Database space saving using proxy attributes for green data centers

Nurul A. Emran[1,*], Noraswaliza Abdullah[1], Nor Hafeizah Hassan[1]

[1]Centre for Advanced Computing Technology (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

[*]Corresponding e-mail: nurulakmar@utem.edu.my

Keywords: Space saving, green data center, proxy-based

ABSTRACT – One challenge in ensuring environmental quality is to minimize CO_2 emissions**.** Green Data Centre concept is proposed to diminish carbon footprint that will reduce CO_2 emission. In this paper, we argue that by minimizing the number of physical data storages to store large data volumes inefficient electric power utilization can be avoided. In this paper, a method to reduce space for storage by optimizing the available database space through attributes substitution is proposed. We present a space-saving strategy called as 'proxy'-based approach that offers attributes substitution through database schema modification. The results show how this method works to optimize database storage space while maintaining query accuracy.

1. INTRODUCTION

The requirement to improve environmental quality in urban space by reducing CO_2 emissions is undeniable in many applications. This requirement can be fulfilled by subscribing to the green data center (or green cloud) concept that offers to decrease carbon footprint and operating costs (e.g. for cooling systems). As the size of the population grows, it is anticipated that telecommunications networks become the medium for the large volume machine-to-machine data transmissions. As the consequence, data providers need to deal not only with data volume issues but also with other data quality issues that call for efficient solutions [1]. Expanding database storage is a way that can be considered by data center providers, however, this method contributes to an additional number of physical data servers. Consequently, the amount of electrical power will increase to accommodate the additional data servers and to cooling-off those servers. A recent estimation stated that the world's data centers consume about 330 billion kWh of electricity annually which is almost equal to the entire electricity demand of the UK [2]. Power consumption that is more than 100 billion kWh generates approximately 40, 568, 000 tons of CO_2 emissions. Thus, in establishing successful green data centers, adding more data servers is not an attractive option to choose in addressing the storage space issue.

2. METHODOLOGY

We argue that by minimizing the number of physical data storage needed to store large data volumes we can avoid inefficient power consumption (which will contribute to unnecessary CO2 emissions). A common method to minimize storage space is by data compression. However, data compression requires a de-compression process that will require the original amount of space before the data can be viewed. This will only a temporary solution for the space problem. In this paper, we focus to study a space-saving technique called as 'proxy'-based approach that applies attributes substitution through database scheme modification.

2.1 Proxy-based Space Saving

The use of the term proxy can be seen in several domains such as in web application and cloud computing where proxies in these contexts refer to the permission to act as a substitute agent of another [3]. In this research, the proxy concept is evaluated in handling storage space-saving. The space savings can be offered via database schema modification and by removing the redundant data from the table. In this approach, free spaces are gained by deleting selected attributes from tables. To illustrate this idea, suppose that *Table A* (as shown in Figure 1) consists of 1000 tuples, and column *c* is selected as the droppable attribute. The amount of space-saving at this stage is 1000 tuples. The consequence of the deletion is information loss in *Table A*. Nevertheless, the loss can be compensated by a structure that is called a proxy map table which is smaller (in size) relative to the size of the dropped attributes. Proxies for the dropped attributes must be selected to avoid query failures. Proxy candidates need to be selected before the proxy map table can be built.

In this approach, the proxy and droppable attributes will be determined by discovering the relationship of the attributes in the table where functional dependency (FD) is present. The proxy candidates need to be related to the dropped attribute to ensure that the replacement works. All the proxy candidates will be stored in the proxy map table. Optimization of storage space also can be gained when the redundancy of data in the table is removed when a proxy-based approach is applied. With the use of a proxy map table, the redundancy of metadata can be removed because it will map the unique values of data. The example of mapping in proxy map table is as the following, where *x* and *y* are the values of the dropped attributes (e.g *Column c*), being mapped to the values of the proxy attribute's values (e.g *Column b*):

x -> {1,2,3,4}
y -> {5,6,7,8}

Therefore, every query against *Table A* that consists of *Column c's* values in the predicate will be directed to the proxy map, after a transformation. The

use of a proxy map will require additional space where, if it is in its optimal size, space-saving can be gained.

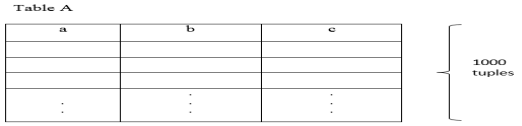

Figure 1 An example of schema modification

2.2 Experiment Design

An experiment is conducted to evaluate the amount of space-saving and the accuracy of the queries that are executed with proxies (based on the amount of error). The flow of the experiment is as shown in Figure 2. Comprehensive Microbial Resource (CMR) data sets that cover microbial bacterial genome types is used in the experiment. The data set consists of genomes collections with their annotations, which is downloadable from the website link: *ftp://ftp.tigr.org/pub/data*

CMR datasets consist of 24 tables of data. In the experiment, three data sets with varying sizes were taken as samples which are *Taxon* (723 rows), *Bug_attribute* (10,165 rows), and *Role_link* (934,206 rows). A data mining algorithm called a TANE algorithm (see [4]) is used to determine functional dependency (FD) scores (called as g3 error) among the columns. Proxy candidates are drawn based on the value of g3 error. Proxy candidates are generated among the non-key attributes with the lowest g3 error. This is because, even though key attributes guarantee substitution accuracy, the proxy map table generated will consume undesirable space as every key value will be mapped to the dropped attribute's value. This will defeat the purpose of the proxy. 30 random queries are used in the experiment for each table under study.

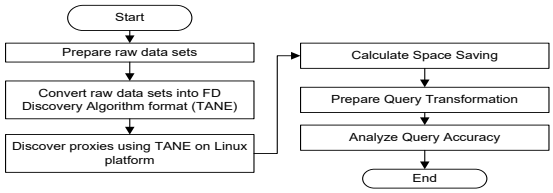

Figure 2 Experiment flow for proxy evaluation

The schema of these tables are as follow:

I. `Taxon(id, taxon_id, kingdom, genus, species, comment, strain, i_rank_1, i_rank_2, i_rank_3, i_rank_4, i_rank_5, i_rank_6, short_name)`
II. `Bug_attribute(id, db_data_id, att_type, method, date, assignby)`
III. `Role_id(id, locus, role_id, AssignBy, datestamp)`

3. RESULT AND DISCUSSION

Table 1 illustrates the amount of space savings (in percentage) for each table after proxies are embedded in the queries. *Role_link* table exhibits the highest amount of space saving. This is followed by the *Bug_attribute* with 13% space savings and finally *Taxon* (8% and -0.12%) Based on the results, we can learn that proxy P1 for *Taxon* is not useful in space saving. However, proxy P2 form the same table offers space-saving by 8%. As the size of the table increases, more space savings can be gained.

Table 1 Space saving results

Table Name	Proxy	Main Table Size (in Bytes)	Size of Proxy Table (in Bytes)	Space Saving (%)
Taxon	P1	68685	-85	-0.12
Taxon	P2	68685	5583	8
Taxon	P3	559075	72275	13
Role_link	P4	60723390	12799254	21

Table 2 shows the results of query accuracy together with the amount of space-saving. Overall, the average error is low (less than 6%), with P3 offers highly accurate queries with the highest amount of space-saving. It can also be observed that g3 error does not influence the query errors.

Table 2 Accuracy and space saving results

Table	Proxy	Space Saving (%)	*Average Error (%)	g3 error (%)
Taxon	P2	8	5.18	0.10
Bug_attribute	P3	13	0	0.06
Role_link	P4	21	2.46	0.02
* Av. error =total (%) error in all queries/total number of query				

4. CONCLUSIONS

In conclusion, the results presented in this paper show the usefulness of the proxy-based storage-saving approach. The proxies have been evaluated in terms of space-saving and query accuracy. Within the scope of the data set under study, useful proxies can be found in all tables. The adoption of proxies will depend on the space-saving and accuracy threshold being set by the implementers. The findings contribute to understanding an alternative for data compression in optimizing storage space.

ACKNOWLEDGEMENT

Authors would like to acknowledge the Universiti Teknikal Malaysia Melaka for supporting this research.

REFERENCES

[1] N. Z. Abidin, A. R. Ismail, and N. A. Emran, Perf. analysis of machine learning algorithms for missing value imputation, *Int. J. Adv. Com. Sci. Appl.*, vol. 9, no. 6, pp. 442-447, 2018.

[2] "How dirty is your data? - Greenpeace International." [Online]. Available: https://www.greenpeace.org/international/publication/7196/how-dirty-is-your-data/.

[3] Hjertström, D. Nyström, and M. Sjödin, Data management for component-based embedded real-time systems: The database proxy approach, *J. Syst. Softw.,* vol. 85, no. 4, pp. 821–834, 2012.

[4] Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen, TANE: an efficient algorithm for disc. func. and approx.. dependencies, *Comp. J.,* vol. 42, no. 2, pp. 100–111, 1999.