3rd International Conference on Energy and Environment: bringing together Engineering and Economics
Porto, Portugal
29-30 June, 2017

AN ECONOMIC ENERGY APPROACH FOR QUERIES ON DATA CENTERS

João Saraiva,[1] Miguel Guimarães[2] and Orlando Belo[2*]

[1] HasLab R&D Center, INESC/University of Minho, Portugal
[2] ALGORITMI R&D Center, University of Minho, Portugal

* Corresponding author: obelo@di.uminho.pt, University of Minho, Campus de Gualtar, 4710-057 Braga, Potugal

KEYWORDS
Energy System Analysis, Database Systems, and Green Queries.

ABSTRACT

Energy consumption is an issue that involves all of us, both as individuals and as members of a society, and covers all our areas of activity. It is something so broad that its impact has important reflections on our social, cultural and financial structures. The domain of software, and in particular database systems, is not an exception. Although it seems to be a little bit strange to study the energy consumption of just one query, when we consider the execution of a a few thousand queries per second, quickly we see the importance of the querying consumption in the monthly account of any company that has a conventional data center. To demonstrate the energy consumption of queries in data centers, we idealized a small dashboard for monitoring and analyzing the sales of a company, and implemented all the queries needed for populating it and ensuring its operation. The queries were organized into two groups, oriented especially to two distinct database management systems: one relational (MySQL) and one non relational (Neo4J). The goal is to evaluate the energy consumption of different types of queries, and at the same time compare it in terms of relational and non-relational database approaches. This paper relates the process we implemented to set up the energy consumption application scenario, measure the energy consumption of each query, and present our first preliminary results.

INTRODUCTION

For a long time, computer manufacturers and software developers primary and single goal was to produce very fast computers and software systems. In this century this has changed significantly. The widespread use of non-wired but powerful computer devices is making battery consumption/lifetime the bottleneck for both manufacturers and software developers. The hardware manufacturers have already realize this concern and much work in terms of optimizing energy consumption by improving the hardware has being done. Unfortunately, the software engineering community has not yet completely realizes this bottleneck, and as consequence there is little support for software developers for reasoning about software energy consumption. Although it is the hardware that consumes energy, is the software that operates the hardware, and as consequence, it can greatly influence such consumption, very much like a driver that operates a car influences its fuel consumption. In fact, the software has a pivotal role on the energy consumed by the hardware: "up to 90% of energy used by ICT hardware can be attributed to software" according to *The Greenhouse Gas Protocol Report* (GGP, 2017). Energy consumption is not a concern of non-wired computer devices, only. In fact, the fast increasing demand for large data centers emerged during the last years associated with the proliferation of mobile devices and the access to the Internet. As a consequence, energy consumption is particularly relevant in data centers, since huge amounts of information are stored and manipulated to satisfy the large community of users that usually access it. Social networks, multimedia data instant brokers, large retail platforms, and telecommunications services managers are particular sensitive to the number of transactions (including queries) that are triggered every minute over their systems, especially with processing and storage requirements. However, during the last few years another issue toke the attention from systems' administrators: energy consumption. Meanwhile, many IT companies and professional carried a large diversity of studies about the energy demands of data centers, inspecting application scenarios and proposing solutions for reducing energy consumption without affecting their regular operation and quality of service (Kumar, 2010). The analysis and evaluation of energy consumption in a data center can be done appealing to several information elements of components involving aspects related to general infrastructures (cooling, power conversion, lighting, etc.) and hardware (servers, storage, network, etc.) of data centers. Today, we know that the energy consumption of a data center represents about one third of the costs involved with the general maintenance of the entire system (Rasmussen, 2011). This means a lot of money and, consequently, calls a lot of attention. The European Community itself takes this problem very seriously. It promoted already a specific program to draw attention to the need to understand energy consumption in a data center and prepare a code of conduct for data centers. Till a few years ago, the large majority of the efforts conduced for improving energy efficiency was applied in the area of hardware components. But today IT researchers and technicians direct a particular attention to software components. It was recognized that these components also contribute with a significant parcel when we are speaking about the energy consumption of large computational platforms like the ones we have supporting data centers.

In this paper we will present and discuss a research work on how to establish the energy consumption for transactional systems, giving particular attention to some selected data operations that usually occur in data centers. Basically, we implemented and measured the energy consumption of a specific set of queries, especially conceived for populating a business analytical dashboard. Our main goal was to discover a way to reduce as much as possible the energy consumption carried out by transactional systems without affecting the overall performance of data centers. To do that, we selected two distinct transactional DataBase Management Systems (DBMS), one relational – MySQL (MySQL, 2017) –, and one non-relational – Neo4J (Neo4J, 2017) –, in order to evaluate their performance – execution time and energy consumption – when supporting operation of the referred analytical dashboard. Having two types of DBMS in analysis we enlarged our test bed in order to cover in our study non-relational databases approaches. The high emergence in the market of the new non-relational systems deserve to be covered in this study, giving us the opportunity to compare them as well, providing a more generalized view and practical approach.

In next sections we present a brief related work, exposing some of the most relevant works done in the domain and related areas, how we prepared and applied the method we designed for evaluating the energy consumption of the queries, giving a general view of the business analytical dashboard we designed and analyzing some of the most relevant queries, relational and non-relational, we used for populating the dashboard, and the results we got, both in terms of performance and energy consumption. Finally, we end this paper with our conclusions, pointing out future work.

RELATED WORK

Nowadays the demand for energy efficiency is heavily present in different fields of expertise. Since our excessive dependency on non-renewable energy to power up all the electronics that we have at our disposal, it is important to find solutions that might help to reduce the energy bill costs. This concern over researching and developing techniques and frameworks to overcome the massive energy consumption, may lead to solutions that has the ability to help solving the energy crisis that the planet faces today. Therefore, developing green software can contribute significantly to preserve the environment resources and reduce the energy consumption costs. A large variety of steps towards the implementation and developing of more energy-friendly and energy-aware approaches was already made in the domain of information and communications technology. The research topic regarding energy efficient software has been tackled in different systems, and the work done on this subject is a solid proof of paradigm shifting in software development. The works developed by Carção (2014), Couto et al. (2014), and Pereira et al. (2016) are some good examples of methods and techniques that allows for measuring the energy consumption at the software's source code level. In (Carção, 2014) an adaption of the Spectrum-based Fault Localization technique to the energy context, in order to detect non-green spots in the program's source code . The work developed by Couto et al. (2014) aimed for finding and detecting anomalous energy consumption in Android Systems. In (Pereira et al., 2016), authors analyzed different Java data structures implementations included in Java Collection Framework (JCF) and defined a green ranking of Java collections. Database systems are no exception to the green movement, a pioneer approach concerning energy consumption was proposed by (Agrawal et al., 2008). The Claremont report main goal was to take into consideration, during the devise and implementation stages the amount of energy consumed by different tasks. Harizopoulos et al. (2009) call attention to different characteristics that might help improving the energy efficiency on data centers. The work presented in (Wang et al., 2011) provides a nice survey about energy efficiency in data management operations. Other research works on energy consumption have also emerged, like the ones presented in (Lang and Patel, 2009) and (Lang et al., 2011). Nevertheless, the focus of those works was centered in hardware base premises. In terms of software, Xu et al. (2010) presented a solution involving the redesign of the DBMS kernel to reduce the amount of energy consumption. Later, the research work done by Kunjir et al. (2012) presented some alternatives to reduce the peak of energy consumption in database management systems. Additionally, Rodríguez et al. (2013) developed some work related to the prediction of the energy consumption of join queries, while Xu et al. (2012), following a similar research line, focused on query optimization with the objective of reduce the energy consumption. More recently, Gonçalves et al. (2014) redesigned the DMBS execution plan to include the estimated value of energy consumed for the most commons database operators as well as the energy estimation for the overall query. Afterward, these authors extended a previous work on a different domain, measuring the energy consumption for star-queries in a data warehousing system environment (Belo et al., 2015). Later on, Guimarães et al. (2016) devised a set of heuristics as a basis recommendation for reducing the energy consumption of a given query inside a relational DBMS. However, today non-relational database systems, also known as NoSQL DBMS, start to be a regular presence in some data centers. Thus, it seems to us to be also important to evaluate the energy efficiency of these emergent DBMS. As a step towards the implementations of greener NoSQL queries, Duarte and Belo (2017) gave a small but important step regarding the evaluation of NoSQL database management systems, with particular focus on document stores based systems. In this work we extend our experience on energy consumption evaluation in database, evaluating and comparing the energy efficiency between a relational and a non-relational DBMS supporting a business intelligence application that could be installed in a regular data center.

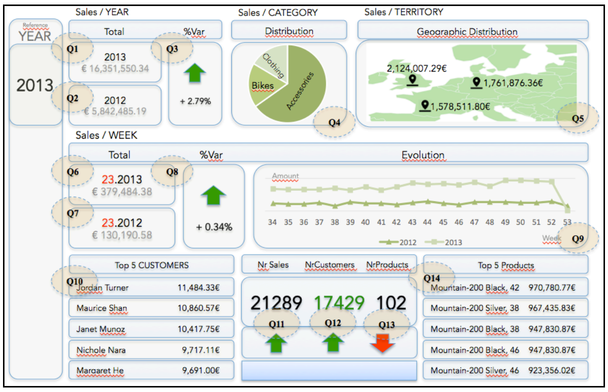

Figure 1: The business analytical dashboard used in the energy consumption test.

EVALUATING QUERYING ENERGY CONSUMPTION

The Case Study

Query processing is one of the most important activities performed by a DBMS supporting regular business activities. In many cases, especially in the business intelligence area, queries are launched in a systematic way to support ad hoc analysis, feed business reports or populate analytical dashboards. This last case imposes regularly a "stream" of queries over one or more DBMS in order to keep up-to-date all the elements that are integrated in their structure. Dashboards are useful analytical tools for faster analysis processing. They are a collection of graphics and visual reports that allow for a rapid access to critical information in an easy readable notation. A business dashboard can provide very interesting pieces of information using several distinct components – e.g. charts, gadgets, maps, tabular reports, ranks, or key performance indicators – in a very powerful and fancy way. Thus, evaluating the energy consumption of all the queries that feed regularly a business dashboard, often several times a minute, provides us a very clear picture about the energy that is consumed by all their queries, and so about the influence of these querying components in a computational platform like data centers.

Table 1: The set of queries used in the energy consumption test.

Nr	Query	NrOfRecords
Q1	Sales of year 2012	1
Q2	Sales of year 2013	1
Q3	Sales variance of year 2013 vs. year 2012	1
Q4	Sales per category in 2013	3
Q5	Sales per region in 2013	10
Q6	Sales per week in 2012	53
Q7	Sales per week in 2013	53
Q8	Sales variance per week in year 2012 vs. year 2013	1
Q9	Sales evolution per year 2012 vs. year 2013, by week	106
Q10	Top 5 Customers in 2013	5
Q11	Number of sales in 2013	1
Q12	Number of distinct customers in sales in 2013	1
Q13	Number of distinct products sold in 2013	1
Q14	Top 5 Products in 2013	5

In our case study, we choose to compose an illustrative dashboard with fourteen visual elements and displayed in Figure 1. This dashboard is populated with data coming from an instance of the "Adventure Works Data Warehouse" (AWC, 2017), a sample database provided by Microsoft for a data warehouse of a fictitious company - Adventure Works Cycles, which was extracted from a specific data mart: "Internet Sales". To populate the dashboard we need to design and implement fourteen distinct queries, which are indicated in Figure 1, with tags ranging from Q1 to Q14. Usually, a conventional business dashboard is supported by relational DBMS, and all the queries are expressed in SQL: a declarative language, which has the ability to cover all the various types of database operation branches. However, in this paper we will restrict our attention only to select operations, the ones that are responsible for retrieving data from databases. Moreover, and as already referred, we extended our case study to a non-relational DBMS. This imposed the translation of the previous 14 SQL queries, implemented in MySQL, to Cypher queries in order to execute them in the Neo4J DBMS, which is supported by a non-relational format based on graphs. At the end, we got 28 different queries organized into two distinct groups: relational (SQL) and non-relational (NoSQL). In Table 1 we can see the meaning and the length (in records) of the result set that each query produced, independently from the support it has. In order to provide a more concrete view about the kind of queries we used in our study, in Figure 2 we present one of the most energy consuming query, the query Q8 (Sales variance per week in year 2012 vs. year 2013), both in SQL and NoSQL format instructions.

a)	`SET @V1 = NULL;SET @V2 = NULL; SET @V1 = (SELECT SUM(F.SalesAmount) AS TOTAL FROM FactInternetSales AS F INNER JOIN DimDate AS T ON F.OrderDateKey = T.DateKey WHERE T.CalendarYear = 2013 AND T.WeekNumberOfYear = 23 GROUP BY T.WeekNumberOfYear); SET @V2 = (SELECT SUM(F.SalesAmount) AS TOTAL FROM FactInternetSales AS F INNER JOIN DimDate AS T ON F.OrderDateKey = T.DateKey WHERE T.CalendarYear = 2012 AND T.WeekNumberOfYear = 23 GROUP BY T.WeekNumberOfYear); SELECT @V2/@V1 WeekSalesVariance`
b)	`MATCH(F:FactInternetSales)-[:ORDER_AT]->(D) WHERE D.CalendarYear = 2013 AND D.WeekNumberOfYear = 23 WITH SUM(F.SalesAmount) as v1 MATCH(F:FactInternetSales)-[:ORDER_AT]->(D) WHERE D.CalendarYear = 2012 AND D.WeekNumberOfYear = 23 RETURN SUM(F.SalesAmount)/v1 as WeekSalesVariance`

Figure 2: The SQL (a) and NoSQL (b) variants for Q8 - Sales variance per week in year 2012 vs. year 2013.

Data and Test Configuration

The DBMS paradigm is gradually shifting from performance to energy, and that is visible by the different efforts that are being made. In this work, we extended a tool that was previous developed by us named gSQL (Guimarães et al., 2016). This tool has the ability to measure and categorize which SQL queries are green and which are not. However, and for the case study that we conduct in this paper, the gSQL tool should be modified to accommodate not only SQL queries but NoSQL queries as well. To measure the energy consumption this tool uses intel jRAPL framework: a Java-based energy consumption estimation mechanism provided by modern intel CPU architectures (Liu et al., 2015). However, this framework needs to fulfill certain specific conditions. The processor used must be an Intel, with support for Machine-Specific Registers (MSR), which will be responsible to store the values for the energy consumption for a certain block of code. Since, the original version gSQL tool lacks support for NoSQL queries, we have extended it to support them as well. To obtain the energy consumption information for SQL or NoSQL queries from gSQL tool, there are a specific list of parameters that should be defined to produce the desired results, namely: a) a configuration file, with the necessary parameters to connect to the DBMS in JSON – e.g. hostname, port, database name, username, and password; b) an input file, with all the queries that will be tested; c) the number of times each test will be repeated; and d) the number of times each query will be repeated.

```
begin
   resultsList ← initializeResults();
   for each query in queriesList do
     begin
          for each execution in executionsList do
             begin
                for each repetition in repetitionList do
                begin
                   initialEnergy ← getEnergy(); initialTime ← getTime () executeQuery(query);
                   finalEnergy ← getEnergy();
                   finalTime ← getTime(); energy ← initialEnergy - final;
                   energy time ← finalTime - initialTime;
                   storeValues(resultsList, energy, time);
                   end
             end
       end
   aggregate ← aggregateResults(results);
   writeFile(aggregate);
end
```

Figure 2: An excerpt of the behavior of the gSQL tool in pseudo code.

In Figure 2, we can present, written in pseudo code, the behavior of the gSQL tool. The results obtained from executing the gSQL tool are the energy consumption and time spent by each query executed. The values are aggregated according to four distinct formulas: maximum, minimum, average, and standard deviation. The amplitude between the maximum and minimum values alongside with the standard deviation value, help us to support the decision if a certain query test needs to be executed again.

ANALYSIS OF RESULTS

All the tests were made in a controlled environment with the machine running only the necessary core processes. This allows us to dissipate the influence of some “pollution elements” that may interfere with the measurements. From the premise that only Intel architectures are suitable to the gSQL tool, the testes were executed in an Intel Core i3-2100

Processor with 3M Cache at 3.10 GHz and a total of 8Gb of RAM. The DBMSs used were MySQL and Neo4j, both populated with the same data provided by the Adventure Works Data Warehouse data set (AWC, 2017). Due to the complexity associated with the data set, a small sample was chosen to be the target of the test queries. Each test was executed 10 times and each query executed 15 times. The data provided by the gSQL tool about the dashboard queries tested by both DBMS (MySQL and Neo4j), offers us some interesting results to conclude our case study: to find out which one of the approaches (SQL and NoSQL) is the greenest. Observing Table 2, it is possible to analyze the average energy consumption in Joules and the average time in seconds spent by each query in a SQL and NoSQL representation. If we observe the values of the average energy consumption for SQL and compare them to the values of the average energy consumption for NoSQL, the conclusion is evident: the approach that consumes less energy to populate all the dashboard elements is the SQL alternative supported by MySQL - every single query in the SQL database engine consumes less energy than its NoSQL counterpart.

Table 2: Average Energy Consumption and Average Time Results.

	Average Energy (Joule)		Average Time (Second)	
Query	**SQL**	**NoSQL**	**SQL**	**NoSQL**
Q1	0.08479	2.63451	0.00723	0.20065
Q2	0.89784	4.49449	0.06719	0.32716
Q3	1.40829	7.15565	0.10653	0.51698
Q4	3.74356	9.04041	0.28339	0.65435
Q5	2.32553	6.4678	0.17902	0.4736
Q6	0.1168	2.68482	0.00803	0.20135
Q7	1.22223	4.63566	0.09047	0.33742
Q8	0.09425	9.56032	0.00665	0.70984
Q9	1.31339	7.84458	0.10182	0.57301
Q10	3.12084	9.22695	0.2445	0.58421
Q11	0.95184	2.50374	0.07485	0.18468
Q12	0.98814	4.07412	0.07493	0.29969
Q13	0.87587	4.31006	0.06584	0.31409
Q14	3.93677	6.23006	0.30894	0.45287

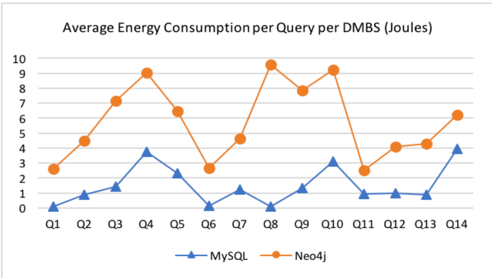

Figure 3: Average energy consumption per query.

The chart presented in Figure 3 offers another perspective over the data collect by the gSQL tool. We can observe that queries Q4, Q8, and Q10 have the highest energy consumption on Neo4j DBMS and Q4, Q10 and Q14 have the highest energy consumption on MySQL DBMS. For the NoSQL DBMS, the measured values can be explained by the query characteristics. For instance, in the NoSQL variant of the query Q4, the MATCH operation is more complex and takes more time to be executed, and consequently consumes more energy (Table 2). This is also true for the traditional database engine, where the query Q4 has a higher join complexity, thus taking more time to execute and therefore consuming more energy. Regarding the Neo4j DBMS, the NoSQL variant of the query Q8 reveals some performance inefficiency, in part due to the operator WITH. It also happens in the case of NoSQL queries Q3 and Q5, both using the WITH operator in the dashboard query tests. The purpose of this operator is to store temporarily the data so it can be used as a starting point (or criteria) for the next query. Even thought, being quite fast to do it (Table 2), the MySQL approach is far more efficient than the Neo4j solution. Due to the different approaches for solving the same problem, we can observe the highest amplitude between the average value measured for the query Q8 on MySQL and Neo4j (as shown in Figure 3) among all the other queries. MySQL offers some operators that do not have a direct translation to Neo4j querying instructions. The SQL approach stores in temporary variables the intermediate values for the SELECT operations (see Figure 2) whereas, the Neo4j approach uses the WITH operator to temporarily store all the nodes calculated in the first MATCH operation, thus the disparity between the measured values.

CONCLUSIONS AND FUTURE WORK

During the last few years, energy consumption has been the target of some serious research efforts in the area of data centers, especially in the aspects related to query execution. Processing a query is one of the most important activities performed by a DBMS, independently from the model it uses for describing, manipulating or control data. In the majority of DBMS, queries are expressed in SQL, which makes it an excellent instrument for studying energy consumption in data centers. In this exploratory study, we extend our previous work on querying energy consumption to the field of NoSQL, in order to understand, in terms of energy consumption, which DBMS approach – SQL or NoSQL – is the greenest. Using as a case study a business dashboard, we evaluate the energy consumption (and the execution time) of two sets of queries, SQL and NoSQL, trying to establish an economic model that shows us what kind of DBMS is the greenest. Based on the results we got, we can say that the Neo4j DBMS has an higher energy consumption than the MySQL DBMS, in general terms. However, at short term, we want to refine this work, trying to be more precise on the characterization of the model we used to evaluate querying energy consumption, being more effective taking into consideration the basic operators that DBMS use on the execution plans they establish for querying, especially in the field of NoSQL DBMS.

REFERENCES

Agrawal, R. et al., 2008. "The claremont report on database research". SIGMOD Rec. 37(3), September, pp. 9–19.

AWC, 2017. Microsoft Adventure Works Database Example. [online] Available at: < https://msftdbprodsamples.codeplex.com> [Accessed 17 March 2017].

Belo, O., Gonçalves, R., Saraiva, J., 2015. "Establishing Energy Consumption Plans for Green Star-Queries in Data Warehousing Systems", The 2015 IEEE International Conference on Green Computing and Communications (GreenCom 2015), Sydney, Australia, December, 11-13.

Carção, T. 2014. "Measuring and visualizing energy consumption within software code". In: Visual Languages and Human- Centric Computing (VL/HCC), 2014 IEEE Symposium on, July, 181–182.

Couto, M., Carção, T., Cunha, J., Fernandes, J., Saraiva, J. 2014. "Detecting anomalous energy consumption in android applications". In Pereira, F.M.Q., ed.: Programming Languages - 18th Brazilian Symposium, SBLP 2014, Maceio, Brazil, October 2-3, 2014. Proceedings. Volume 8771 of Lecture Notes in Computer Science., Springer, pp. 77– 91.

Duarte, D., Belo, O., 2017. "Evaluating Query Energy Consumption in Document Stores", In Proceedings of The 1st EAI International Conference on Emerging Technologies for Developing Countries (AfricaTek'2017), Marrakech, Morocco, March 27–29.

GGP, The Greenhouse Gaz Protocol, 2017. [online] Available at: < http://www.ghgprotocol.org/standards/corporate-standard> [Accessed 17 March 2017].

Gonçalves, R., Saraiva, J., Belo, O., 2014. "Defining Energy Consumption Plans for Data Querying Processes", In Proceedings of The 4th IEEE International Conference on Sustainable Computing and Communications (SustainCom 2014), Sydney, Australia, 3-5 December.

Guimarães, M., Saraiva, J., Belo, O., 2016. "Some Heuristic Approaches for Reducing Energy Consumption on Database Systems", In Proceedings of The Eighth International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA'2016), Lisbon, Portugal, June 26 - 30.

Harizopoulos, S., Shah, M.A., Meza, J., Ranganathan, P., 2009. "Energy efficiency: The new holy grail of data management systems research," in CIDR 2009, Fourth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7.

Kumar, R., 2010. Data center power, cooling and space: A worrisome Outlook for the next two years.

Kunjir, M., Birwa, P., Haritsa, J., 2012. "Peak power plays in database engines". In E. A. Rundensteiner, V. Markl, I. Manolescu, S. Amer-Yahia, F. Naumann, and I. Ari, eds.: 15th International Conference on Extending Database Technology, EDBT '12, Berlin, Germany, March 27-30, 2012, Proceedings, ACM, 2012, pp.444–455.

Lang, W., Kandhan, R., Patel, J., 2011. "Rethinking query processing for energy efficiency: Slowing down to win the race". IEEE Data Eng. Bull. 34(1), pp. 12–23.

Lang, W., Patel, J., 2009. "Towards eco-friendly database management systems". In CIDR 2009, Fourth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7.

Liu, K., Pinto, G., Liu, D., 2015. "Data-oriented characterization of application-level energy optimization", in: Proceedings of the 18th International Conference on Fundament al Approaches to Software Engineering, FASE'15.

MySQL, 2017. MySQL Community Edition. [online] Available at: < https://www.mysql.com> [Accessed 17 March 2017].

Neo4J, 2017. Neo4J Community Edition. [online] Available at: < https://neo4j.com> [Accessed 17 March 2017].

Pereira, R., Couto, C., Saraiva, J., Cunha, J., Fernandes, J. 2016. "The influence of the java collection framework on overall energy consumption". In: Proceedings of the 5th International Workshop on Green and Sustainable Software, GREENS '16, (New York, NY, USA), pp. 15–21, ACM.

Rasmussen, N., 2011. Determining total cost of ownership for data center and network room infrastructure. Acecced at: http://www.apcmedia.com/salestools/cmrp-5t9pqg/cmrp-5t9pqg_r4_en.pdf.

Rodríguez et al., 2013. "Analysing power and energy consumption of large join queries in database systems," Industrial Electronics and Applications (ISIEA), 2013 IEEE Symposium on, Kuching, pp. 148-153.

Xu, Z., Tu, Y., Wang, X., 2010. "Exploring power-performance tradeoffs in database systems". In Li, F., Moro, M.M., Ghandeharizadeh, S., Haritsa, J.R., Weikum, G., Carey, M.J., Casati, F., Chang, E.Y., Manolescu, I., Mehrotra, S., Dayal, U., Tsotras, V.J., eds.: Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA, IEEE, pp. 485– 496.

Xu, Z:, Tu, Y., Wang, X., 2012. "PET: reducing Database Energy Cost via Query Optimization," in Proc. VLDB Endow. 5, 12, August, pp. 1954-1957.

Wang, J., Feng, L., Xue, W., Song, Z., 2011. "A Survey on Energy-efficient Data Management," in SIGMOD Rec. 40, 2, September, pp. 17-23.