Yukun Ma
Week #7: Data Exchange
Page #1

Problem Statement

Data exchange is used when two programs or systems need to share some data. More formally, data exchange is the procedure when data is transforming into some schema from another schema.[1]

In this paper, the data exchange problems will be described and then the role that XML as well as JSON plays in these problems will be evaluated.

References

[1] Wikipedia. Data exchange - wikipedia. `https://en.wikipedia.org/wiki/Data_exchange`. Accessed April 12, 2018.

[2] Wikipedia. Http compression. `https://en.wikipedia.org/wiki/HTTP_compression`. Accessed April 12, 2018.

Overview of Data Exchange

When the data need tranforming from a kind of schema to another schema, usually data exchange will be used.

Let me take the data exchange in a web app as an example. Let's say a certain web app has a front-end page, it could be a page for users to log in or a page for users to create accounts. When users filled in the form out and then clicked the "submit" button, the front-end page should transfer the data that users just typed in to the backend to validate the account information or other information. But how to transfer the data? In AJAX[1], JSON and XML are usually used as a exchange format.

Usually people use an exchange format or interchange format so that they can translate the data in the source schema to data in this form as a intermediate step. And XML and JSON described above can be used in this way.

Critical Thinking

Table 1: The Differences Between XML and JSON

	XML	JSON
Encoding	slower	quicker
Decoding	hard	much easier
Weight	heavier	lighter
Popularity	more popular	popular in AJAX

[1] Asynchronous JavaScript and XML

Question

In the application of web services, if we are transferring a huge amount of XML ,JSON or data in other format, how to speed up the process of the data exchange.

Method

Describe how you are going to answer your own question stated above.

Analysis and Discussion

As a student developed writes sereral web apps, I think when we are dealing with a huge amount of data, the first thing we should think about is: Do we really need so huge amount of data being transferred in a web apps. We can use several techniques to reduce the amount of data in the data exchange. For example, we can make the frontend page to request data only when the current user have a change to see it. Let's say that a user is reading a very long page of article, we can request the currently visible passages of this article in this frontend page from the backend server. We can also divide this article to different pages.

In addition to considering application design issues, there are actually other technologies that can help us reduce the amount of data transferred, thus increasing the speed of data exchange. HTTP compression is a method that web servers and clients can use to compress the HTTP requests and responses to improve the speed of data exchange.[2] When a server sends out a HTTP response (the HTTP response may contain HTML or JSON) to the client, it can use some HTTP compression method to compress the response to make the interchange data smaller in size. For example, gzip is a broadly used compression algorithm that can be used in HTTP compression. No matter the response is a JSON or not, the server just need to compress the content of the response and add "Accept-Encoding: gzip, deflate" to the header of the response. Then the client (browser) will automatically decompress the HTTP content and parse it in a way that the content should be parsed.

In summary, we should first consider the design of web services. If the effect is not good, we can consider using HTTP compression technology to improve the efficiency of data exchange.