DOI: 10.11591/ijeecs.v7.i1.pp250-262

Efficiency of JSON for Data Retrieval in Big Data

Mohd Kamir Yusof*[1], Mustafa Man[2]
[1]Universiti Sultan Zainal Abidin, 22200 Besut, Terengganu, Malaysia
[2]School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia
*Corresponding author, e-mail: kamir2020@gmail.com

Abstract

Big data is the latest industry buzzword to describe large volume of structured and unstructured data that can be difficult to process and analyze. Most of organization looking for the best approach to manage and analyze the large volume of data especially in making a decision. XML is chosen by many organization because of powerful approach during retrieval and storage processes. However, XML approach, the execution time for retrieving large volume of data are still considerably inefficient due to several factors. In this contribution, two databases approaches namely Extensible Markup Language (XML) and Java Object Notation (JSON) were investigated to evaluate their suitability for handling thousands records of publication data. The results showed JSON is the best choice for query retrieving speed and CPU usage. These are essential to cope with the characteristics of publication's data. Whilst, XML and JSON technologies are relatively new to date in comparison to the relational database. Indeed, JSON technology demonstrates greater potential to become a key database technology for handling huge data due to increase of data annually.

Keywords: *big data, data retrieval, JSON, XML*

1. Introduction

Big data is referring to huge amount of data that are so large and complex that traditional data processing applications are inadequate to deal with them [1]. Big data is often distributed over many storage devices, can be in several locations. Big data is currently a catch phrase in industries such as information technology, business, health care, etc. [2]. In simplest term, big data refers to the tools, process and procedures that allow an organizations to create, manipulate, and manage very large data sets and storage families. The challenges of big data include analysis, capture, data duration, sharing, storage, transfer, visualisation, querying, updating, and information privacy. According to sociology and research article, a number of reports and academic publications have pointed to the growing use of big data across economic sectors and its potential to bolster productivity, efficiency and growth [3]. One of the issues in accessing big data or large dataset is efficiency. The efficiency of accessing large dataset can be measured by the time it fetch the data based on the query. Two current approaches have implemented to handle large dataset which are relational database and XML. Relational database is traditional approach for storing and managing big data. By using this approach, the data can be represented in a table form. Database Management System (DBMS) is used to control and manipulate the data [4]. However, by using this approach, time to fetch the data are considerably inefficiency. One of the solution to handle this problem is XML approach. XML is an emerging standard for exchanging representation over the Internet [5]. XML is widely used to store and manage huge of data. This approach is currently used by most of industries such as health care, education, business, etc. In this research, a JSON approach is proposed for storing and managing huge of data. JSON is chosen because of flexibility and can handle high throughput and low latency without sacrificing and scalability [6]. JSON also is directly supported inside JavaScript and the best suited for JavaScript application; thus provide significant performance compare to XML and relational database [7]. JSON is proven produce better performance compared to XML for web service applications [8].

In experimental, datasets DBLB is used as a benchmark dataset. The performance of JSON approach will compared with XML approach. The comparisons are made from the following aspects: query performance and CPU usage for data retrieving process. The rest of

Received March 10, 2017; Revised May 27, 2017; Accepted June 13, 2017

contribution is organized as follows: Section I gives the related works. Section III describes about the kinds of data model such as relational database, XML and JSON. Section IV discusses the two database approaches concerned based on experimental results and our experience in the development. Finally, a conclusion is given in Section V.

2. Related Works

Based on past researches, the most popular approach compared to relational database is XML. XML stands for eXtensible Markup Language a standard for data exchange issued by the World Wide Consortium (W3C) in 1998 [9]. XML approaches have been implemented for clinical data storage [10]. This technique is effective to manage the clinical data and transform the data into structured format. The advantages of XML approach for clinical data are better in term of scalability, flexibility and extensibility. Native XML approach also has been implemented in external and distributed database [11]. The purpose of native XML is to minimize the query retrieval speed [12]. XML approach successful to handle huge data around 100000 records. In chemical industry, XML also used for integration of chemical data [13]. The implementation of XML approach because of chemistry community has been slower to adopt the Internet as a central service for exchanging information. Chemical data involves with large number of data file. XML approach can improve the efficiency of query processing when involves with the large number of data file. XML also can be used as a web services platform which is provides functionality for data exchange [14]. XML is implemented to overcome the information sharing each other and large number of databases issues. Through XML approach, different systems can share and exchange the information easily. By implementation of XML approach in different domains, XML is proven to handle large number of data. The efficiency of query processing using XML is efficiency compared to relational database. However, the efficiency of query processing using XML still can improve by using another approach as an alternative database approach. The researchers still looking the best technique for handling huge data. In this paper, JSON is introduced as a new approach to handle and manage huge data. JSON is a lightweight data-interchange format that easy for humans to read and write, and for machines to parse and generate [15, 16]. JSON approach has been implemented and able to handle 1000 records to 20000 records [17]. The result shows JSON approach is powerful and more efficient in term of storage and query retrieval compared to XML. In this paper, comparison will made between XML and JSON approach to handle huge data which is more than 20000. This is important to shows the efficiency of JSON approach for handling huge data

3. Database Model

This section described about three different data model for publication data. They are relational database approach, XML approach and JSON approach. Figure 1 shows the diagram which is contains publication data. Based on Figure 1 publication data coming from different sources such as article, book, inproceeding, master thesis (sthesis), proceeding, website/URL (www) and PhD thesis (phdthesis).

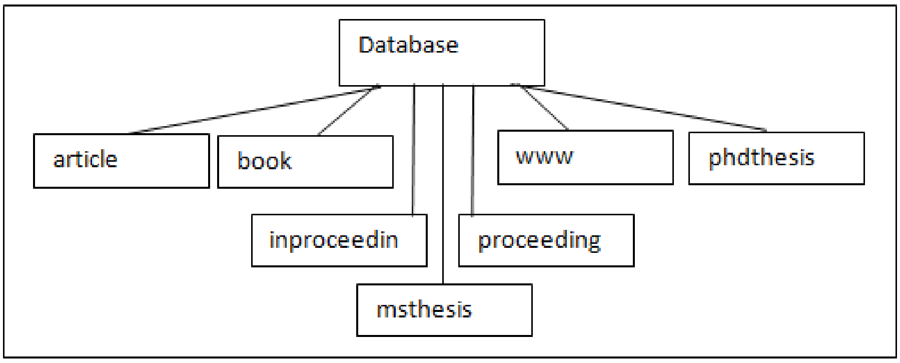


Figure 1. List of tables

3.1. Ralational Database Model

The definition about relational database is one whose symbol are organized into a collection of relations [2], [18]. Relational data is complex, it mimics the way people think by grouping similar objects together and breaking down complex objects into similar ones [3]. Table 1 until Table 7 shows how publication data is stored. Tables that contains the publication data is divided into two part; row and column. Column represent attributes name and rows represent number of data (something is called tuples).

Definition 1: A tuple has the form $\{A_1 = v_1, \ldots, A_n = v_n\}$, where $A_1, \ldots, A_n$ are attributes and $v_1, \ldots, v_n$ are their values.
Definition 2: The signature of the tuple, S, is the set of all its attributes $\{A_1, \ldots A_n\}$, if t is tuple of signature S, the projection t.Aj computes to the value v_i. If X is a set of attributes $\{B_1, \ldots, B_m\} \in S$ and t is a tuple with signature S, we write t.X for the sequence of values $(t.B_1, \ldots, t.B_m)$.
Definition 3: A relation R of signature S is a set of tuples with signature S.

Example 1:

Table 1 has 9 attributes: id, author, title, pages, year, volume, journal and url. The attributes are listed on the first line of the table, whereas the other lines each represent a tuple.

The first tuple is {id=274222, author=N.Prati, title=A Partial Moldel of NP with E., pages=1245-1253, year=1994, volume=59, journal=J.Symb. Log., url=db/journals/jsyml59.html#Prati94}.

If we call this tuple t, then t.id=274222. The signature is the set {id, author, title, pages, year, volume, journal, url}. The whole relation is represented in Table 1.

Table 1. Article

Id	Author	Title	Pages	Year	Volume	Journal	Url
274222	N. Prati	A Partial Model of NP with E.	1245-1253	1994	59	J. Symb. Log.	Db/journals/jsyml/jsyml59.html#Prati94
274224	J. Barkley Rosser	Godel Theorems for Non-Constructive Logics.	129-137	1937	2	J. Symb. Log.	Db/journals/jsyml/jsyml2.html#Rosser37
296027	Andreas Dandalis, Viktor K. Prasanna	Run-time performance optimization of an FPGA-based deduction engine for SAT solvers.	547-562	2002	7	ACM Trans. Design Autom. Electr. Syst.	Db/journals/todaes/todaes7.html#DandalisP02

Example 2:

Table 2 has 9 attributes: id, isbn, author, title, series, volume, publisher, year and url. The attributes are listed on the first line of the table, whereas the other lines each represent a tuple.

The first tuple is {id=214, isbn=3-540-55382-7, author= Andrew Cheese, title=Parallel Execution of Parlog, series= Lecture Notes in Computer Science, volume=586, publisher= Springer, year=1992, url=-}

If we call this tuple t, then t.id = 214. The signature is the set {id, isbn, author, title, series, volume, publisher, year and url}. The whole relation is represented in Table 2.

Table 2. Book

Id	Isbn	Author	Title	Series	Volume	Publisher	Year	Url
214	3-540-55382-7	Andrew Cheese	Parallel Execution of Parlog	Lecture Notes in Computer Science	586	Springer	1992	-
219	3-540-12282-6	Heinz Bender	Korrekte Zugriffe zu verteilten Daten	Informatik-Fachberichte	63	Springer	1983	-

Example 3:

Table 3 has 7 attributes: id, author, title, pages, year, booktitle and url. The attributes are listed on the first line of the table, whereas the other lines each represent a tuple.

The first tuple is {id=338405, author= Alf Smith, title= On Recursive Free Types in Z, pages=3-39, year=1991, booktitle= Z User Workshop, url= db/conf/zum/zum1991.html#Smith91}

If we call this tuple t, then t.id = 338405. The signature is the set {id, author, title, pages, year, booktitle and url}. The whole relation is represented in Table 3.

Table 3. Inproceeding

Id	Author	Title	Pages	Year	Booktitle	Url
338398	Rosalind Barden, Susan Stepney	Support for Using Z.	255-280	1992	Z User Workshop	db/conf/zum/zum1992.html# BardenS92
338405	Alf Smith	On Recursive Free Types in Z.	3-39	1991	Z User Workshop	db/conf/zum/zum1991.html# Smith91
338419	David Gries	Equational Logic: A Great Pedagogical Tool for Tea	508-509	1995	ZUM	508-509

Example 4:

Table 4 has 5 attributes: id, author, title, year and school. The attributes are listed on the first line of the table, whereas the other lines each represent a tuple.

The first tuple is {id=14, author=Peter Van Roy, title=A Prolog Compiler for the PLM., year=1984, school= University of California at Berkeley}

If we call this tuple t, then t.id=14. The signature is the set {id, author, title, year and school}. The whole relation is represented in Table 4.

Table 4. Msthesis

Id	Author	Title	Year	School
12	Tolga Yurek	Efficient View Maintenance at Data Warehouses.	1997	University of California at Santa Barbara, Departm
14	Peter Van Roy	A Prolog Compiler for the PLM.	1984	University of California at Berkeley
15	Tatu Ylnen	Shadow Paging Is Feasible.	1994	Helsinki University of Technology, Department of C

Example 5:

Table 5 has 6 attributes: id, editor, title, year, month and school. The attributes are listed on the first line of the table, whereas the other lines each represent a tuple.

The first tuple is {id=1, editor=Joann J. Ordille, title=Descriptive Name Services for Large Internets., year=1993, month=-, school= Univ. of Wisconsin-Madison}

If we call this tuple t, then t.id=1. The signature is the set {id, editor, title, year, month and school}. The whole relation is represented in Table 5.

Table 5. Phdthesis

id	editor	Title	year	month	school
1	Joann J. Ordille	Descriptive Name Services for Large Internets.	1993		Univ. of Wisconsin-Madison
2	Francisco Reverbell	Persistence in Distributed Object Systems: ORB/ODB...	1996	April	University of New Mexico
4	Dietmar Seipel	Decomposition in Database and Knowledge-Base Systems.	1989		Uni Wurzburg

Example 6:

Table 6 has 10 attributes: id, editor, title, booktitle, series, volume, publisher, year, isbn and url.

The attributes are listed on the first line of the table, whereas the other lines each represent a tuple

The first tuple is {id=1330, editor=Naveen Prakash, Colette Rolland, Barbara Pernici, title=Information System Development Process, Proceedings of the IFIP WG8.1 Working Conference on Information System Development Process, Como, Italy, 1-3 September, 1993, booktitle=Information System Development Process, series= IFIP Transactions, volume=A-30, publisher=North-Holland, year=1993, isbn=0-444-81594-5, url=db/conf/ifip8-1/ifip8-1-1993.html}

If we call this tuple t, then t.id=1330. The signature is the set {id, editor, title, year, month and school}. The whole relation is represented in Table 6.

Table 6. Proceeding

Id	Editor	Title	Booktitle	Series	Volume	Publisher	Year	Isbn	Url
1330	Naveen Prakash, Colette Rolland, Barbara Pernici	Information System Development Process, Proceedings of the IFIP WG8.1 Working Conference on Information System Development Process, Como, Italy, 1-3 September, 1993	Information System Development Process	IFIP Transactions	A-30	North-Holland	1993	0-444-81594-5	db/conf/ifip8-1/ifip8-1-1993.html
1341	Tom J. van Weert, Robert Munro	Informatics and The Digital Society: Social, Ethical and Cognitive Issues, IFIP TC3/WG3.1&3.2 Open Conference on Social, Ethical and Cognitive Issues on Informatics and ICT, July 22-26, 2002, Dortmund, Germany	SECIII	IFIP Conference Proceedings	244	Kluwer	2003	1-4020-7363-1	db/conf/ifip3-1/ifip3-1-2002.html

Example 7:

Table 7 has 6 attributes: id, editor, title, booktitle, year and url. The attributes are listed on the first line of the table, whereas the other lines each represent a tuple.

The first tuple is {id=2, editor=Mary F. Fernandez, Jonathan Robie, title=XML Query Data Model, booktitle=-, year=2001, url=http://www.w3.org/TR/query-datamodel}

If we call this tuple t, then t.id=2. The signature is the set {id, editor, title, booktitle, year and url}. The whole relation is represented in Table 7.

Table 7. www

id	editor	title	Booktitle	year	url
2	Mary F. Fernandez, Jonathan Robie	XML Query Data Model	-	2001	http://www.w3.org/TR/query-datamodel
3	Arnon Rosenthal	The Future of Classic Data Administration: Objects	SWEE	1998	http://www.mitre.org/support/swee/rosenthal.html

3.2. XML Data Model

XML provides a standard for the semantic management of data. It is a formal meta-language facility for defining a markup language. The basic unit in an XML file is entity or chunk that contains content and markup. The markup describes a content. More generally, markup consists of tags, attributes, comments, and processing instructions for the content. In a start tag, the name and any additional information are surrounded by the “<” and “>” characters. Similarly, an end tag consists of the tag name surrounded by the “< /” and “>”. XML is case sensitive so start and end tag names must match exactly. Figure 2 shows how the publication data is represented in XML format.

```
<record>
        <article>
        <id>274222</id>
        <author>N. Prati</author>
        <title>A Partial Model of NP with E.</title>
        <pages>1245-1253</pages>
        <year>1994</year>
        <volume>59</volume>
        <journal>J. Symb. Log.</journal>
        <url>db/journals/jsyml/jsyml59.html#Prati94</url>
        </article>
        :
        :
        <book>
        <id>211</id>
        <isbn>3-540-60058-2</isbn>
        <author>Marco Cadoli</author>
        <title>Tractable Reasoning in Artificial Intelligence</title>
        <series>Lecture Notes in Computer Science</series>
        <volume>941</volume>
        <publisher>Springer</publisher>
        <year>1995</year>
        <url>...</url>
        </book>
        :
        :
        <inproceeding>
        <id>338396</id>
        <author>Regine Laleau, Amel Mammar</author>
        <title>A Generic Process to Refine a B Specification into</title>
        <pages>22-41</pages>
        <year>2000</year>
        <booktitle>ZB</booktitle>
        <url>db/conf/zum/zb2000.html#LaleauM00</url>
        </inproceeding>
        :
        :
        <sthesis>
        <id>11</id>
        <author>Kurt P. Brown</author>
        <title>PRPL: A Database Workload Specification Language, v1.3.</title>
        <year>1992</year>
        <school>Univ. of Wisconsin-Madison</school>
        </sthesis>
        :
        :
        <phdthesis>
        <id>1</id>
        <editor>Joann J. Ordille</editor>
        <title>Descriptive Name Services for Large Internets.</title>
        <year>1993</year>
        <month>...</month>
        <school>Univ. of Wisconsin-Madison</school>
        </phdthesis>
        :
        :
        <proceeding>
        <id>1325</id>
        <editor>Elen Balka, Richard Smith</editor>
        <title>Woman, Work and Computerization: Charting a Course to the Future, IFIP TC9/WG9.1
```

```
            Seventh International Conference on Woman, Work and Computerization, June 8-11, 2000,
            Vancouver, British Columbia, Canada</title>
            <booktitle>Woman, Work and Computerization</booktitle>
            <series>IFIP Conference Proceedings</series>
            <volume>172</volume>
            <publisher>Kluwer</publisher>
            <year>2000</year>
            <isbn>0-7923-7864-4</isbn>
            <url>db/conf/ifip9-1/ifip9-1-2000.html</url>
            </proceeding>
            :
            :
            <www>
            <id>1</id>
            <editor>...</editor>
            <title>Java Language Home Page</title>
            <booktitle>...</booktitle>
            <year>...</year>
            <url>http://java.sun.com/</url>
            </www>
</record>
```

Figure 2. Tree representation of publication XML

3.3. JSON Data Model

In this approach, data is represented in array format. JSON is built on two structures. The first is a collection of name/value of pairs. In various language, this is realized as an object, record, structure, dictionary, hash table, keyed list, or associate array. The second is an ordered list of values. In most language, this is realized as an array, list or sequence. Each object begins with “{“and ends with “}”. Array is an ordered collection of values. An array begin with “[” and ends with “]”. Meanwhile, a value can be a string in double quotes, or a number, or true or false, or an object or an array. Figure 3 shows how the publication data is represented in JSON format.

```
{“article”:[{“id”:”274222”,”author”:”N. Prati”,”title”:”A Partial Model of NP with E.”,”pages”:”1245-
1253”,”year”:”1994”,”volume”:”59”,”journal”:”J.Symb. Log.”,”url”:”db\/journals\/jsyml\/jsyml59.html#Prati94”}],

“book”:[{“id”:”211”,”isbn”:”3-540-60058-2”,”author”:”Marco Cadoli”,”title”:”Tractable Reasoning in Artificial
Intelligence”,”series”:”Lecture Notes in Computer
Science”,”volume”:”941”,”publisher”:”Springer”,”year”:”1995”,”url”:”...”}],

“inproceeding”:[{“id”:”338396”,”author”:”Regine Laleau, Amel Mammar”,”title”:”A Generic Process to Refine a B
Specification into”,”pages”:”22-41”,”year”:”2000”,”booktitle”:”ZB”,”url”:”db\/conf\/zum\/zb2000.html#LaleauM00”}],

“sthesis”:[{“id”:”11”,”author”:”Kurt P. Brown”,”title”:”PRPL: A Database Workload Specification Language,
v1.3.”,”year”:”1992”,”school”:”Univ. of Wisconsin-Madison”}]

“phdthesis”:[{“id”:”1”,”editor”:”Joann J. Ordille”,”title”:”Descriptive Name Services for Large
Internets.”,”year”:”1993”,”month”:”...”,”school”:”Univ. of Wisconsin-Madison”}],

“proceeding”:[{“id”:”1325”,”editor”:”Elen Balka, Richard Smith”,”title”:”Woman, Work and Computerization: Charting
a Course to the Future, IFIP TC9\/WG9.1 Seventh International Conference on Woman, Work and
Computerization, June 8-11, 2000, Vancouver, British Columbia, Canada”,”booktitle”:”Woman, Work and
Computerization”,”series”:”IFIP Conference
Proceedings”,”volume”:”172”,”publisher”:”Kluwer”,”year”:”2000”,”isbn”:”0-7923-7864-4”,”url”:”db\/conf\/ifip9-1\/ifip9-1-
2000.html”}],

“www”:[{“id”:”1”,”editor”:”...”,”title”:”Java Language Home
Page”,”booktitle”:”...”,”year”:”...”,”url”:”http:\/\/java.sun.com\/”}
```

Figure 3. Publication data in JSON format

4. Experimental Results

In this section, we evaluate the performance of the accessing the data from XML and JSON. Four different queries are used in experiments. The systems are build using a personal computer equipped with 2.40GHz Intel® Core ™ i7-5500U CPU, 8.00 GB RAM and a 250 GB solid-state drive. The operating system is Microsoft Windows 10. The database implementing the XML database (approach I) using X-Path for querying purposes and JSON database (approach II). We use benchmark dataset DBLP [19]. The variation in query time with the size of the database is also studied. For each of two database approaches, the time to query and CPU usage with varying complexity specified above is measured with databases containing 1000, 5000, 10,000 and 50,000 records respectively. For query retrieval, at each setting, the query is made for 10 times to calculate the average time and standard deviation [10]. The discussion is based on two experiments in the databases development and their application for the storage of structured data, from the perspectives of test data, query retrieval performance, CPU usage and t-test analysis performance.

4.1. Test Data

The performance of two database approaches is evaluated by using benchmark dataset DBLP. The data contain 50,000 records. Table 8 shows the queries with different complexity and Table 9 shows the queries constructed in the SQL statement.

Table 8. Queries with Different Complexity [5]

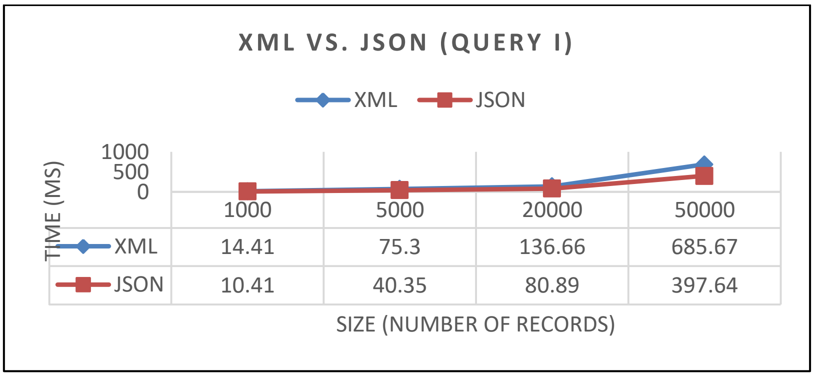
Query	Query description
I	List out all the URLs which begin with the “db/journals” path
II	List out all the titles of the master thesis which contains the “Data” keyword
III	List the titles of inproceeding where the author is “Regine Laleau, Mammar”
IV	Count the number of phd thesis published in each year

Table 9. Queries Constructed in SQL Commands

Query	Query description
I	Select * from url where text like ‘%db/journals/%’
II	Select *from title where text like ‘%Data%’
III	Select title from inproceeding where author=’Regine Laleau, Mammar’
IV	Select count(id), year from phdthesis group by year

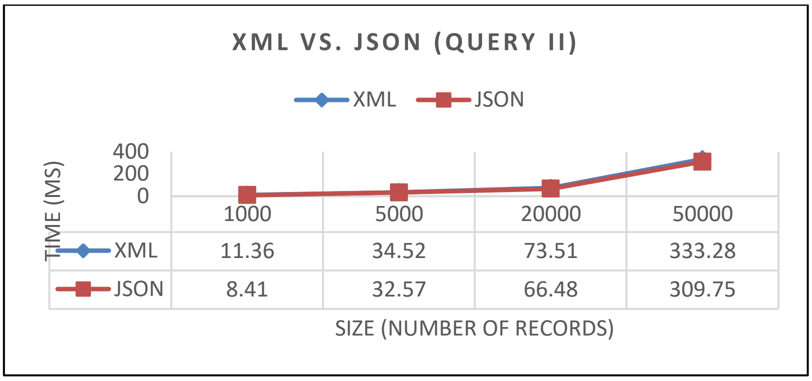
4.2. Query Retrieval Performance (XML vs. JSON)

In this section, we evaluated the performance of search the data from XML and JSON format. Four (4) different queries were executed and time for query retrieval are executes in 10 times. Figure 4 to Figure 7 depict the query retrieval performance in term of time are taken to process the query in milliseconds (ms). The data are split into 5:- 1000 records, 5000 records, 10,000 records and 50,000 records. Mean and standard deviation are calculated based on standard algorithm.



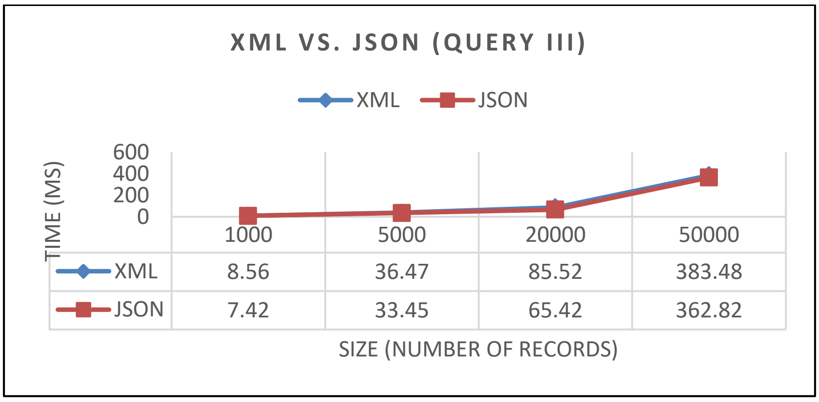
	1000	5000	20000	50000
XML	14.41	75.3	136.66	685.67
JSON	10.41	40.35	80.89	397.64

Figure 4. Query performance of two approaches on database with different size (Query I)



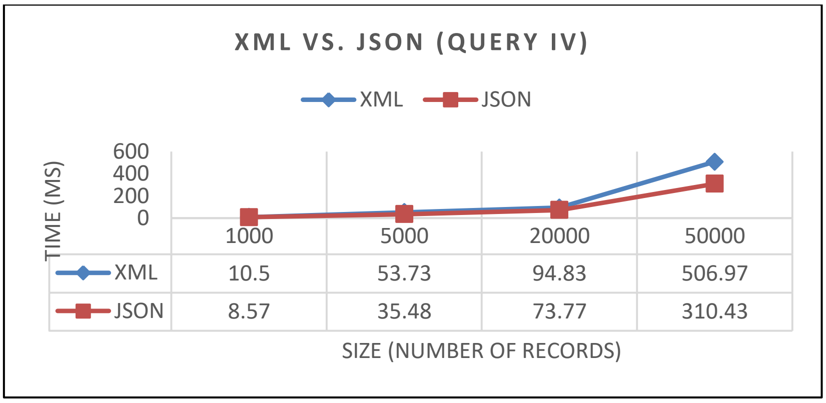
	1000	5000	20000	50000
XML	11.36	34.52	73.51	333.28
JSON	8.41	32.57	66.48	309.75

Figure 5. Query performance of two approaches on database with different size (Query II)



	1000	5000	20000	50000
XML	8.56	36.47	85.52	383.48
JSON	7.42	33.45	65.42	362.82

Figure 6. Query performance of two approaches on database with different size (Query III)

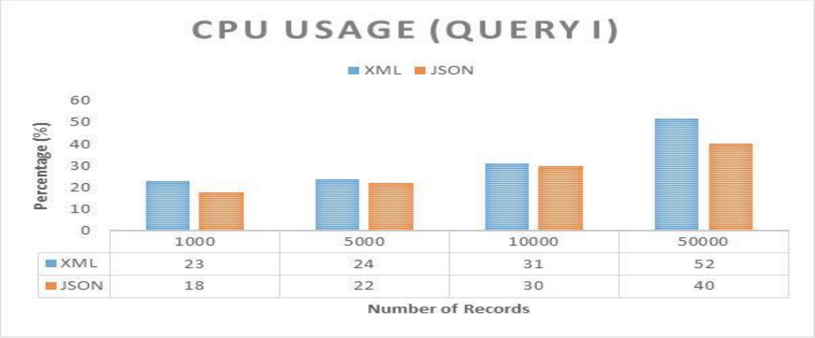


	1000	5000	20000	50000
XML	10.5	53.73	94.83	506.97
JSON	8.57	35.48	73.77	310.43

Figure 7. Query performance of two approaches on database with different size (Query IV)

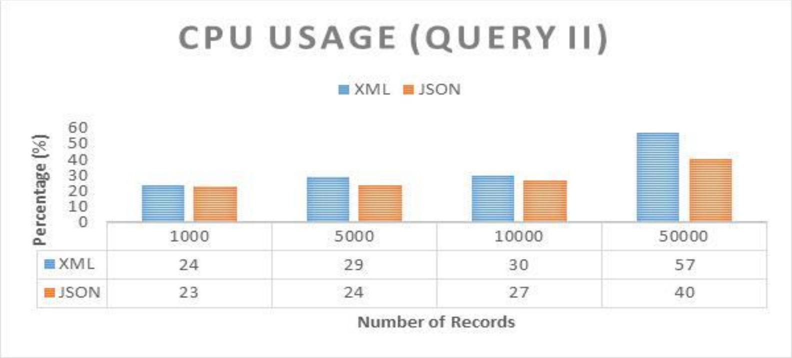
4.3. CPU Usage Performance

The performance of two database approaches is evaluated by using benchmark dataset DBLP. The data contain 50,000 records. Figure 8 until Figure 11 shows the queries with different complexity.



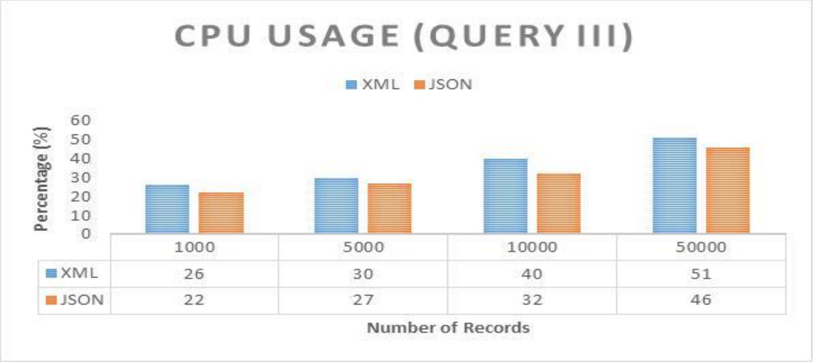
	1000	5000	10000	50000
XML	23	24	31	52
JSON	18	22	30	40

Figure 8. Query performance of the two approaches on database with different size: Query I



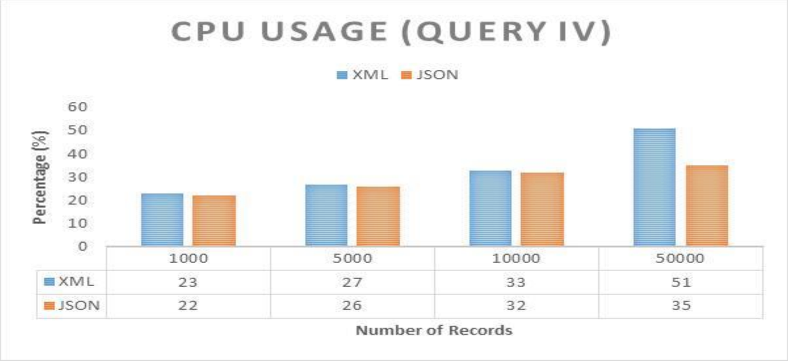
	1000	5000	10000	50000
XML	24	29	30	57
JSON	23	24	27	40

Figure 9. Query performance of the two approaches on database with different size: Query II



	1000	5000	10000	50000
XML	26	30	40	51
JSON	22	27	32	46

Figure 10. Query performance of the two approaches on database with different size: Query III



	1000	5000	10000	50000
XML	23	27	33	51
JSON	22	26	32	35

Figure 11. Query performance of the two approaches on database with different size: Query IV

4.4. T-Test Analysis

T-Test formulation are used to evaluate the significant of execution time. Nine steps involves in t-test formulation [21]. They are:-

a. Find t value and degrees of freedom
b. Determine critical value for t

Table 10. T-Test for XML vs. JSON (Query I)

Criteria's	XML	JSON	XML	JSON	XML	JSON	XML	JSON
	1000 records		5000 records		10000 records		50000 records	
Mean	14.4123	10.4125	11.3558	8.4183	8.5561	7.4181	10.502	8.5734
Variance	0.0798	0.123	0.099	0.1006	0.0751	0.0487	0.1406	0.0804
Stand. Dev.	0.2825	0.3507	0.3146	0.3172	0.274	0.2207	0.375	0.2835
n	10	10	10	10	10	10	10	10
t_i	28.0899		20.7928		10.2277		12.9738	
Degree of freedom	18		18		18		18	
Critical value	2.101		2.101		2.101		2.101	

Table 11. T-Test for XML vs. JSON (Query II)

Criteria's	XML	JSON	XML	JSON	XML	JSON	XML	JSON
	1000 records		5000 records		10000 records		50000 records	
Mean	75.3039	40.3511	34.5243	32.5712	34.4689	33.4511	53.7311	35.4841
Variance	0.1733	0.0428	0.0613	0.1137	0.1022	0.0555	0.1884	0.0955
Stand. Dev.	0.4163	0.2069	0.2476	0.3372	0.3197	0.2356	0.4341	0.309
n	10	10	10	10	10	10	10	10
t_i	237.7756		14.7642		24.0314		108.2843	
Degree of freedom	18		18		18		18	
Critical value	2.101		2.101		2.101		2.101	

Table 12. T-Test for XML vs. JSON (Query III)

Criteria's	XML	JSON	XML	JSON	XML	JSON	XML	JSON
	1000 records		5000 records		10000 records		50000 records	
Mean	136.6557	80.8877	73.5107	66.4804	85.5156	65.4205	94.8307	73.766
Variance	2.0681	0.1805	0.1	0.0544	1.1359	0.9778	0.2877	0.5261
Stand. Dev.	1.4381	0.4249	0.3162	0.2332	1.0658	0.9888	0.5364	0.7253
n	10	10	10	10	10	10	10	10
t_i	117.6092		56.5883		43.7101		73.8393	
Degree of freedom	18		18		18		18	
Critical value	2.101		2.101		2.101		2.101	

Table 13. T-Test for XML vs. JSON (Query IV)

Criteria's	XML	JSON	XML	JSON	XML	JSON	XML	JSON
	1000 records		5000 records		10000 records		50000 records	
Mean	685.7859	397.343	333.2841	309.7495	383.4779	362.8226	506.966	310.4284
Variance	8.7895	5.2784	9.6759	1.5833	3.376	0.7878	2.9617	0.5619
Stand. Dev.	2.9647	2.2975	3.1106	1.2583	1.8374	0.8876	1.721	0.7496
n	10	10	10	10	10	10	10	10
t_i	243.186		22.1794		32.0096		331.1018	
Degree of freedom	18		18		18		18	
Critical value	2.101		2.101		2.101		2.101	

The results in Table 10 until Table 13 indicates t_i is less than critical value. The means of XML and JSON in Table 10 until Table 13 are significantly different at $p < 0.05$.

5. Conclusion

Two different database approaches have been implemented in this research to be one of the alternative traditional database approach also knows as relational database. Three experiments have been conducted which are query retrieval performance, CPU usage and T-Test analysis. Dataset DBLP has been used to conduct the experiments above. This dataset are divided into four (4) part of data amount; 1,000 records, 5,000 records, 10,000 records and 50,000 records. The part of data amount is important to measure the scalability of query retrieve performance when involves with huge data. Four (4) different types of queries are tested to measure the query retrieval performance and CPU usage.

In query retrieval experimental, the results in Figure 4 until Figure 7 indicates, JSON approach is faster compare to XML approach. In JSON, elapsed time in Figure 4 until Figure 7 shows slowly increased depend on number of records. This results indicates JSON approach is stable and flexible. Meanwhile, in XML, elapsed time in Figure 5 and Figure 6 shows steadily increased but in Figure 4 and Figure 7, the elapsed time rapidly increased for 50,000 records. In this case, XML is quick unstable and inflexibility compared to JSON approach.

In CPU usage experiments, Figure 8 until Figure 11 indicates JSON approach is better compared to XML approach. In JSON approach, Figure 8 until Figure 11 shows CPU usage are increase steadily depends on number of records. In XML approach, the results in Figure 8 until Figure 11 shows CPU usage are increase steadily. But, in Figure 9 and Figure 11, XML approach shows CPU usage are increase significantly compared to JSON approach. This results shows JSON approach is more stable and flexible.

In t-test experiments, value of elapsed time for each queries execution time are calculated in order to get means value. The means of JSON and XML are measured to identify either the means is significant or not. Table 10 until Table 13 shows, means of JSON and XML are significant. In this case, JSON and XML is reliable.

The study attempts to explore the vast opportunities JSON technologies in management of huge data. The prototype system developed is initially tested with maximum of 50,000 records only. Further evaluation using larger datasets, or even multiple databases and data warehouse, should give more comprehensive and thorough findings on the performance of query retrieval and CPU usage of XML and JSON approaches.

References

[1] Sulistyo Heripracoyo & Roni Kurniawan. Big Data Analysis with MongoDB for Decision Support System. *TELKOMNIKA*. September 2016; 14(3): 1083–1089.

[2] Maureen Molly Knapp. Big Data. *Journal of Electronic Resources in Medical Libraries*. Nov 2013; 10(4): 215–222.

[3] Ralph Schroerder. Big data business models: Challenges and Opportunities. Cogent Social Science. 2016: 1-15.

[4] B. Douglas Blansit MPS, MLIS. The Basics of Relational Database Using MySQL. *Journal of Electronic Resources in Medical Libraries*. 2006: 135-148.

[5] Haw Su Cheng, Lee Chien Sing, and Norwati Mustapha. Bridging XML and Relational Databases: Mapping Choices and Performance Evaluation. IETE Technical Review. Jul-Aug 2010: 4(40: 308-317.

[6] Miki Enoki, Jerome Simeon, Hiroshi Horri, Martin Hirzel. Event Processing over a Distributed JSON Store: Design and Performance. WISE 2014, Part II, LNCS 8787. 2014: 395–404.

[7] Nurzhan Nurseitov, Micheal Paulson, Randall Reynolds, Clemete Izurieta. Comparison of JSON and XML Data Interchange Formats: A Case Study.

[8] Dunlu Peng, Lidong CAO, Wenjie XU. Using JSON for Data bn Exchanging in Web Service Applications. *Journal of Computational Information System*. 2011; 7(16): 5883–5890.

[9] M Meneghello. XML (extensible markup language)–The New Language of Data Exchange. *Cartography*. June 2001; 30(1): 51–57.

[10] Ken Ka-Yin Lee, Wai-Choi Tang, Kup-Sze Choi. Alternatives to relational database: Comparison of NoSQL and XML approaches for clinical data storage. *Computer Methods and Programs in Biomedicine*. 2013; 110: 99-109.

[11] Andrew Clarke, Eric Pardede, Robert Steele. External and Distributed Databases: Efficient and Secure XML Query Assurance. *International Journal of Computational Intelligence System*. 2012: 421–433.

[12] Haw Su-Cheng and Lee Chien Sing. Efficient Preprocesses for Fast Storage and Query Retrieval in Native XML Database. *IETE Technical Review*. Sep 2014; 26(1): 28–40.

[13] SM Bachrach. Integration of Chemical Data using XML. *SAR and QSAR in Environmental Research*. 2002; 13(3-4): 381-390.

[14] Decheng Qiu, Junning Liu. Design and Application of Data Integration Platform based on Web Services and XML. *IEEE*. 2016: 253–256.

[15] Ankit Bharthan and Devesh Bharathan. Relational JSON, An Enriched Method to Store and Query JSON Records. *International Journal of Computer Application*, July 2014; 98(7): 1–4.

[16] Zhen Hua Liu, Beda Hammerschmidt, Doug McMohan. *JSON Data Management–Supporting Schema–less Development in RDBMS*. Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. June 22–27: 1247–1258.

[17] Mohd Kamir Yusof, Mustafa Man. Efficiency of JSON Approach for Data Extraction and Query Retrieval. *Indonesian Journal of Electrical Engineering and Computer Science*. Oct 2016; 4(1): 203-214.

[18] Hugh Darwen. An Introduction to Relational Database Theory. bookboon.com, 2010.

[19] DBLP, Available from https://kdl.cs.umass.edu/display/public/DBLP

[20] T.J. Quirk et al., Two-Group t-test of the Difference of the Means for Independent Groups. Excel 2007 for Biological and Life Sciences Statistics, 2013: 77–102.