Language Models, too [38]. Expanding this type of research could combine the ease of use of Large Language Models with the fact-based knowledge bases implemented using RDF.

References

[1] Gelenbe E. Electricity consumption by ICT: Facts, trends, and measurements. Ubiquity 2023;(August):1-15.

[2] Patel CD, Shah AJ. Cost model for planning, development and operation of a data center. Hewlett-Packard Laboratories Technical Report 2005;107:1-36.

[3] Bianchini R, Rajamony R. Power and energy management for server systems. Computer 2004;37(1):68-76.

[4] Alex Woodie. Big Growth Forecasted for Big Data [Internet]. Datanami [updated 2022 Jan 11; cited 2024 Aug 12] Available from: https://www.datanami.com/2022/01/11/big-growth-forecasted-for-big-data

[5] De Mauro A, Greco A, Grimaldi M. A formal definition of Big Data based on its essential features. Library review 2016;65(3):122-35.

[6] Martínez-Prieto MA, Cuesta CE, Arias M, Fernández JD. The solid architecture for real-time management of big semantic data. Future Generation Computer Systems 2015;47:62-79.

[7] The World Wide Web Consortium. W3C RDF [Internet] The World Wide Web Consortium [updated 2014 Feb 25; cited 2024 Aug 12] Available from: https://www.w3.org/RDF/.

[8] Cruz IF, Xiao H. The role of ontologies in data integration. Engineering intelligent systems for electrical engineering and communications 2005;13;245.

[9] VrandeČiĆ D, Krötzsch M. Wikidata: a free collaborative knowledgebase. Communications of the ACM 2014;57(10):78-85.

[10] Harris S, Seaborne A, Prud’hommeaux E. SPARQL 1.1 query language. [Internet] The World Wide Web Consortium [updated 2013 Mar 21; cited 2024 Aug 12] Available from: https://www.w3.org/TR/sparql11-query.

[11] O’Neil PE. Database Performance Measurement. In: Tucker AB, editor. The Computer Science and Engineering Handbook. Boca Raton, USA: CRC Press, 1997. p.1078-92.

[12] Pickavet M et al. Worldwide energy needs for ICT: the rise of power-aware networking. Proceeding of 2nd International Symposium on Advanced Networks and Telecommunication Systems; 2008 Dec 15-17; Mumbai, India. Piscataway, USA: IEEE; 2008. p. 1-3.

[13] Park WY, Phadke A, Shah N. Efficiency improvement opportunities for personal computer monitors: implications for market transformation programs. Energy Efficiency 2013;6:54-69.

[14] ENERGYSTAR. ENERGY STAR program requirements [Internet] ENERGYSTAR [updated 2010 Aug 1; cited 2024 Aug 12] Available from: https://www.energystar.gov/sites/default/files/specs/private/Computers_Program_Requirements.pdf

[15] BuildComputers. Power consumption of PC components in watts. buildcomputers.net [Internet]. BuildComputers [updated 2013 Mar 15; cited 2024 Aug 12] Available from: http://www.buildcomputers.net/power-consumption-of-pc-components.html

[16] Barroso LA, Hölzle U. The case for energy-proportional computing. Computer 2007; 40(12):33-7.

[17] Lang W, Kandhan R, Patel JM. Rethinking query processing for energy efficiency: Slowing down to win the race. IEEE Data Engineering Bulletin 2011:34:12-23.

[18] Economou D, Rivoire S, Kozyrakis C, Ranganathan P. Full-system power analysis and modeling for server environments. In: Eeckhout, Lieven, and J. YI, editors. Proceedings of the Second Annual Workshop on Modeling, Benchmarking and Simulation (MoBS) Held in Conjunction with the 33rd Annual International Symposium on Computer Architecture (ISCA-33); 2006 Jun 17-21; Boston, MA. Piscataway, USA: IEEE; 2006. p. 70-7.

[19] Arenas M, Gutierrez C, Pérez J. Foundations of RDF databases. Heidelberg, Germany: Springer; 2009.

[20] Wylot M, Hauswirth M, Cudré-Mauroux P, Sakr S. RDF data storage and query processing schemes: A survey. ACM Computing Surveys (CSUR) 2018;51(4):84.

[21] Bizer C, Schultz A. Benchmarking the performance of storage systems that expose SPARQL endpoints. Proceeding of 4 th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS); 2008 Oct 21; Karlsruhe, Germany. CEUR; 2008. p. 39.

[22] Broekstra J, Kampman A, Van Harmelen F. Sesame: A generic architecture for storing and querying RDF and RDF schema. Proceeding of the Semantic Web-ISWC 2002: First International Semantic Web Conference; 2002 June 9-12; Sardinia, Italy. Heidelberg, Germany: Springer; 2002. p. 54-68.

[23] Nacional T, Niinimaki M, Heikkurinen M. RDF Databases - Case Study and Performance Evaluation. MATTER: International Journal of Science and Technology 2019;5(3):1-14.

[24] Hernández D, Hogan A, Krötzsch M. Reifying RDF: What works well with Wikidata? Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems; 2015 Oct 11; 2015; Bethlehem, USA. p. 32-47.

[25] Shakhovska N, Veres O, Bolubash Y, Bychkovska L. Big data information technology and data space architecture. Sensors & Transducers 2015;195:69-76.

[26] Kambatla K, Pathak A, Pucha H. Towards optimizing hadoop provisioning in the cloud. Proceedings of the 2009 conference on Hot topics in cloud computing (HotCloud'09); 2009 Jun 15; San Diego, USA. USA: USENIX Association; 2009. Article 22.

[27] Schätzle A, Przyjaciel-Zablocki M, Neu A, and Lausen G. Sempala: Interactive SPARQL query processing on Hadoop. The Semantic Web–ISWC 2014: 13th International Semantic Web Conference; 2014 Oct 19-23; Riva del Garda, Italy. Heidelberg, Germany: Springer; 2014. p. 164-179.

[28] Kawises J and Vatanawood W. A development of RDF data transfer and query on Hadoop Framework. 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS); 2016 June 26-29, Okayama, Japan. Piscataway, USA: IEEE; 2016. p. 1-4.

[29] Husain MF, Doshi P, Khan L, Thuraisingham BM. Storage and retrieval of large RDF graph using Hadoop and MapReduce. In: Jaatun MG, Zhao G, Rong C, editors, Proceeding of the first International Conference (CloudCom 2009); Beijing, China, 2009 December 1-4. Lecture Notes in Computer Science 2009 (5931):680-86.

[30] Niinimaki M, Niemi T, Thanisch P. Dataspace management with ETL and RDF support. Naresuan University Journal: Science and Technology (NUJST) 2020;28:36-49.

[31] Poess M, Nambiar RO, Vaid K, Stephens Jr JM, Huppler K, Haines E. Energy benchmarks: a detailed analysis. Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking; 2010 Apr 13-15; Passau, Germany. New York, USA: ACM; 2010. p. 131-40.

[32] Tsirogiannis D, Harizopoulos S, Shah MA. Analyzing the energy efficiency of a database server. Proceeding of the 2010 ACM SIGMOD International Conference on Management of data; 2010 Jun 6-11; Indianapolis, USA. New York, USA: ACM; 2010. p. 231-42.

[33] Niinimäki M, Abaunza F, Niemi T, Thanisch P, Kommeri J. Energy-efficient query processing in a combined database and web service environment. Green Computing Strategies for Competitive Advantage and Business Sustainability. Pennsylvania, USA: IGI Global; 2018. p. 62-88

[34] Hasemann H, Kröller A, and Pagel M. RDF Provisioning for the Internet of Things. Proceeding of the 3rd IEEE International Conference on the Internet of Things; 2012 Oct 24-26; Wuxi, China. Piscataway, USA: IEEE; 2012 p. 143-50.

[35] Prud'hommeaux E, Carothers G. Turtle - Terse RDF triple language. [Internet] The World Wide Web Consortium [updated 2015 Feb 25; cited 2024 Aug 12] Available from: https://www.w3.org/TR/turtle

[36] Flood J. Porteus Linux. [Intenet] Porteus [updated 2023 Oct 4; cited 2024 Aug 12]. Available at: http://porteus.org

[37] Wu K, Arpaci-Dusseau A, Arpaci-Dusseau R, Sen R, Park K. Exploiting Intel Optane SSD for Microsoft SQL server. In: 15th International Workshop on Data Management on New Hardware; 2019 July 1; Amsterdam, Netherlands. New York, USA: ACM; 2019. Article 15.

[38] Frey J, Meyer LP, Arndt N, Brei F, Bulert K. Benchmarking the abilities of large language models for RDF knowledge graph creation and comprehension: How well do LLMs speak Turtle. Proceeding of the ISWC 2023 Workshop on Deep Learning for Knowledge Graphs; 2023 Nov 6-7; Athens, Greece. CEUR; 2023.

Author's Profile

Mr. Marko Niinimaki gained his Ph.D. in computer science at the University of Tampere, Finland. He has worked as a researcher at CERN, and lectured computer science in many universities in Europe and Asia. His special interests are databases and data management. He currently works at Chulalongkorn University School of Integrated Innovation.

Mr. Kitichai Chanyalikit gained his M.Sc. at the Assumption University of Bangkok. He shares his time between Chulalongkorn University and his web consultancy business.

Ms. Atamfon Udofia is currently a 3rd year student of the Chulalongkorn School of Integrated Innovation. She is an enthusiast of Artificial Intelligence and Machine Learning with 3 years of pre-medical studies at the University of Benin, Benin City, Nigeria.

Article History:

Received: August 22, 2023

Revised: August 13, 2024

Accepted: August 21, 2024