СРАВНИТЕЛЬНЫЙ АНАЛИЗ ФОРМАТОВ СЕРИАЛИЗАЦИИ И ПЕРЕДАЧИ ДАННЫХ JSON, XML, CBOR И GPB

COMPARATIVE ANALYSIS OF JSON, XML, CBOR AND GPB SERIALIZATION AND DATA TRANSFER FORMATS



УДК 004.62

Шульман Виталий Дмитриевич, Студент 2 курса магистратуры, МГТУ им. Баумана, e-mail: vitalian42@mail.ru

Шабанов Василий Васильевич, Бакалавр, выпускник МГТУ им. Баумана, e-mail: bravoxmail@gmail.com

Сухов Павел Александрович, Бакалавр, выпускник МГТУ им. Баумана, e-mail: suhov-p@bk.ru

Чунихин Антон Олегович, Бакалавр, выпускник МГТУ им. Баумана, e-mail: anton_chunihin@mail.ru

Shulman Vitaly, 2nd year Master's student, Bauman Moscow State Technical University

Shabanov Vasily, Bachelor's degree, graduate of Bauman Moscow State Technical University

Sukhov Pavel, Bachelor's degree, graduate of Bauman Moscow State Technical University

Chunikhin Anton, Bachelor's degree, graduate of Bauman Moscow State Technical University

Аннотация

Статья посвящена четырём форматам сериализации данных и их сравнительному анализу. Форматы рассматриваются в контексте их использования для передачи данных. В процессе анализа использовались как объективные количественные характеристики, так и субъективные оценочные. Итоговое сравнение приведено в виде сводной таблицы. Сделано заключение о целесообразности использования каждого из форматов.

Annotation

The article is devoted to four formats of data serialization and their comparative analysis. Formats are considered in the context of their use for data transmission. In the process of the analysis, both objective quantitative characteristics and subjective evaluative ones were used. The final comparison is given in the form of a summary table. The conclusion is made about the expediency of using each of the formats.

Ключевые слова: формат сериализации, производительность, текстовый формат, бинарный формат, JSON, XML, CBOR, GPB

Keywords: serialization format, performance, text format, binary format, JSON, XML, CBOR, GPB

Форматы сериализации данных JSON, XML, CBOR и GPB используются для обмена данными в информационных системах, чаще всего в веб-приложениях с клиент-серверной (в частности, REST) архитектурой. Согласно REST, форма хранения данных должна быть независима от формата, в котором осуществляется их передача (рис. 1), что и обеспечивается рассматриваемыми форматами сериализации [1].

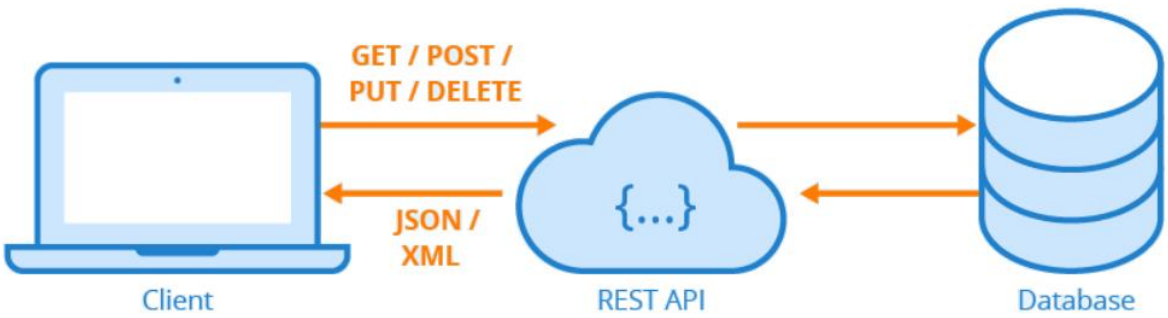

Рисунок 1 – Концепция REST

При проектировании и разработке информационных систем, где подразумевается активный обмен данными между компонентами системы по сети (как локальной, так и сети Интернет), выбор формата серилизации является непростой и одной из первостепенных задач. Непростой по причине большого множества вариантов. Первостепенной по причине того, что данный выбор оказывает существенное влияет на эффективность функционирования всей системы в будущем [2].

При выборе формата сериализации можно руководствоваться 2 типами характеристик:

– объективные (эффективность сжатия, скорость сериализации и т. д.);

– субъективные (удобство чтения, распространённость и т. д.).

Объективные характеристики представляют из себя количественные метрики, которые могут быть взяты из открытых источников, подсчитаны аналитически или получены экспериментально. Субъективные характеристики не могут быть однозначно выражены в виде некой количественной оценки, что делает проблематичным их использование при оценке. Несмотря на явное превосходство объективных характеристик над субъективными, на практике в анализе применяются оба типа [2].

К объективным характеристикам можно отнести:

– эффективность сжатия (итоговый процент занимаемого объема по отношению к объему до сериализации);

– скорость сериализации (время, требуемое для преобразования данных из исходного формата в формат сериализации);

– скорость десериализации (время, требуемое для преобразования данных из формата сериализации в формат структуры данных).

Объективные характеристики оказывают влияние на работу системы непосредственно во время её функционирования. Их значимость легко можно проследить по рисунку 2, описывающему суть применения сериализации.

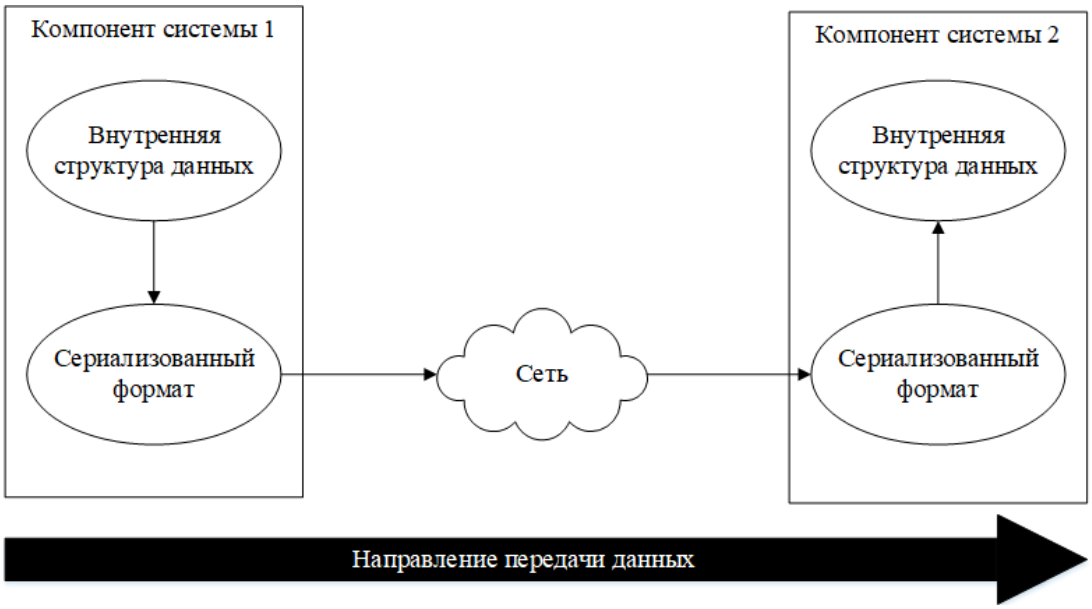

Рисунок 2 – Принцип сериализации

Субъективных характеристик выделяют значительно больше. К основным субъективным характеристикам относят:

– удобство чтения;

– распространённость;

– простота использования;

– динамика развития;

– поддерживаемость;

– наличие стандарта;

– качество документации.

Все форматы сериализации данных можно разделить на 2 типа: текстовые (JSON и XML) и бинарные (CBOR и GPB). В текстовых форматах сериализации данные представляются в человекочитаемом виде, в случае же бинарного формата передаваемые данные представляют собой последовательность байтов.

JSON (JavaScript Object Notation) – текстовый формат сериализации данных, созданный в 2001 году и основанный на подмножестве языка программирования JavaScript, с помощью которого создают интерактивность на веб-страницах [3]. Несмотря на то, что JSON практически полностью повторяет синтаксис описания данных класса в языке программирования JS, этот формат поддерживается во всех современных высокоуровневых языках программирования.

XML (eXtensible Markup Language) – текстовый документоориентированный формат сериализации данных, созданный в 1998 году и основанный на подмножестве метаязыка SGML [3]. Язык именуется расширенным, т. к. не обладает фиксированной разметкой (схемой). Разработчик волен самостоятельно определять структуру документа XML, руководствуясь особенностями предметной области и синтаксическими правилами XML [2].

CBOR (Concise Binary Object Representation) – бинарный JSON-подобный формат сериализации данных, созданный в 2013 году [4]. Был спроектирован таким образом, чтобы агрегировать данные в максимально компактном виде, но при этом обладать достаточно простым кодом реализации. В формате поддерживается как обратная, так и прямая совместимость.

GPB (Google Protocol Buffers) – бинарный формат сериализации данных, созданный в 2001 году компанией Google. Является кросс-языковым, кросс-платформенным XML-подобным форматом, но более экономным по памяти, производительным и обладающим простым кодом реализации [5].

Даты создания описываемых форматов значительно разнятся (рис. 3).

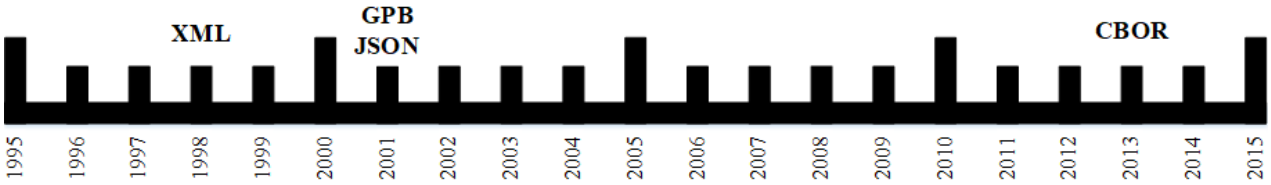

Рисунок 3 - Хронология

CBOR достаточно молодой формат по сравнению с JSON, XML и GPB, что, по всей видимости, сказывается на количестве данных о его распространенности. Для оценки популярности рассматриваемых форматов использовалась статистика ресурса Stack Overflow за период с 2009 по 2021 год (рис. 4) [6].

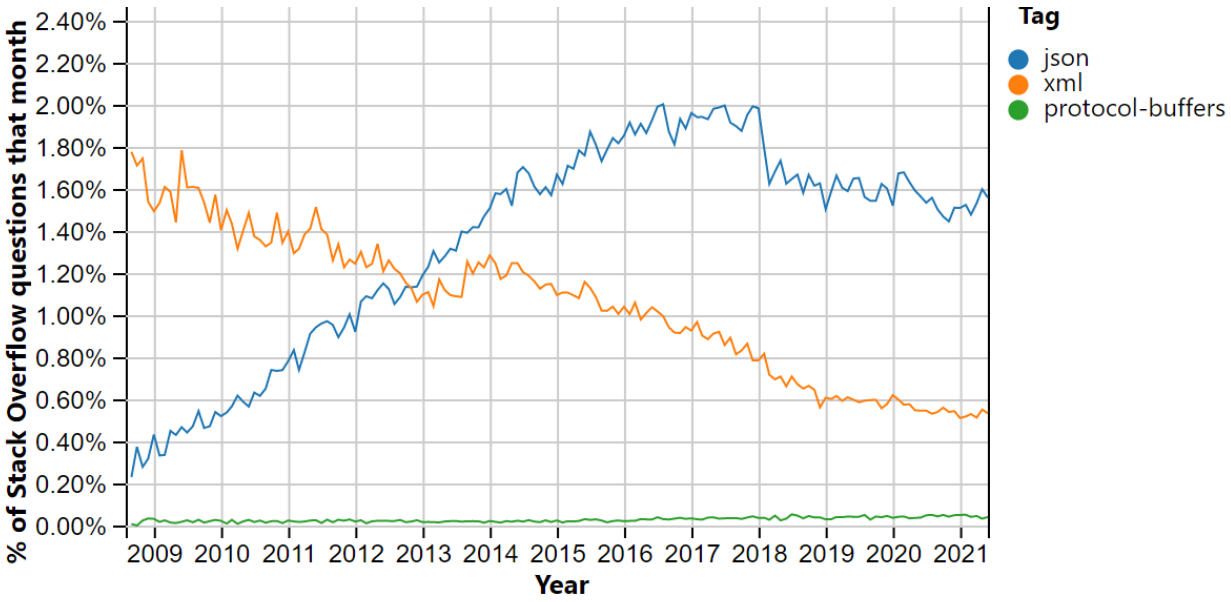

Рисунок 4 – Динамика популярности JSON, XML, protocol-buffers

На графике не отображен CBOR, что обусловлено слишком малым количеством его упоминаний на платформе Stack Overflow. Можно предположить, что популярность CBOR, как минимум, ниже GPB.

Можно сразу отметить, что популярность текстовых протоколов (JSON и XML) на порядок выше, чем бинарных (GPB). В качестве важного замечания нужно отметить, что если в 2009 году XML был примерно в 4 раза популярнее JSON, то уже в 2021 году ситуация диаметрально изменилась и JSON на данный момент примерно в 3 раза популярнее XML.

Исходя из раннее представленного графика, в рамках проводимого сравнительного анализа необходимо определить, что является причиной роста популярности JSON относительно XML и что является причиной крайне низкой популярности бинарных форматов сериализации по сравнению с текстовыми.

JSON является более компактным и минималистичным форматом сериализации [3]. Если в JSON массивы записываются посредством перечисления элементов в квадратных скобках, то в XML такая запись выглядит как набор отдельных элементов, каждый из которых обернут парой тегов [3].

Процесс сериализации и десериализации данных в JSON проще, чем XML. «Для получения данных из XML используя JS, нужно: определить XML-документ, использовать DOMParser() для создания парсера, с помощью парсера и метода

parseFromString() создать XMLDOM-объект, определить новые переменные и присвоить им значения. В случае с JSON нужно всего лишь использовать функцию JSONparse(), получив JavaScript-объект» [3].

Двух приведенных преимуществ JSON над XML стало достаточно, чтобы в период популяризации клиент-серверной архитектуры JSON догнал и перегнал XML по распространённости, кроме того, одно из исследований [7] показало, что JSON колоссально превосходит XML по производительности.

Несмотря на то, что JSON, как и остальные текстовые форматы сериализации, значительно уступает бинарным форматам (например, в сравнении с GPB) по производительности (таблицы 1 и 2) [8], он на порядок превосходит их в распространенности [6].

Таблица 1.

Сравнение производительности JSON и GPB для числовых данных

Формат	**Сериализация**	**Десериализация**	**Итоговый размер**
JSON	4358,13 ms	2730,19 ms	36 mb
GPB	865,06 ms	77,38 ms	11 mb

Таблица 2.

Сравнение производительности JSON и GPB для строковых данных

Формат	**Сериализация**	**Десериализация**	**Итоговый размер**
JSON	5958,92 ms	4039,24 ms	124 mb
GPB	1856,20 ms	256,02 ms	73 mb

Из таблиц видно, что по всем 3 объективным характеристикам бинарный формат GPB превосходит текстовый JSON, в большей мере на числовых данных и в меньшей на строковых. Исходя из данных таблиц, можно прийти к выводу, что популярность текстовых форматов сериализации обусловлена не объективным характеристиками.

В различных источниках [1–3] говорится, что текстовые форматы сериализации зачастую выбираются разработчиками исходя не из производительности, а из

простоты, читаемости, гибкости, наличия качественной документации и легкости отладки. Напрашивается вывод, что при разработке, если не предъявляются жесткие требования к производительности и объему трафика, лучше использовать текстовые форматы сериализации.

Что же касается сравнения CBOR и GPB, то оно может быть представлено в виде таблиц 3 и 4 [8].

Таблица 3.

Сравнение производительности CBOR и GPB для числовых данных

Формат	**Сериализация**	**Десериализация**	**Итоговый размер**
CBOR	924.74 ms	144.96 ms	11 mb
GPB	865,06 ms	77,38 ms	11 mb

Таблица 4.

Сравнение производительности CBOR и GPB для строковых данных

Формат	**Сериализация**	**Десериализация**	**Итоговый размер**
CBOR	2087.63 ms	472.31 ms	75 mb
GPB	1856,20 ms	256,02 ms	73 mb

Из таблиц видно, что, несмотря на более позднюю дату создания, CBOR уступает GPB в скорости десериализации, при этом по остальным характеристикам разница между этими двумя форматами незначительна.

Подводя общий итог, имеет смысл собрать все имеющиеся существенные данные из источников вместе для обобщенного сравнения 4 рассматриваемых форматов сериализации. Для окончательного сравнительного анализа используется таблица 5.

Таблица 5.

Сравнение форматов сериализации данных XML, JSON, CBOR, GPB

Формат	**Произво-дительность**	**Поддержка массивов?**	**Человеко-читаемый?**	**Стандар-тизирован?**	**Поддержка ссылок?**

XML	+	Нет	Да	Да	Да
JSON	+++	Да	Да	Да	Да
CBOR	+++++	Да	Нет	Да	Да
GPB	++++++	Да	Нет	Нет	Нет

В заключение можно сказать, что если к системе не предъявляются строгие требования по производительности, то лучше использовать текстовый формат сериализации, например, JSON (эффективный и универсальный). В ином случае имеет смысл выбрать какой-нибудь бинарный формат, отвечающий задачам предметной области. При этом, если специальных требований нет, то разумно будет использовать GPB ввиду его высокой производительности.

Литература

1. Танатканова А.К. Построение клиент-серверных приложений / Танатканова А.К., Жамбаева А.К. // Наука и образование сегодня – 2019. – № 6– 2 (41).
2. Канаев К.А. Сравнительный анализ форматов обмена данными, используемых в приложениях с клиент-серверной архитектурой / Канаев К.А., Фалеева Е.В., Пономарчук Ю.В. // Фундаментальные исследования – 2015. – Т. 25 – № 2.
3. Казначеева Е.О. Эволюция форматов обмена данными на веб-платформе на примере XML и JSON , 2017. – 114–117с.
4. Bormann C.Concise binary object representation (cbor) / C. Bormann, P. Hoffman – , 2013.
5. Protocol Buffers// Google Developers [Электронный ресурс]. URL: https://developers.google.com/protocol-buffers (дата обращения 10.07.2021).
6. Stack Overflow Trends// Stack Overflow [Электронный ресурс]. URL: https://insights.stackoverflow.com/trends (дата обращения 11.07.2021).
7. Nurseitov N. Comparison of JSON and XML data interchange formats: a case study. / Nurseitov N., Paulson M., Reynolds R., Izurieta C. // Caine – 2009. – Т. 9 – С.157–162.
8. Рянов Н. Сравнение форматов сериализации// Habr [Электронный ресурс].

URL: https://habr.com/ru/post/458026/ (дата обращения 12.07.2021).

Literature

1. Tanatkanova A.K. Building client-server applications / Tanatkanova A.K., Zhambaeva A.K. // Science and Education Today - 2019. - № 6–2 (41).
2. Kanaev K.A. Comparative analysis of data exchange formats used in applications with client-server architecture / Kanaev K.A., Faleeva E.V., Ponomarchuk Yu.V. // Basic Research - 2015. - V. 25 - No. 2.
3. Kaznacheeva E.O. Evolution of data exchange formats on a web platform on the example of XML and JSON, 2017. - 114-117p.
4. Bormann C. Concise binary object representation (cbor) / C. Bormann, P. Hoffman -, 2013.
5. Protocol Buffers // Google Developers [Electronic resource]. URL: https://developers.google.com/protocol-buffers (date of access 07/10/2021).
6. Stack Overflow Trends // Stack Overflow [Electronic resource]. URL: https://insights.stackoverflow.com/trends (date accessed 07/11/2021).
7. Nurseitov N. Comparison of JSON and XML data interchange formats: a case study. / Nurseitov N., Paulson M., Reynolds R., Izurieta C. // Caine - 2009. - T. 9 - C.157–162.
8. Ryanov N. Comparison of serialization formats // Habr [Electronic resource]. URL: https://habr.com/ru/post/458026/ (date of treatment 07/12/2021).