COMPARISION BETWEEN JSON AND XML IN APPLICATIONS ON AJAX

Boci Lin
Transportation Management College
Dalian Maritime University
Dalian, China
danshuiboci@126.com

Yan Chen
Transportation Management College
Dalian Maritime University
Dalian, China

Xu Chen
Shangqiu Electricity Bureaut
Henan, China

Yingying Yu
Transportation Management College
Dalian Maritime University
Dalian, China

***Abstract*—As the core technology of Web 2.0, Ajax has caught more and more attention. Xml, as the traditional data load format, needs to be resolved by DOM (Document Object Model) both in client-side and server-side, which wastes system resource and makes a great reduction of user-friendliness. In this paper, a light-weightdata-interchanging format ——JSON (JavaScript Object Notation) will be introduced, which provides a higher level of flexibility and efficiency. We make a comparison between JSON and XML through expriment, then use JSON as data-transfering format in an actual project. Results show that JSON is more suitable as a data-loading tool for Ajax applications**

Keywords- Ajax; XML; JSON; J2EE

I. INTRODUCTION

With the advent of the Web 2.0, the user-friendliness has become an important part of the construction of a modern information system, which can not be ignored. In the era of Rich Client, the traditional network system can not meet the needs of increasingly demanding user experience. In this context, a lot of excellent rich client technology came into being, the most concern was undoubtedly the Ajax (Asynchronous JavaScript and XML) technology. AJAX is not a new programming language, but a new way to use existing standards, such as JavaScript, XML, DHTML and DOM, It sends requests to server-side according to the specific action of client-side, and then process the data returning from server by client-side scripting JavaScript, which greatly reduces workload of server side, pressure on server side and client-side's waiting time, and enhances the user experience.

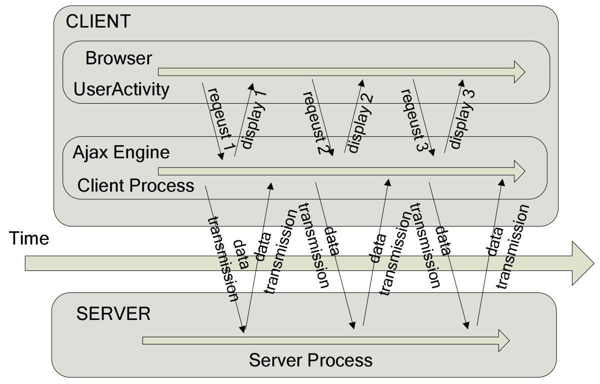

Figure 1. Ajax work flow chart

As traditional Ajax data-transfering format, there is no doubt that XML has the advantages of standardization and scalability. However, due to its own characteristics,files in both client and server need to be complicatedly parsed with the help of DOM, SAX or some other technologies,which is especially fatal in the case of large amount of data and lack of PC's terminal performance and accordingly leads to a big discount of good user experience bringing by Ajax. This article will introduce a new data format JSON, compare it with XML and describes its application in practical projects.

II. INTRODUCTION TO XML AND JSON

A. XML

XML is an abbreviation for Extensible Markup Language, a simple, very flexible text format derived from SGML. XML started in 1995, at the same year, proposed to W3C, and in February 1998 it was recognized as a W3C standards (XML 1.0). XML is a cross-platform, standardized language, which has already become the accepted industry standards. XML uses elements and attributes to describe data. Actually XML organized data as a tree structure. In the data transfer process, XML has always maintained hierarchy relations of the elements, which show the subordinate relationship between the elements clearly and easily.

Next, a simple example will be given to show how XML describes the data elements. Here will send a list of classes which give a description of station. This class has only two attributes: ID and NAME. XML describes these data as the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<stations>
        <station>
                    <ID>1</ID>
                    <NAME>Beijing</NAME>
        </station>
        <station>
                    <ID>2</ID>
                    <NAME>Tianjin</NAME>
        </station>
        ……
</stations>
```

In a tree structure that can more clearly demonstrate its data structures, as shown below:

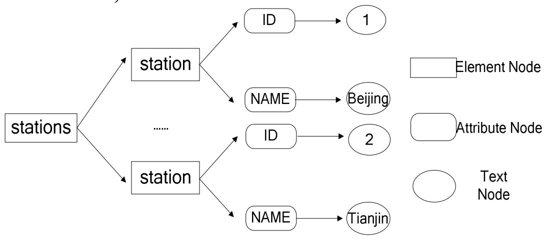

Figure 2. XML data graph in tree structure

B. *JSON*

JSON is a lightweight data interchange format, format-friendly, easy to read and write, more importantly, as opposed to XML, it's easier for generating and parsing, which enables Ajax applications can do more smooth data presentation. It is based on a subset of JavaScript (Programming Language, Standard ECMA-262 3Rd Edition December, 1999), using the text format that completely independent from programming language, but at the same time it also use the habits like C family (including C, C++, C#, Perl, Python, etc.). These properties make JSON quickly become a popular data interchange format.

JSON describes data object mainly in two forms: arrays and objects. An array in JSON is an ordered set of values.It starts with a "[" and end with a "]", and the values in array are separated by ",". Object is a disorder set of name/value pairs, its form is similar to the obeject map in Java. An Object in JSON starts with a "{" and ends with a "}", every name is follwing by a ":", and every name/value pair is separated by a ",". The following illustrations describe these two data structures.

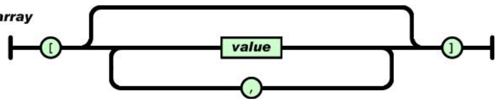

Figure 3. Array in JSON

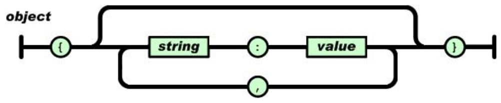

Figure 4. Object in JSON

Reference to the example used above, we describe the data by JSON as the follwing code:

[{ "ID" : "1", "NAME": "Beijing" } , { "ID" : "2", "NAME": "Tianjin" } ……]

III. The Efficiency Comparison Between XML and JSON

After the introduction above, we have a preliminary understanding how XML and JSON describe the data. Next, we will do some experiment to compare these two data interchange format.

A. *Comparison of the data transfer efficiency*

In order to compare the data transmission efficiency of JSON and XML, we set up a test environment, respectively, using these two data interchange format to transfer a same set of data from server-side, and gradually increase the amount of data to observe the changes of record delivery time, and thus indirectly to compare data transmission efficiency of them.

Taking into account the length of this article, we will not go into details about the test environment here. Due to network stability and other factors, the experimental data will have some fluctuations. To make the experimental data as accurate as possible, we take the mean value through 10 tests. The test results are shown in the table below.

TABLE I. Comparison of Data Transmission Time Between XML and JSON

Data Amount	**Time consume (ms)**	
	Xml	***JSON***
100	23.16	16.07
200	24.45	16.65
500	40.14	27.55
800	44.36	33.08
1000	69.35	47.25
2000	120.15	78.45

The test results shows that opposed to XML, JSON has higher data transmission efficiency, and with the amount of data to futher increase this gap will become increasingly significant. It is beacause that as data transmission format in plain text, they have special syntax rules and established format requirements. In JSON, we take simple symbols such as “,”, “{”, “[”as the dividing line for datas. However, in XML, these separators appear more “bloated”, like the example above <stations></stations>. This has resulted in the same amount of data transfer, the redundant load data of XML is higher than JSON, and this gap will increase with the complexity of data structures and the expansion for amount of data.

B. *The comparision of client efficiency*

The data passed over from the server side by JSON and XML, should be deserialize in the client. XML is based on the DOM tree, and the relationship between its parent and child elements of the tree structure needs to be taken into consideration in its deserialization. It’ s difficult for the analytical work of the client. In contrast, the JSON deserialization relatively much more convenient, the client －side only need to use the JavaScript function called eval () to change the data from server side with JSON into JavaScript obeject.

The following two groups of test code in the client to contrast the same amount of data in these two data-transmision formats’ deserialization time, the following two sets is the testing JavaScript code:

```
Testing code for XML:
var xmlDom=request. responseXML;
var allStation= xmlDom.getElementsByTagNam
e("station");
for(var j=1;j<6;j++){
    var startT ime=new Date().getT ime();
    for(var i=0;i< allStation.length*j; i++){
        var station=allStation[i];
        var ID= station.getE lem entsByT agNam
e("ID")[0].firstChild.data;
        var NAME= station.getE
lementsByTagName("NAME")[0].firstChild.data;
        }
      var endTime=new Date().getT ime();
      var spendTime= endTime - startT ime;
}
```

```
Testing code for JSON:
var jsondoc=request. responseText;
var allStation = jsondoc.getElementsByTagNam
e("station");
for(var j=1;j<6;j++){
    var startT ime=new Date().getT ime();
    for(var i=0;i< allStation.length*j; i++){
        var allStation =eval(‘(‘+request. responseText+’)’);
        var NAME= allStation.station.nam e;
        var ID= allStation.station..ID;
    }
    var endTime=new Date().getT ime();
    var spendTime= endTime - startT ime;
}
```

Test results are shown as below:

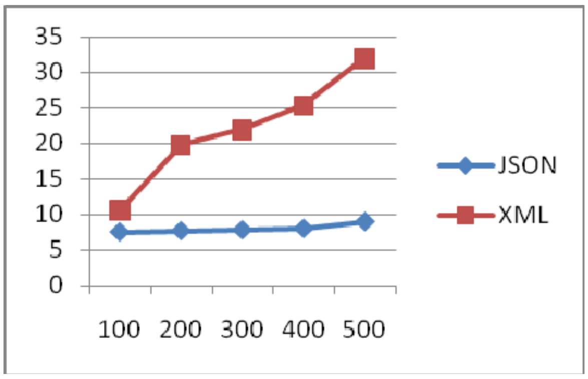

Figure 5. Comparison chart of the test results for Client resolver efficiency

From the test results, we can see JSON in the client's efficiency is much higher than the XML, and with the amount of data increases, the JSON’s deserialization time-consuming has no significant increase.However, XML with the amount of data increases, the client parse time appears to grow significantly.

IV. JSON in Actual Application

From the comparision above, we can see the advantages of JSON in Ajax Application, so we selected JSON as data-transmision format in actual project. Taking into account the length of this paper, here only describes a simple application.

The project uses the traditional J2EE framework system: WSH (Webwork + Spring + the Hibernate). In order to call the Ajax functionality in a convenient and efficient way, we use the Ajax framework: Dojo in client. The client and server-side independent of each other, each perform their respective functions, the overall project structure are shown as below.

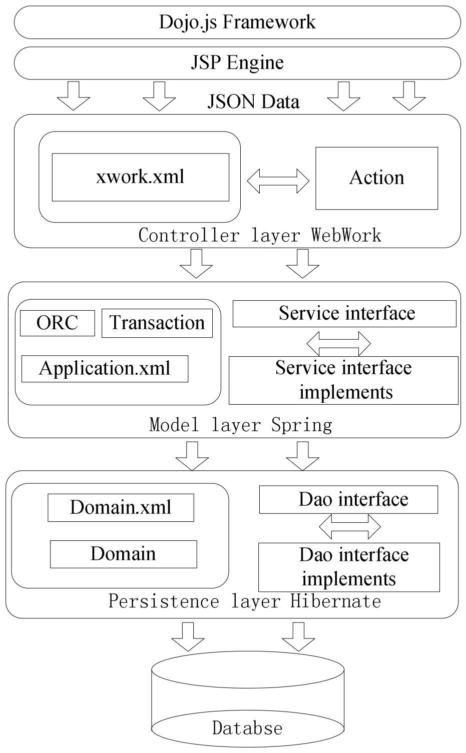

Figure 6. The Structure graph of this project

We put the JSON-parsing work on the client, use json.jsp to resolve the queue data obtained, the code is shown below:

```
<%@ page contentType="text/html;charset=utf-8"%>
……
/*
jsonData3={
     identifier: 'id',
     label: 'name',
     items: [
          <ww:iterator value="harborList" status="ture">

               {id:'<ww:property
       value="id"/>',name:'<ww:property
       value="name"/>'
      </ww:iterator>
     ]
}
*/
```

The code for sending Ajax request is shown below:

```
function refresh() {
    var
    appID=document.getElementsByName("proForHo
    meHarbor")[0].value;
    dojo.xhrGet({
       url:
          "<%=request.getContextPath()%>/shiprepair/g
          etShipRepairRefresh.action?appID="+appID,
       handleAs: "json-comment-filtered",
       load: function(jsonData)
       {
          refreshSelect(jsonData);
       },
       error: function(response)
       {
        alert("Abnormal data transfer");
       }
     });
  }
```

The practice shows that the JSON is very suitable for Ajax applications.

V. CONCLUSION

Through the introduction of JSON above, as well as its application in actual projects, we can see JSON, as a lightweight data carrier, has characteristics of small space occupancy and fast transmission speed, which give it a unique advantage in Ajax applications. Meanwhile, its corresponding variety of development kit and its concise and readable syntax features, also provide a great convenience for application developers. There is still much room of JSON on web services and data aspects for development. Of course, in comparison with XML, JSON still has inadequacies in standardization, security and other aspects at present. But with the widely spreading of JSON, I believe that these deficiencies will be gradually improved in the future.

ACKNOWLEDGMENT

This work was supported by the supported by the Fundamental Research Funds for the Central Universities (Grant No. 2011QN157) and the Major scientific research training project of Dalian Maritime University (Grant No. 2011ZD007).

REFERENCES

[1] Paul J.Deitel, Harvey M.Deitel.Ajax, Rich Internet Applications, and Web Development for Programmers[M].Beijing: Publishing House of Electronics Industry, 2010.5.

[2] Can Cui, Hong Ni,Optimization Simulation of XML Performance Based on JSON. Communication technology, 2009, Vol.42：108-110, 114 (In Chinese).

[3] Qiang Wang, Wei Wang and Xiao Mei Zhang. AJAX-RMI Plug-in Based on JSON and IoC. Computer Engineering, 2010,36(7): 52-54.. (In Chinese).

[4] Li Tan, Zhong Yuan Yang, Data Response Optimization of Ajax. Computer Engineering, Vol 22, pp.108-110, 2006(In Chinese).

[5] Feng Yang, Jian-bo Xu, Research on the Performance Improvement of the AJAX technology. Computer Engineering and Science,

[6] json.Introducing JSON[EL/OL].http://www.json.org, 2006