PERFORMANCE EVALUATION OF JSON AND XML FOR DATA EXCHANGE IN MOBILE SERVICES

Ivar Jørstad
Ubisafe/New Generation Communication, Bjølsengata 15, NO-0468 Oslo, Norway

Elias Bakken
Department of Informatics, University of Oslo, Gaustadalléen 23, NO-0371, Oslo, Norway

Tor Anders Johansen
Department of Informatics, University of Oslo, Gaustadalléen 23, NO-0371, Oslo, Norway

Keywords: Mobile services, JSON, XML, mobile middleware, AJAX.

Abstract: One of the major challenges of developing mobile services is that of data transfer back and forth between the mobile device and the server side application parts. This paper provides a performance analysis and comparison on the use of XML and JSON for data exchange for mobile services. Results obtained from common emulators are provided, as well as results from live operation on state-of-the-art mobile devices.

1 INTRODUCTION

The future success of advanced mobile services depends on the ability to seamlessly integrate mobile application parts with server side components, similar to how Web-based services and other distributed services are realised.

This paper provides a study of two different technologies for data representation when exchanging data between a mobile application and a server; eXtensible Markup Language (XML) and JavaScript Object Notation (JSON).

2 RELATED WORKS

(Arora & Hardy, 2007) introduces an architecture for mobile AJAX with Java ME technology. The paper argues that since many existing products support AJAX in the back-end, it might be sensible to use also AJAX for mobile applications based on JavaME as well.

(Zyp, 2008) provides a performance analysis of Ajax, and shows how to optimize Web-based applications. The analysis points to areas for improvements in traditional Web-applications, but does not consider mobile applications.

(GGGeek, 2007) gives a performance analysis of different JSON parser libraries for PHP, but it does not treat the subject with regards to mobile applications and the client side limitations.

3 PERFORMANCE EVALUATION

3.1 Goals

The primary goal of this performance evaluation is to achieve knowledge on the general differences in performance of using XML or JSON for data exchange in mobile services.

3.2 Hypothesis

As JSON is a less verbose data representation format than XML, it is expected that the parsing from JSON representation into JavaME data structures is less demanding on the processing and memory

Jørstad I., Bakken E. and Anders Johansen T. (2008).
PERFORMANCE EVALUATION OF JSON AND XML FOR DATA EXCHANGE IN MOBILE SERVICES.
In *Proceedings of the International Conference on Wireless Information Networks and Systems*, pages 237-240
DOI: 10.5220/0002025402370240

capabilities of the mobile device, compared to the parsing of XML. Also, due to the different ways of representing data structures (e.g. the characters used to represent boundaries), how hierarchies are constructed etc., it is expected that there is some difference in performance.

It is also expected that some usage scenarios might favour one of the data representation methods, whereas others might favour the other method.

3.3 Test Framework

A JavaME MIDlet was developed which can request data from a remote server. The data retrieved is represented either in XML or JSON. The JSON library used on the client side was JSON-J2ME (http://tavon.org/work/JSON-J2ME), and the XML parser used was the JavaMe port of NanoXML (http://nanoxml.sourceforge.net/orig/kvm.html), which is a DOM-style XML parser (i.e., creates an in-memory representation of the XML document, as opposed to an event-based parser). For future studies, other parsers should be included as well.

The test cases used are:

- *Parsing of many primitive elements*
- *Parsing of large elements*
- *Parsing of deep trees*
- *Parsing of array elements*

3.4 Test Environments

All tests have been performed live on a Nokia N82 running in a MIDP 2.0 environment. The tests have also been run in the Sun Wireless Toolkit Emulator. By running in the emulated environment, the processing capacity and memory of the host device, and the variations of these according to unpredictable behaviour of the device, become closer to irrelevant, and have less influence on the measurement results.

3.5 Results

3.5.1 Primitive Elements

This test consists of parsing JSON and XML documents with an increasing number of primitive elements. The results are seen in Figure 1 and Figure 2. The number of elements ranges from 10 to 5000 with 10 elements interval. All the elements are at the same (second to top) level of the document. As can be seen of the graph, there is no large difference in the time spent processing XML or JSON in this case. However, when run in the emulator with more memory and processing power, it is evident that JSON is performing better than XML. Both for JSON and XML, the function is close to linear, where the rate of increase in processing time is approx. 17.9 for JSON, whereas for XML it is 21.2 (on the device).

3.5.2 Single Element with Increasing Size

This test consists of parsing JSON and XML documents with one element of increasing size. The results can be seen in Figure 3 (on device).

These functions are also linear, where the rate of increase for JSON is close to 2.3, whereas for XML it is close to 6.7.

3.5.3 Complex Structure/Deep Tree

This test consists of parsing JSON and XML documents of increasing complexity. The complexity is introduced by creating a tree of primitive elements. The results can be seen in Figure 4. The function is in this case exponential for XML, whereas it is linear for JSON. Running the same case in the emulator, JSON (with rate of increase in processing time at 0.37) is not visible in the same graph as XML.

3.5.4 Array of Elements

The results of this test are seen in Figure 5 (device). The graphs show that in this case, JSON also scales much better with increasing size of the data structure (array size).

3.5.5 Summary of Results and Discussion

In most of the cases used for this performance study, JSON performs better than XML. Only in the case with an increasing number of primitive elements is XML close to the JSON performance, and in the emulator with “unlimited” resources available JSON still performs better.

The XML parser used in this analysis is a non-validating one. It is reasonable to expect that the difference in performance would be even larger if JSON had been compared to a validating parser. In this situation XML does not have any clear advantages compared to JSON; both can be used to represent the same type of data structures.

There are other selection criteria than performance when selecting a data exchange format. Readability is one. However, XML is marginally more readable than JSON. There exists fewer

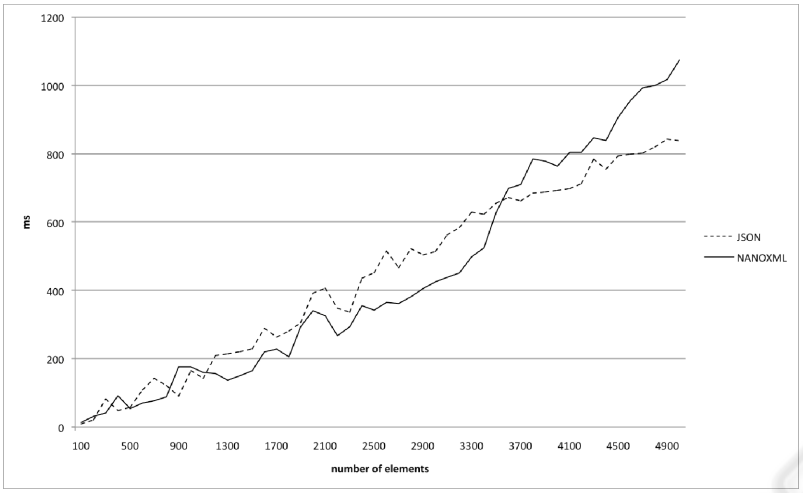

Figure 1: Increasing number of primitive elements on device.

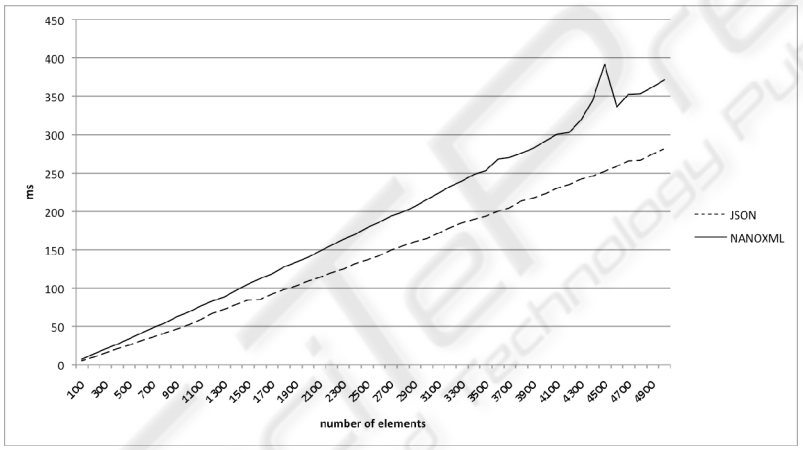

Figure 2: 100-5000 primitive elements in emulator.

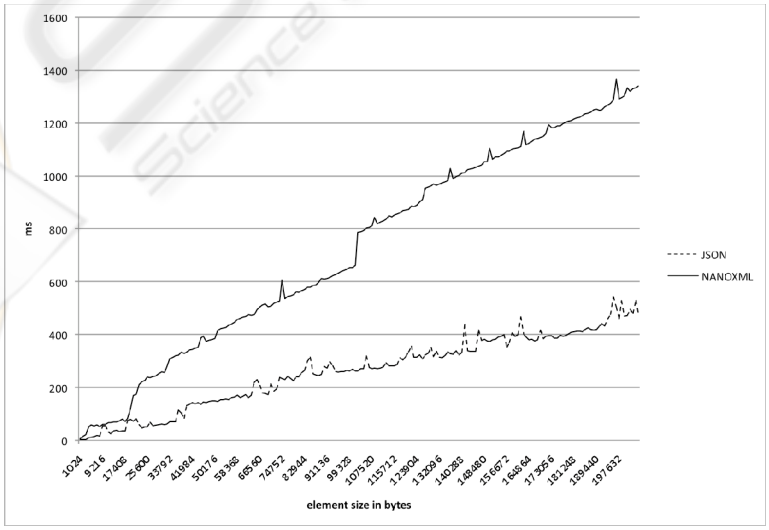

Figure 3: Element size from 1024 bytes to 200 kilobytes on mobile device.

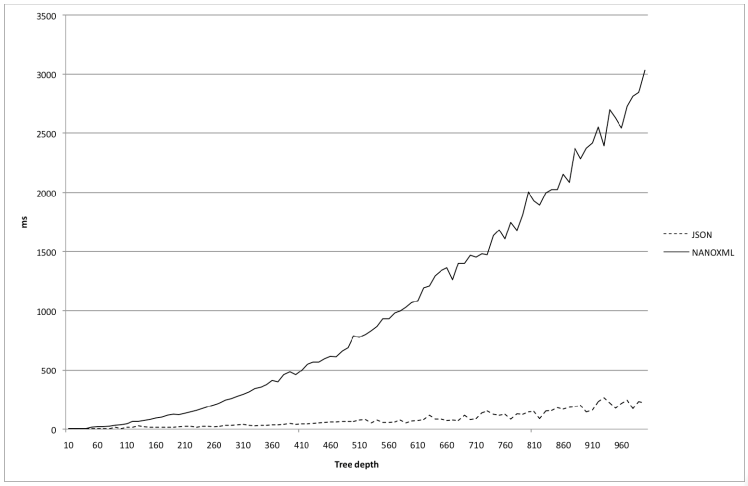

Figure 4: Increasing the tree depth.

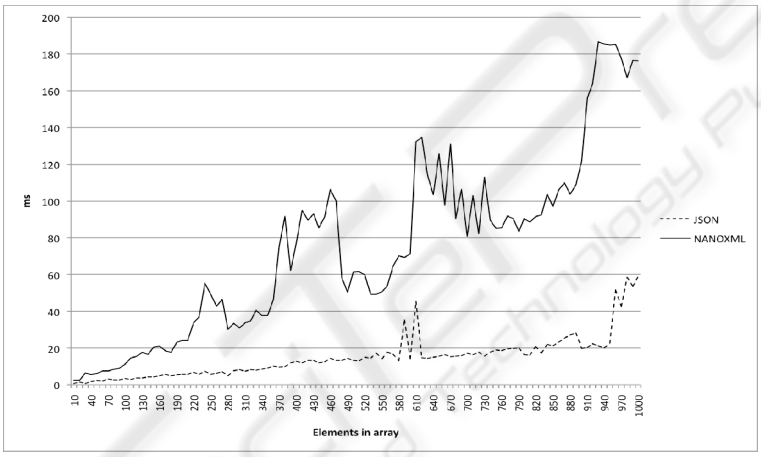

Figure 5: Increasing the size of an array from 10 to 1000 elements on the device.

libraries for JSON than for XML, and depending on the programming language and platform used, this can dictate the final selection.

4 FUTURE WORK

This study will be extended to cover additional parsers, native parsers, as well as other changes in the test input data. It should be interesting to see how attributes on the XML elements influence the results.

5 CONCLUSIONS

This paper has provided a study of how JSON and XML parsers differ in performance on mobile devices, in particular for services built on the JavaME platforms. According to different characteristics of the data transferred between peers in the distributed system, it is possible to select the better solution for each situation; there is no standard answer to which one is the better one, and the developer should weigh the performance shown by this paper against other requirements as well.

REFERENCES

Arora, A., Hardy, V., 2007. Mobile Ajax for Java ME Technology, World Wide Web Consortium (W3C)

GGGeek, 2007. A completely fair and balanced comparison of php json libraries, http://gggeek.altervista.org/sw/article_20061113.html

Zyp, K. W., 2008. Ajax performance analysis, IBM developerWorks, April 24, 2008