Journal of Universal Computer Science, vol. 20, no. 10 (2014), 1488-1498
submitted: 30/10/13, accepted: 20/6/14, appeared: 1/10/14

Exploiting the Performance-Energy Tradeoffs for Mobile Database Applications

Puyuan Yang
(University of Science and Technology of China, Hefei, China
Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences
yangpuy@mail.ustc.edu.cn)

Peiquan Jin
(University of Science and Technology of China, Hefei, China
Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences
jpq@ustc.edu.cn)

Lihua Yue
(University of Science and Technology of China, Hefei, China
Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences
llyue@ustc.edu.cn)

Abstract: In recent years, mobile operating systems, represented by android and iOS, have become more and more popular in smartphones. However, the energy issue in smartphones, which refers to the poor battery capacity and management schemes, has been the bottle-neck to further advance the development in this area. Generally, mobile applications have to consider both performance and energy consumption. But unfortunately, few works have been focused on this issue. In this paper, we aim at conducting an experimental study on the performance and energy tradeoffs for mobile applications. In particular, we focus on mobile database applications as they are one of the basic applications on mobile operating systems. In detail, we use the android system as the basic mobile platform, and the TPC-H benchmark as the workload, and build a benchmark platform called TPCdroid to conduct performance and energy measurement. In TPCdroid, we control the mobile database performance by changing the processor frequency. The initial experimental results show that lowering frequency is not always helpful for reducing energy consumption. Moreover, we found that the traditional energy reducing technologies based on X86 and magnetic disks are not suitable for mobile database applications running on ARM and flash memory. Finally, we analyse the relationship between performance and energy consumption under different kinds of workloads, which is done by adjusting the parameters reflecting performance and energy consumption for mobile database applications. The results show that the energy consumption has a dynamic connection with the performance in mobile database applications.

Keywords: Android, Mobile database, Energy consumption, Performance, Energy efficiency
Categories: H.3.4, B.8.2

1 Introduction

With the rapid development of mobile operating systems as well as new hardware technologies, mobile smart devices, represented by android phones, iPhone and Windows phones, are increasingly becoming important assistants in people's daily life. From the viewpoint of end users, more advanced hardware is welcome for all the mobile operating systems. In fact, different kinds of advanced hardware components including GPS sensors, Wifi, and Camera, have been widely integrated in modern smart phones. However, these hardware components bring a much serious overhead on the energy consumption [Pathak, 11; Balasubramanian, 09]. Meanwhile, the current battery technology in smart phones is far behind the speed of hardware development. As a consequence, the poor battery endurance has been the bottleneck of mobile smart devices [Zahid, 11]. Therefore, how to achieve the balance between performance and energy consumption in smart phones has been the focus of both academia and industry.

Some previous works have been conducted for the energy saving issues in different domains such mobile multimedia [Trestian, 12], mobile location-based services [Zhuang, 10], wireless network [Balasubramanian, 09], and so on. Besides, many researchers proposed approaches to improve energy efficiency in computing or communication [Cuervo, 10; Ra, 10]. However, to the best of our knowledge, little effort has been focused on the energy issues for mobile database applications. According to a previous study [Pathak, 12], the I/O operation is one of the most important factor influencing the energy consumption of mobile database applications. Since the performance of mobile database applications is also I/O sensitive, it is necessary to make it clear whether I/O is dominant for both performance and energy consumption in mobile database applications.

In this paper, we aim at making an experimental study on the performance-energy tradeoffs for mobile database applications. The contributions of the paper can be summarized as follows:

(a) We designed a benchmark system called TPCdroid to measure the energy consumption and performance metrics for mobile database applications, which is based on the popular Andriod system and SQLite. The latter is a well-known embedded database management system for smart devices. In TPCdroid, we use the standard TPC-H benchmark as the workload [TPC, 12] (see Section 3).

(b) We performed an experiment-based analysis on the correlations between performance and energy consumption for mobile database applications. Our experimental results show that lowering the processor frequency is not always able to saving power, and there is a non-linear relationship between the frequency and energy. In addition, we found the traditional energy-efficient techniques in database area are not suitable to mobile database applications. As a result, the relationship between performance and energy in mobile database applications is dynamic, i.e., when the performance is fluctuating on a lowering level, it is a positive relation between the energy and the performance. When the performance is fluctuating on a higher level, it is a negative relation between them. The dynamic relationship between the performance and energy disagrees with the relationship of the traditional database [Xu, 10] (see Section 4).

2 Related Work

Recent research works on database energy efficiency were mainly focused on personal computers (PCs) and servers. Some early works showed that [Wang, 11; Tsirogiannis, 10; Li, 11], for PCs and servers, energy consumption was positively related to performance and the performance improvement will cost additional power. Another result was that reducing energy consumption will have to sacrifice performance to some extent [Xu, 10]. Thus, traditional studies came to a rule for database energy efficiency, which said that it is possible to reduce energy consumption with some acceptable performance sacrifice. In some other works [Rodriguez, 11; Lang, 11], techniques for measuring energy consumption in current computer-based environment were proposed from the point view of hardware control, through which people can test the dynamic changing of energy consumption and performance metrics by adjusting some hardware-related parameters. Those techniques motivated this paper, and we also use hardware-adjustment-based measurement on energy and performance in our experiments. Although those previous works proposed different descriptions for energy consumption and performance, their conclusions are very similar, i.e., energy consumption was positively related to performance and energy efficiency can be improved by tuning performance metrics.

While in mobile database applications, there is much difference on software and hardware compared with database applications on PCs or servers. Therefore, mobile database applications usually show different scheme in energy consumption and performance metrics. Typical mobile database applications run on a mobile operating system, e.g., android or iOS, and on an embedded database management system, e.g., SQLite. Generally, mobile operating systems as well as embedded DBMSs are very different from those in traditional applications. Those differences can be summarized as follows:

(a) Different implemental mechanism for database software. Traditional database software, such as PostgreSQL, Oracle, and Mysql, is running as an independent process in the operating system. The applications access the database in the way of process communication. Thus database energy efficiency is controlled by the database process itself. On the other hand, SQLite is not an independent process but a dynamically linked library embedded into applications. If a multi-task operating system like android is executing multiple SQLite-based tasks, there will be several SQLite processes running, which means that the energy efficiency in mobile database applications has to consider the multiple processes environment. Basically, the multi-process implemental mechanism in mobile database applications causes more energy consumption than the single-process way in traditional database applications.

(b) Different hardware used. Traditional database applications run typically on PCs and servers. Most of these applications are built based on X86. However, mobile database applications mainly run on portable devices such as smartphones and tablet PCs. Most of these applications are built on ARM. Besides, in traditional database applications we usually use magnetic disks as data storage, while in mobile database applications flash memory is popularly used. Those different hardware between traditional and mobile database applications will result in different algorithms in measuring and controlling energy consumption and performance, and some previous studies have demonstrated this [Myreen, 09; Soundararajan, 10].

3 System Design of TPCdroid

In this section, we discuss the general architecture of TPCdroid. To the best of our knowledge, there are no tools designed for energy and performance measurement of mobile database applications on android. Meanwhile, there are also no standard benchmark traces for this purpose.

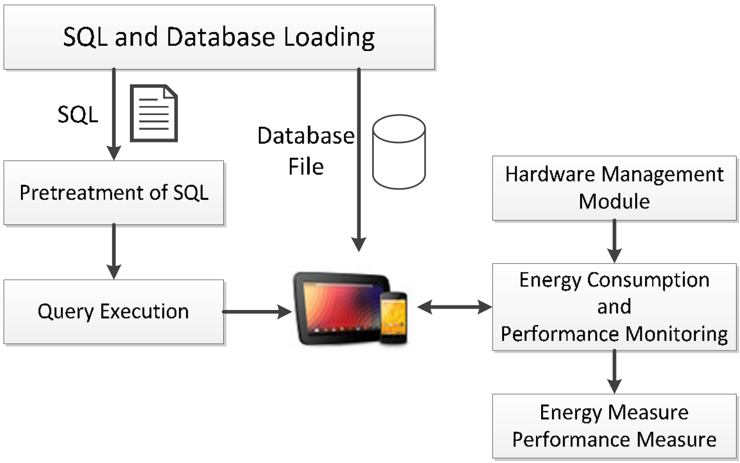

Figure 1: The architecture of TPCdroid

Although the databases in android applications usually do not have large amounts of data, they generally consist of several tables and involve many join operations on multiple tables. For example, the contact application of android consists of several tables, which needs many join operations in daily use. In order to simulate real mobile databases, multiple tables and join queries have be designed in the workload. In our TPCdroid, we use the TPC-H benchmark as the fundamental workload and finally generate test traces for mobile database applications.

Figure 1 shows the general architecture of TPCdroid. As Fig.1 shows, TPCdroid consists of five parts: (1) SQL and database loading module. (2) SQL pre-processing module. (3) Query execution module. (4) Hardware management module. (5) Database power and performance monitoring module. We select the queries in TPC-H which contain multiple-table join operations as our test queries. As a consequence, the following queries in TPC-H, Q3, Q5, Q7, Q8, Q9, Q18, Q1, Q6, and Q14, are chosen as the test workload in TPCdroid. They are numbered from Query 1 to Query 9 in this paper, where Query 1 refers to Q3, Query 2 is Q5, and Query 9 equals Q14.

4 Energy Consumption vs. Performance

TPCdroid runs on the smartphone Samsung galaxy with system CM10(android 4.0). The size of the test database is set to 1GB. In order to measure the energy consumption of mobile database applications under different performance metrics, we execute each query 10 times under three processor frequencies, i.e., 400MHz, 800MHz, and 1GHz.

4.1 Performance Measurement

First, we measure the run time of query 1~6 in TPCdroid, and the results are shown in Fig.2. As Fig.2 shows, the run time reaches the best at 1GHz and the worst at 400MHz. It indicates that the run time of mobile database applications is significantly associated with the processor frequency.

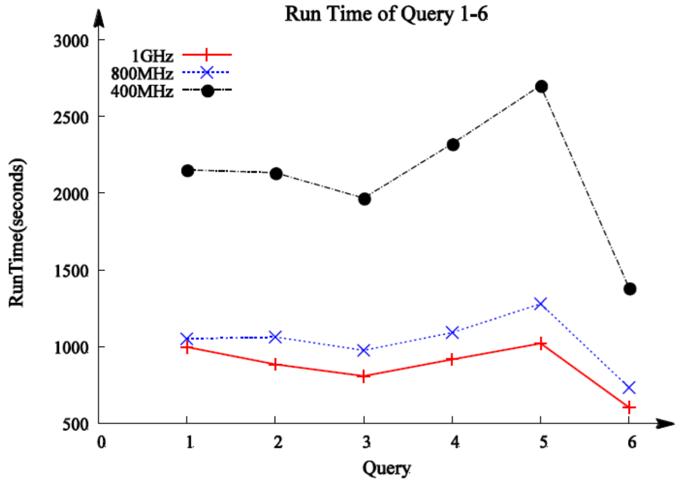

Figure 2: Run time of test queries

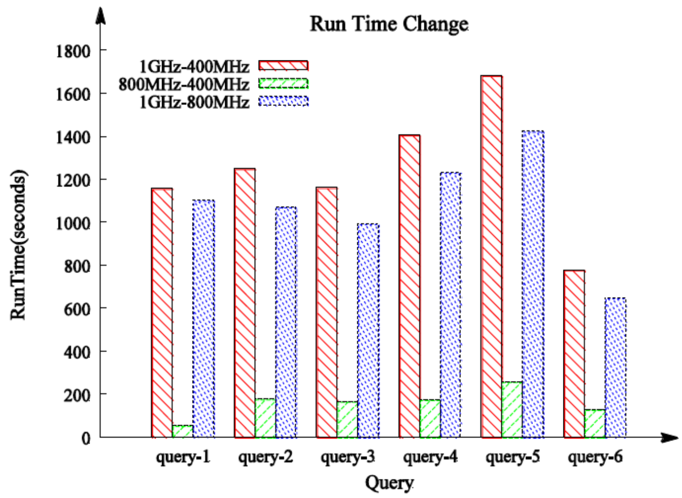

Figure 3: The changing magnitude of the performance

The changing trend of the mobile database performance at different processor frequencies is shown in Fig.3. The performance of 6 queries has the similar changing trends. There is the biggest gap between the performances at 1GHz and 400MHz, which is to be expected. It is noteworthy that the gap between the performances at 1GHz and at 800MHz also almost reaches the biggest. And the gap is obviously small between 800MHz and 400MHz, which tells that the mobile database performance is not proportional to the processor frequency. The performance drops smoothly from 1GHz to 800MHz, and drops dramatically from 800MHz to 400MHz, it suggests that increasing the processor frequency does not always bring the significant improvement, for example from 800MHz to 1GHz. So there may be a relationship of curve shape between the mobile database performance and hardware.

4.2 Energy Consumption

The energy consumption is measured by the tool TPCdroid at different processor frequencies, as shown in Fig.4, and the ordinate axis represents the ratio of consumption to the total energy. As Fig.4 shows, although the processor frequency is the top at 1 GHz, the energy consumption is not most except query 2 and query 4. Moreover, the processor frequency is the lowest at 400MHz, but the energy consumption is not the least, even the most for query 2 and query 4. All the queries have the least energy consumption at 800MHz. The result also suggests there is a curvilinear relationship between energy consumption and processor frequency. All the discussion above proves lowering the processor frequency cannot always be able to save power, the most commonly used method simply lowering the processor frequency on the mobile device, is not effective in mobile database application.

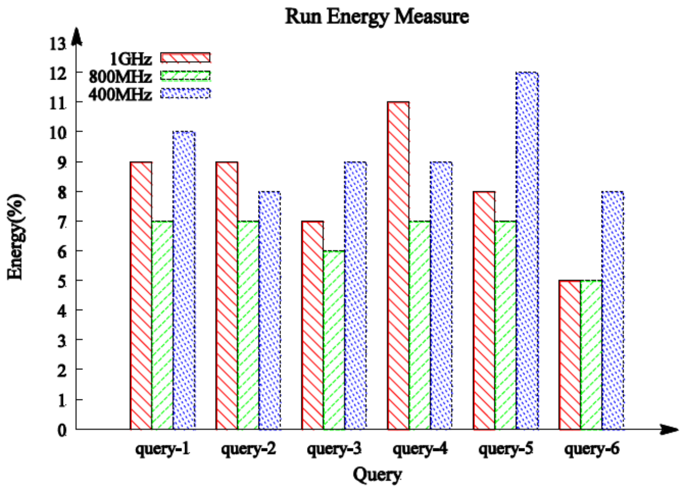

Figure 4: Energy consumption of database

From Fig.4, we can further see that the test data at 1GHz and 400MHz shows too much lowering frequency cannot reduce the energy consumption but increases it. The test data at 800MHz and 400MHz shows increasing frequency may not increase the energy consumption but reduce it. So if the processor frequency is changing in a small range, the energy consumption is reduced with the frequency being reduced, which is fit to the traditional acknowledge. But if the processor frequency is changing in a big range, reducing the frequency may increase the energy consumption and increasing the frequency may reduce the energy consumption, which is new to the traditional view.

4.3 Relationship between Power and Performance

Based on the performance and energy measurement discussed in above sections, we can see that the energy consumption is positively related to the performance and saving power must take performance reduced as cost. Fox example, the performance is worst at 400MHz but the energy consumption is not least, which suggests lowering performance may not bring energy consumption reduced. The energy consumption is least at 800MHz but the performance is not lowest and it is obviously higher than that at 400MHz, being close to that at 1GHz, which is different to the traditional viewpoint, suggesting that improving the performance cannot always increasing the energy consumption. The test data of query 1~6 is selected to show the variation trend of the performance and energy consumption, as shown in Fig.5

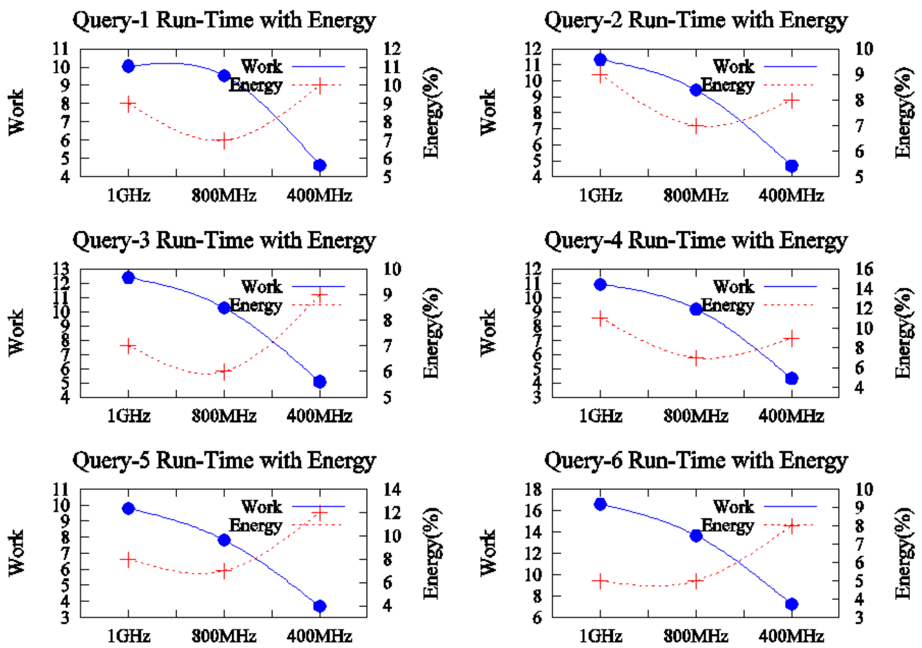

Figure 7: Performance and energy consumption

The line marked as "*work*" represents the performance of the database executing the query. The performance can be scaled by the run time of execution. We do normalization of the test data of run time as formula (1). The test data of 6 queries shows a curve linear relationship between the performance and energy consumption.

Taking the test data at 800MHz as a datum, the energy consumption is positively related to the performance when the performance is changing at a higher level and the energy consumption has a negative correlation to the performance when the performance is changing at a lower level.

Generally, the processor frequency and the amount of computation have great influence on the power and execution time decides the total energy consumption, both of which not only has correlation to the performance but also have great impact on the energy consumption.

$$Work = \left(\frac{run \quad time}{10000} \right)^{-1} \tag{1}$$

As Figure shows, when the database performance is changing at a high level (≥11), the performance may be so close to the upper bound that any performance variation means the amount of calculation is changed, the energy consumption is mainly influenced on by the amount of calculation which means more power. That time, a little performance improvement needs much more calculation, which brings more power and the run time can be just reduced a little. So actually the total energy consumption goes up. When the database performance is changing at a low level (≤10), the performance stand at a so low level that the performance variation not only means the amount of calculation is changed but also makes the run time changed greatly. The influence from the run time changed can goes beyond the influence from the calculation. That is, when the performance is improved, although the power goes up, the greatly reduced run time cuts the total energy consumption down.

The discussion above suggests the mobile database improving work must take the significant improvement as target. If it is hard to make the mobile database great improved, we should sacrifice a little performance for less energy consumption. The improving work, which can bring only a little improvement, may be negative for the power saving.

The formula representing the database energy efficiency is shown as formula (2). The molecular represents the performance and the denominator represents the energy consumption. The quotient of both represents the performance can be obtained per unit of energy consumption which is called energy efficiency.

$$EE = \frac{Work}{Energy \quad Consumption} \tag{2}$$

The performance can be represented by the reciprocal of the run time, as shown in formula (1). By joining the formula (1) and (2), the energy efficiency formula can be represented as formula (3).

$$EE = \frac{(run \quad time / 10000)^{-1}}{Energy \quad Consumption} \tag{3}$$

Based on the performance test and energy consumption test in Section 4.1 and 4.2, calculated by the formula (3), the energy efficiency of the mobile database at 3 processor frequencies is shown in Fig.6. As Fig.6 shows, the energy efficiency is weakest at 400MHz, and both the energy efficiency at 800MHz and 1GHz are close. Both of the Figure and Figure shows, the energy efficiency is almost unchanged when the performance and the energy consumption have a positive correlation. But when the performance and the energy consumption have a negative correlation, the energy efficiency is positively related to the performance which means the energy efficiency is getting weaker with the performance getting down.

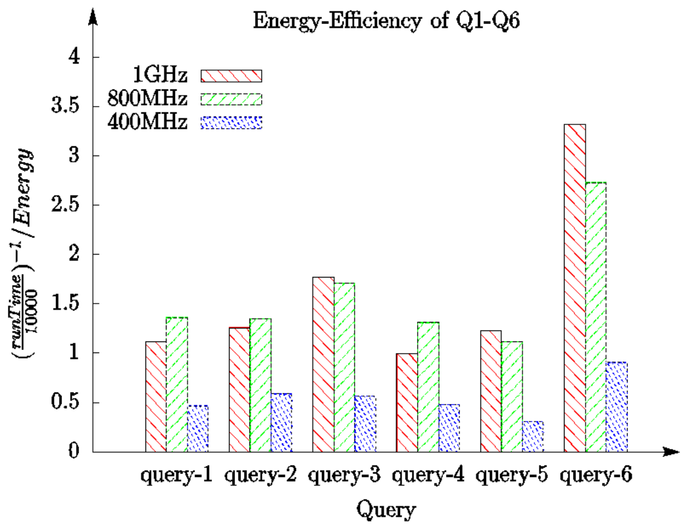

Figure 6: Energy efficiency of the query 1~6

According to the discussion above, we can find that the energy consumption has opponent appearance at different levels of the performance, but the energy efficiency is different from the energy consumption. The improvement, of the performance at a low level, can strengthen the energy efficiency which means the magnitude of improving performance is beyond that of energy consumption increased. This is the feature of the mobile database. But the performance, at a high level being changed, cannot have any correlation to the energy efficiency which has just little change. All this suggests only the improving work being able to bring the significant performance improvement can strengthen the energy efficiency and the work of this kind is valuable for both performance and energy consumption.

5 Conclusion

In this paper, we implemented a tool called TPCdroid to measure the energy consumption and performance metrics for mobile database applications. Based on TPCdroid, we conducted systematic experiments to study the relationship between energy and performance in mobile database applications, and concluded some new results. Firstly, lowering processor frequency to save power is not always right for mobile database applications. Secondly, there is a curvilinear relationship between energy consumption and performance in mobile databases. The energy consumption is positively related to the performance when the performance is at a high level and the energy consumption is negatively related to the performance when the performance is at a low level.

Acknowledgements

This work is supported by the National Science Foundation of China under the grant no. 61379037 and no. 61073039.

References

[Balasubramanian, 09] Balasubramanian N, Balasubramanian A, Venkataramani A. Energy consumption in mobile phones: a measurement study and implications for network applications[C]//Proc. of the 9th ACM SIGCOMM conference on Internet measurement conference. ACM, 2009: 280-293.

[Cuervo, 10] Cuervo E, Balasubramanian A, Cho D, et al. MAUI: making smartphones last longer with code offload[C]//Proc. of the 8th international conference on Mobile systems, applications, and services. ACM, 2010: 49-62.

[Lang, 11] Lang W, Kandhan R, Patel J M. Rethinking Query Processing for Energy Efficiency: Slowing Down to Win the Race[J]. IEEE Data Eng. Bull., 2011, 34(1): 12-23.

[Li, 11] Li Z, Grosu R, Sehgal P, et al. On the energy consumption and performance of systems software[C]//Proc. of the 4th Annual International Conference on Systems and Storage. ACM, 2011: 8.

[Myreen, 09] Myreen M O, Gordon M J C. Verified LISP implementations on ARM, x86 and PowerPC[M]//Theorem Proving in Higher Order Logics. Springer Berlin Heidelberg, 2009: 359-374.

[Pathak, 11] Pathak A, Hu Y C, Zhang M, et al. Fine-grained power modeling for smartphones using system call tracing[C]//Proceedings of the sixth conference on Computer systems. ACM, 2011: 153-168.

[Pathak, 12] Pathak A, Hu Y C, Zhang M. Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof[C]//Proc. of the 7th ACM european conference on Computer Systems. ACM, 2012: 29-42

[Ra, 10] Ra M R, Paek J, Sharma A B, et al. Energy-delay tradeoffs in smartphone applications[C]//Proc. of the 8th international conference on Mobile systems, applications, and services. ACM, 2010: 255-270.

[Rodriguez, 11] Rodriguez Martinez M, Valdivia H, Seguel J, et al. Estimating Power/Energy Consumption in Database Servers[J]. Procedia Computer Science, 2011, 6: 112-117.

[Soundararajan, 10] Soundararajan G, Prabhakaran V, Balakrishnan M, et al. Extending SSD Lifetimes with Disk-Based Write Caches[C]//FAST. 2010: 101-114

[TPC, 12] "TPC-H.": http://www.tpc.org/tpch/.

[Trestian, 12] Trestian R, Moldovan A N, Ormond O, et al. Energy consumption analysis of video streaming to android mobile devices[C]//Proc. Of Network Operations and Management Symposium, IEEE, 2012: 444-452.

[Tsirogiannis, 10] Tsirogiannis D, Harizopoulos S, Shah M A. Analyzing the energy efficiency of a database server[C]//Proc. of SIGMOD. ACM, 2010: 231-242.

[Wang, 11] Wang J, Feng L, Xue W, et al. A survey on energy-efficient data management[J]. ACM SIGMOD Record, 2011, 40(2): 17-23.

[Xu, 10] Xu Z, Tu Y C, Wang X. Exploring power-performance tradeoffs in database systems[C]//Data Engineering (ICDE), 2010 IEEE 26th International Conference on. IEEE, 2010: 485-496.

[Zahid, 11] Zahid I, Ali M A, Nassr R. Android smartphone: Battery saving service[C]//Research and Innovation in Information Systems (ICRIIS), 2011 International Conference on. IEEE, 2011: 1-4.

[Zhuang, 10] Zhuang Z, Kim K H, Singh J P. Improving energy efficiency of location sensing on smartphones[C]//Proc. of the 8th international conference on Mobile systems, applications, and services. ACM, 2010: 315-330