Impacts of data interchange formats on energy consumption and performance in Smartphones

Bruno Gil
DCTI, ISCTE-IUL
Lisboa, Portugal
bfpgl@iscte.pt

Paulo Trezentos
DCTI, ISCTE-IUL
Lisboa, Portugal
paulo.trezentos@iscte.pt

ABSTRACT

In this paper, we analyze the impacts of three formats for data interchange, as regards the management of energy and the cost of processing each of the proposed formats in a mobile device; XML, JSON and Protocol Buffers. The tests focused on two network interfaces, WiFi and 3G. The results of this study contribute to the development of mobile applications that require transfer, independently or not, data for a Web Server, like applications of data backup, monitoring of information or any mobile application that synchronizes data with the Web, having as main objectives the impacts of energy and processing in mobile equipment.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics.

General Terms

Measurement, Performance, Experimentation.

Keywords

Cloud-based mobile applications, XML, JSON, Protocol Buffers, Smartphones, Mobile battery consumption

1. INTRODUCTION

In this new era of communication and information, the Internet has emerged as an important part of social and professional activities, emerging new tools and services that depend on it. However, mobile devices still suffer from limitations in terms of memory, energy and bandwidth, which must be taken into account when developing applications.

The evolution of mobile communication networks allowed the emergence of devices able to offer new services to users. These devices have evolved from basic communications equipment to sophisticated information and communication equipment's.

Mobile devices have similar capabilities to personal computers, with the potential to join us in our daily activities. These types of equipment are constantly connected to the Internet offering us a variety of services, which are strong growth over the past year, adding more and more applications that make use of Web connections for data transfer. This growing number of services and applications available has not been matched to the same extent by an increase in battery capacity of mobile devices, so it became a fundamental rationalization and implementation of good practices in its development. All these services and applications use data interchanges types. These data interchange types have different performances over different network interfaces. For applications that synchronize data between a mobile device and a Web Server, it's fundamental to understand the impact of each data type in battery consumption. Our investigation used three types of data interchange, one binary format, Protocol Buffers, which is a flexible, efficient, automated mechanism for serializing structured data [3] and two text formats, Extensible Markup Language (XML) and JavaScript Object Notation (JSON). XML is a simple, very flexible text format derived from SGML (ISO 8879), and was originally designed to meet the challenges of large-scale electronic publishing [1]. JSON is a lightweight data-interchange format that's easy for humans to read and write and it's easy for machines to parse and generate [2].We have evaluated the importance of data compression for each format type, in terms of performance and energy consumption, on different data sizes.

The central points of analysis were:

- Average time querying and compression data on mobile device. We studied several formats: XML, JSON and Protocol Buffers;
- Average time for parsing each format on server side with and without data compression;
- Average time obtained in the data synchronization with the Web server, in various formats and interfaces for WiFi and 3G, with data compression and no data compression;
- Energy expended with CPU to synchronize data with and without compression on two network interfaces analyzed;
- Energy expended in WiFi interface to synchronize data with and without compression;
- Energy expended in 3G interface to synchronize data with and without compression on two network interfaces analyzed;
- Size, in bytes, of each data volume on each data format

2. THE IMPORTANCE OF BATTERY

Mobile application development requires a special attention to management the battery consumption. Most of the cloud based applications synchronizes data with the Web repeatedly in pre-determinate intervals of time. This technique is known as "polling" and has a major impact in battery life. Some techniques were developed to improve the capacity of battery in mobile devices.

OSDOC 11, July 11, 2011, Lisbon, Portugal.

In [4] is proposed a model that allows synchronization together with other applications that can tolerate delays in synchronization, minimizing the time that the device is in the highest state of consumption, improving efficient in 3G networks, in terms of energy expended. For applications that cannot tolerate delays in their requests, the proposed model taking into account the behavior of users on the Web. Other technique [5] focused on studying the efficiency of battery management in the context of individual use. This type of approach proposes the idea of a management model of resources and processes to learn from users, and on that basis for making management more efficient battery. Studies have identified the peak usage in Smartphone applications in the context of space and time, and this information can be used to allocate energy for such applications in the future.

For applications without delay tolerance or application that synchronizes repeatedly data with the web, especially in slow network interfaces, it's very important to study the impacts of data interchange format in areas like CPU and network energy expended with synchronizations. One of the major factors that create impact on energy expended with network is the amount of data to be synchronized, so it's important analyzed the data sizes of each format for a certain data volume.

3. MEASUREMENT

3.1 Methodology

The methodology used consisted on synchronization, for all data interchange formats, two different volumes of data, on different network interfaces and measure the energy expended, in joules, on CPU processing and amount of network used for each format. To measure performance we have made two types of tests: first we measure the time expended, in milliseconds, for compression data on each data format, and second we measure the time spent on synchronization between the Smartphone and the Web Server, including parsing data on server side. In terms of performance the results obtained express the average time of thirty synchronizations. These tests were made in a platform where information is mostly sent to a Web Server.

3.2 Device and Tools

Our experience are performed using a Smartphone HTC Hero, with a processor Qualcomm ® MSM7200A ™, 528 MHz with a 512 MB ROM and a RAM of 288 MB. We use an application called PowerTutor[1] [9], for measure the impacts on energy, in two network interfaces (WiFi and 3G) and CPU. The mobile application runs on Android 2.1. The environment of test it's a platform for monitoring information of a Smartphone via Web. This platform implements a mobile application, that runs on Android and a Web application for parsing data.

[1] PowerTutor was developed by University of Michigan Ph.D. students Mark Gordon, Lide Zhang and Birjodh Tiwana under the direction of Robert Dick and Zhuoqing Morley Mao at the University of Michigan and Lei Yang at Google. The work was supported in part by Google and the National Science Foundation (under award CNS-0720691), and was done in collaboration with the joint University of Michigan and Northwestern University Empathic Systems Project

3.3 Data Volume

We use two different volumes of data, to measure the impacts of data size on synchronization in each format type.

Table 1. Description of two data volumes used

Type	Information
Volume1 (650 objects)	• SMS
Volume2 (1512 objects)	• Personal contacts • SMS • Applications names • Calls information • Web History • Web Bookmarks

For each volume type we measured the size, in bytes, with and without Gzip compression. The amount of data synchronized has a major impact on performance and energy expended.

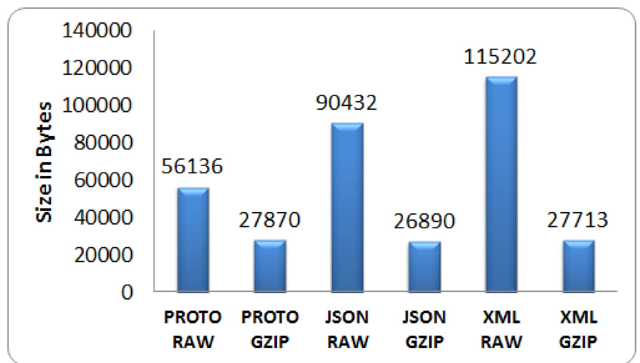

Figure 1 - Data size of volume1 for each format

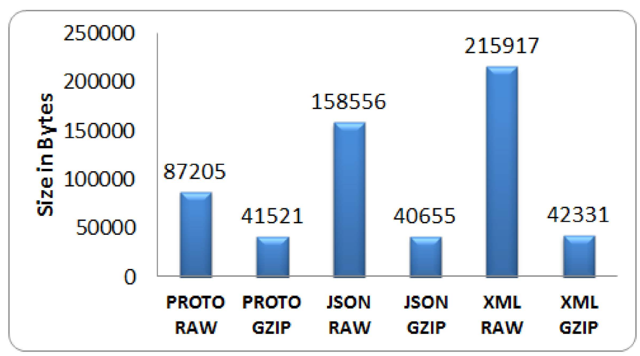

Figure 2 - Data size of volume2 for each format

The data reveals that the binary format, Protocol Buffers, is much more efficient than the JSON and XML format, as regards the size, in bytes, of information synchronized. The data size of binary format is 38% and 48% smaller than the data size of JSON and XML formats, respectively. The difference is somewhat more pronounced with increasing amounts of data, as shown in figure 2, where binary formats increases the efficiency of data size for 40% and 60% to JSON and XML formats, respectively. The advantage of binary format is reversed with data compression, with a marked reduction, for about 66% of initial size for text formats and only a reduction of about 50% for binary format, Protocol Buffers.

3.4 Format data types vs Performance

The performance of each data interchange formats is an important step to building more efficient systems with better performances.

3.4.1 Compression format on mobile device

The testing process consisted of measuring time, in milliseconds, from the beginning of the selection of data in the database of mobile equipment and the end of the data compression for various formats. In this case the tests were conducted on 651 records of text messages (SMS) for 30 times for each format, yielding the following results:

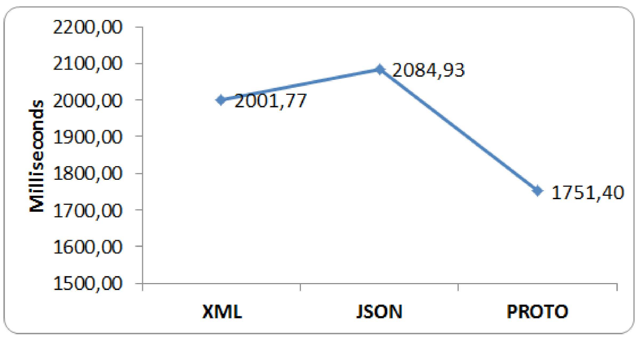

Figure 3 - Average time obtained on selection and compression

The binary format Protocol Buffers had the best average times of performance, 1751.40 milliseconds, and the lower standard deviation of 133. Between the two text formats the results showed no major discrepancies, with however a slightly better performance of the XML format, with 2001.77 milliseconds average and a standard deviation of 166.41. The JSON format got an average time of 2084.93 milliseconds with a standard deviation of 165.32.

3.4.2 Parsing data on server side

The data parsing process has different cost on performance. This result represents the time consumption for parsing and inserts data on Database Management System, in server side for each format type. The tests were conducted on 650 records of text messages (volume1) for 30 times for each format.

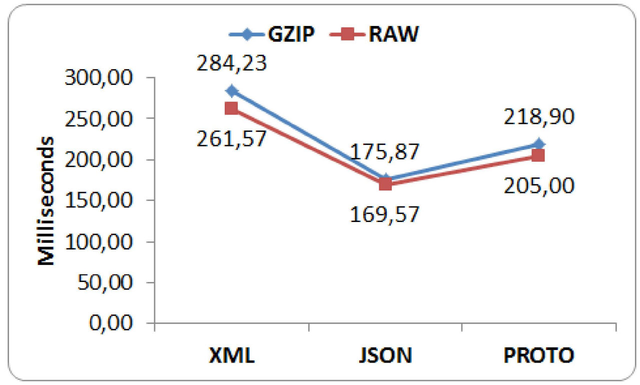

Figure 4 - Average time obtained on parsing data with and without compression (volume1)

In parsing uncompressed data, there is a slight improvement in performance, however, it's not significant given the small amount of data. The cost of compression is greater for XML and Protocol Buffers compared with the JSON format that achieved the best average time with and without compression. It's necessary verify if this loss of performance, when data is compressed, represents a gain in synchronization process.

3.4.3 Synchronization process

Our experiment measures the average time consumption, in milliseconds, for time synchronization expended with server, including data compression on mobile device and data decompression on server side. The tests were conducted on data of volume1 and volume2 for 30 times for each format, yielding the following results:

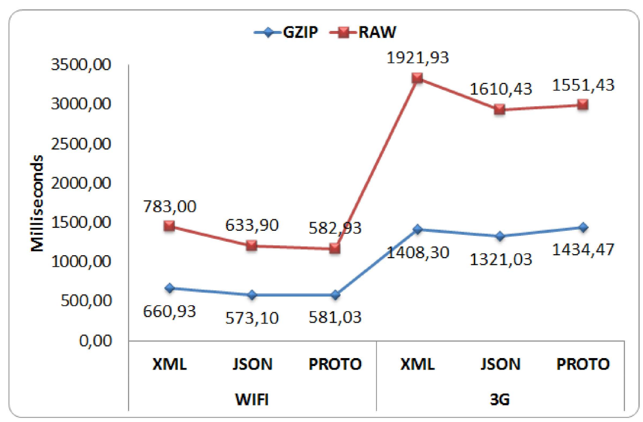

Figure 5 - Time expended for data synchronization in different network interfaces (volume1)

On a fast network interface, like WiFi, data size has a small impact on performance of synchronization comparing to more slower networks, however, we can see that the biggest differences are found in text formats, because the compression reduces by 66% the size of information synchronize, this difference was not so evident in Protocol Buffers format, where the average times obtained differ very little.

In a slower interface, such as 3G, the sizes of transferred data have a greater impact in the average time synchronization. In this case data compression is extremely advantageous in text formats, with significant performance improvements in data synchronization, no revealing the same to the binary format, in part explained by the lower performance obtained with data compression, however there is a gain of time.

The advantage of using data compression, results of analyzing the costs of processing data compression over the benefits obtained in synchronization with the transmission of compressed data. With larger sizes of data we obtained these results:

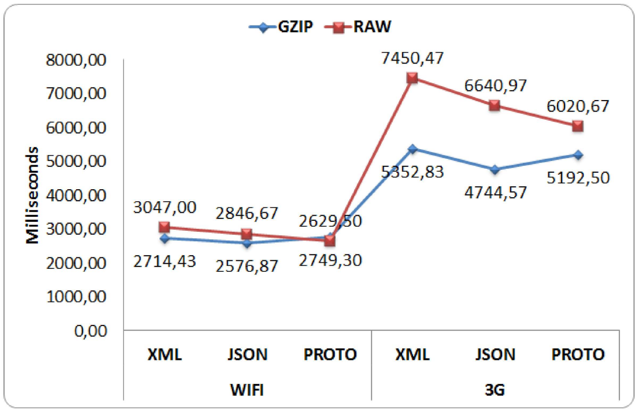

Figure 6 – Time expended for data synchronization in different network interfaces (volume2)

Even with a greater data volume, the use of compression increases performance of synchronization, with the exception of Protocol Buffers on WiFi network interface, which has a performance slightly lower compared to data synchronization without compression. The format Protocol Buffers presents fewer discrepancies between compression and not data compression, particularly in faster network interface, revealing a higher cost on performance when data compress is used. For 3G networks interface, the relationship between size and performance is maximized and is advantageous compressing data in all formats, having a significant improvement in performance of synchronization.

4. ENERGY CONSUMPTION

Other key point when we develop cloud mobile applications it's minimize the impact of synchronizations on battery. In this study we measure these impacts for each format type and volume type (volume1 and volume2) in two different network interface and CPU processing. The results are expressed in joules.

4.1 CPU

For measure energy lost with CPU processing all process of synchronization is recorded. The next two figures represent energy expended, in joules, on CPU processing for two different data sizes.

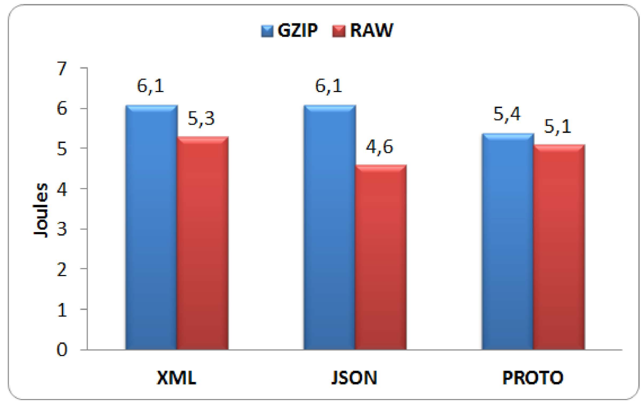

Figure 7 - Energy expended on CPU with data synchronization (volume1)

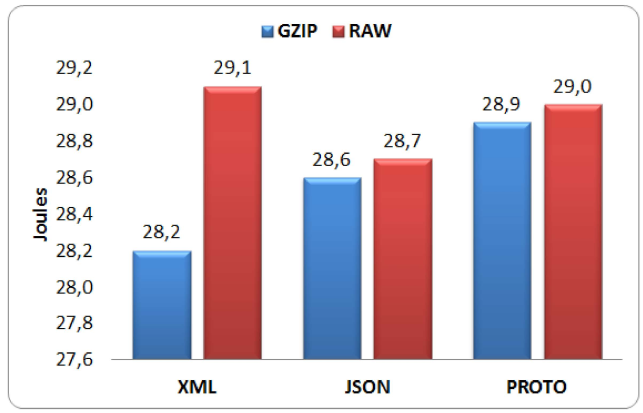

Figure 8 - Energy expended on CPU with data synchronization (volume2)

From this results we can see there is an energy gain in all formats of data studied for no data compression, however, for small amounts of data the impact on battery it's much bigger. It's necessary to assess whether this gain in energy is not lost in the process synchronization.

4.2 Network interfaces – WiFi and 3G

The major energy expended with synchronization occurs when the device uses a network interface. The impact on battery depends on time that the network interface is used. This impact has a close relationship with data size transferred. To understand the advantage of using compression the next two figures shows the energy expended, in joules, for synchronizing two different data sizes, with and without compression, in each network interface.

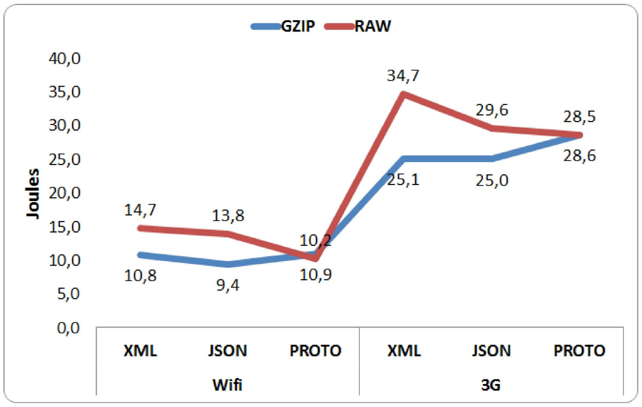

Figure 9 - Energy expended on synchronization for network interfaces (volume1)

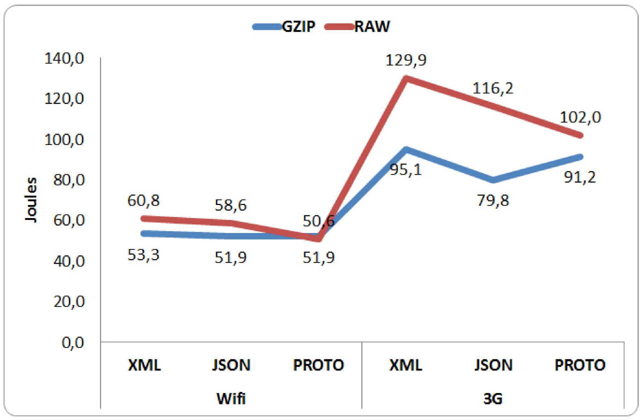

Figure 10 - Energy expended on synchronization (volume2)

On a synchronization process the energy expended in compression of text formats, is largely offset by better performance in power management in data synchronization, whatever the network interface or size used. The binary format, Protocol Buffers, presents greater energy expenditure when there is a data compression, on faster network interfaces. This increased energy expenditure, although not significant, is indicative of the inefficiency of data compression in binary formats for faster networks when the data volume is not significant, although, in a slower network, still there are gains of energy with data compression. The data obtained are proportional to the size of information and have given an insight into the behavior in terms of energy management for other sizes.

5. IMPORTANCE OF COMPRESSION

Without compression, binary formats are much more efficient than text formats, especially in terms of size. This has huge implications on performance because Smartphones are devices with limited capacity in terms of energy and CPU processing. With compression, this advantage is dissipated and text formats takes the lead on data size. The question of the compression costs in terms of battery are shown in this experiment, although, there is a small loss of performance, the gains obtained with compression, especially in energy consumption and performance on network used, is much bigger and permits a faster synchronization with minor impact on energy. Our experiment reveals that on small amounts of data there is still a good impact of using compression, although, in binary formats, compression has to be evaluated, especially for quickly network interfaces.

6. RELATED WORK

Some studies were made, especially for developers at Google and IBM, but in different approaches. In [6], the author tested these three formats in terms of performance and showed the benefits of using compression in text formats. The experiment allowed evaluated performance of synchronization each format type between Web server and a mobile device, where the parsing job was in mobile device side. Our experiment evaluated essentially the data format compression on mobile devices and data parsing each format type on server. However, the results were similar to our experiment, they showed that Protocol Buffers is much more efficient, in terms of time synchronization and data size, than JSON and XML formats for raw data. For compressed data, Protocol Buffers showed the worst results for data size. In other experiment [7], the author tested synchronization performance of these formats: Protocol Buffers, JSON and XML. The results showed that Protocol Buffers is more efficient than any text formats and JSON is more efficient than XML. The results were similar to our experiment despite the different approach that we used in synchronization process. The author showed the impact of compressing on three different network interfaces (WiFi, 3G and EDGE). The results were identical to our experiment and indicated that compression has major impact, in terms of time synchronization, on slow network interfaces like 3G and EDGE but not for a faster network interface, like WiFi. These studies analyzed the performance of interchange format data types and not the energy consumption for parsing and compressing each format, in the synchronization process. Other similar work [8], but in PC environment (client and server), compared two data interchange formats; XML and JSON. The authors conducted a case study that compared the resource utilization and the performance of applications that use the interchange formats. For this different environment, they concluded that JSON is significantly faster than XML and uses fewer resources.

7. CONCLUSIONS

For applications that synchronize data with a Web server repeatedly power management is a key factor. Our investigation shows that between XML, JSON and Protocol Buffers, the better format for data synchronization was the JSON format with compression, because has better efficiency in terms of time synchronization, parsing on server side and better performance in battery management. The worst performance of JSON format was in compression on mobile device, but it was completely absorbed in other positive measures results. The XML format showed weaker results compared to JSON format at all levels, except in battery management with processing cost of CPU when there is data compression and in compression on mobile device. The binary format, Protocol Buffers, proved to be inefficient, compared to JSON text format, when there is data compression, however, it's more efficient than XML format and shows that has broad benefits to all levels when there is no data compression.

This format showed that was more suitable for transferring big data volumes, where the process of data compression is more painful or in situations where data compression is not possible. Because it's a binary format, Protocol Buffers can include binary information such as images, videos or other media types. For raw data, Protocol Buffers have better performance synchronization and better management energy. Data compression has proved that have great importance in text formats, optimizing the speed of synchronization and power management, given its good performance in reducing the size of information to synchronize (about 66% of its original size). The most impressive results are obtained in the slower network interfaces such as 3G interface with not so big discrepancy in faster network interfaces, like WiFi. For both data sizes (volume1 and volume2) the compression always had a positive impact.

In our future work we will extend our experiments to other data interchange formats using multimedia data types.

8. REFERENCES

[1] W3C (2011), Extensible Markup Language. http://www.w3.org/XML/

[2] JSON. json.org. http://www.json.org

[3] Protocol Buffers (2011). http://code.google.com/intl/pt-PT/apis/protocolbuffers/

[4] Balasubramanian, N., Balasubramanian, A. and Venkataramani, A. (2009). Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. Department of Computer Science, University of Massachusetts Amherst.

[5] Rodriguez, V. N., Hui, P., Crowcroft, J. and Rice, A. 2010. Exhausting Battery Statistics. Understanding the energy demands on mobile handsets.University of Cambridge and Deutsche Telekom Laboratories.

[6] Galpin, M. 2010. Using Internet data in Android applications Parse XML, JSON, and protocol buffers data. © Copyright IBM Corporation2010. http://www.ibm.com/developerworks/xml/library/x-dataAndroid/?ca=drs-

[7] Sharkey, J. 2009. Coding for Life -- Battery - Life, That Is. Google I/O 2009 – Developer Conference. http://www.google.com/events/io/2009/sessions/CodingLifeBatteryLife.html

[8] Nurseitov N., Paulson M., Reynolds R, XIzurieta C 2009. Comparison of JSON and XML Data Interchange Formats: A Case Study. Department of Computer Science Montana State University – Bozeman, Montana, 59715, USA

[9] PowerTutor. http://powertutor.org