

АНАЛІЗ МЕТОДІВ СЕРІАЛІЗАЦІЇ ОБ'ЄКТІВ ДЛЯ ПОБУДОВИ ПЛАТФОРМИ ВЕЛИКИХ ІНДУСТРІАЛЬНИХ ДАНИХ

Попенко Д.В., Курдеча В.В.

Інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна

E-mail: demyd.popenko@gmail.com

Analysis of object serialization methods for building a platform of big industrial data

The article describes existing methods of object serialization, investigates and analyzes the performance of methods for different amounts of computational objects, which is important for building a platform for collecting big industrial data. A method has been identified that will most effectively convert data structures into a bit sequence.

У статті описуються існуючі методи серіалізації об'єктів, досліджено і проаналізовано продуктивність методів при різній кількості обчислювальних об'єктів, що є важливим для побудови платформи великих індустріальних даних. Визначено метод, який дозволить за критерієм часу найбільш ефективно перетворювати структури даних у послідовність бітів.

Індустріальний Інтернет речей - це система об'єднаних комп'ютерних мереж і промислових об'єктів, які пов'язані між собою вбудованими датчиками і програмним забезпеченням для збору, обробки та обміну даними. Це забезпечує проведення віддаленого моніторингу та управління в автоматичному режимі без втручання людини.

Метою даної роботи є вибір методу серіалізації, який забезпечить мінімальні часові витрати на обробку об'єктів.

Отримана з пристроїв на підприємстві інформація може бути використана для запобігання позапланових простоїв, поломок устаткування, скорочення позапланового техобслуговування і збоїв в управлінні поставками, тим самим дозволяючи підприємству функціонувати більш ефективно. Але різні види пристроїв мають різні типи даних, які в сукупності являють собою масивні неоднорідні набори інформації з багатьох джерел. Це призводить до великих витрат на збір даних та інтеграцію обладнання з системою обробки даних. Саме тому необхідно вибрати найбільш продуктивний метод серіалізації, який забезпечить мінімальний час перетворення будь-яких структур даних в бітову послідовність і передачу об'єктів по мережі.

На рисунку 2 наведено загальний принцип процесу серіалізації.

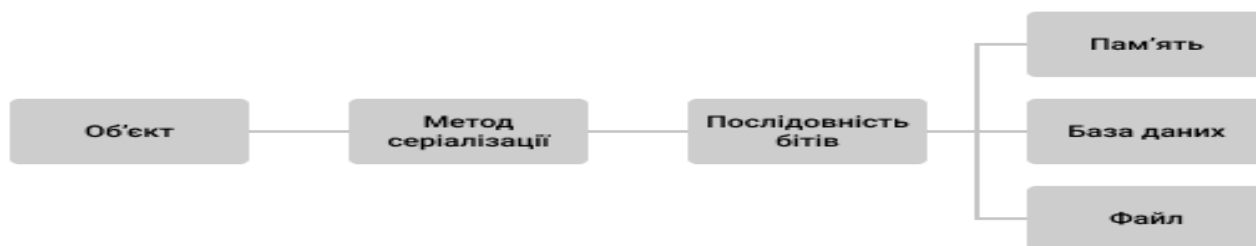


Рис.2. Загальний процес серіалізації.

Розглянемо чотири методи: Protobuf, Kryo, Protostuff, Jackson.

Jackson [2] орієнтований на поєднання швидкості та ергономічності. Він пропонує три варіанти обробки JSON. Модель дерева забезпечує оптимізовані для обробки дані, представлені у вигляді дерева JSON документа.

Protostuff [3] – це розширена версія протокольних буферів (Protocol Buffers) від Google. Protostuff підтримує багато форматів, серед яких JSON, XML, YAML та KVP.

Протокол буферів (Protocol Buffers) [4] використовується в Google для кодування структурованих даних в ефективному форматі. Він кодується до двійкових даних для забезпечення меншої та швидшої серіалізації.

Kryo [5] - це середовище серіалізації Java, орієнтоване на швидкість, ефективність та зручність API. Фреймворк надає клас Kryo в якості основної точки входу до всіх його функціональних можливостей.

Для оцінювання часу серіалізації, використовуючи дані дослідження [6] необхідно визначити функції, якими можна описати час серіалізації та десеріалізації в залежності від кількості оброблюваних об'єктів. Для цього необхідно заздалегідь підготувати набори інформації, які ми будемо подавати як вхідні дані для чотирьох методів серіалізації. Для фіксації часу і отримання точніших результатів проведемо три ітерації для різної кількості об'єктів. У таблиці 1 наведені результати для кожного методу.

Таблиця 1. Час серіалізації та десеріалізації для різних методів.

Кількість об'єктів серіалізації	1*10 ⁴	1*10 ⁵	1*10 ⁶	1*10 ⁷	1*10 ⁸
Kryo (сек)	0.3	0.44	2.33	18.42	215
Protostuff (сек)	0.11	0.24	1.11	8.37	87
Protobuf (сек)	0.45	0.62	1.2	8.73	82
Jackson (сек)	1.7	1.89	3.58	18.67	246

Використовуючи метод лінійної інтерполяції [7] отримали систему рівнянь для кожного методу. Для прикладу нижче наведено формулу, яка описує час серіалізації для методу Kryo.

$$t = \begin{cases} 0.3 + 1.556 * 10^{-6} * (n - 10^4), & 0 \leq n \leq 10^5 \\ 0.44 + 2.1 * 10^{-6} * (n - 10^5), & 10^5 < n \leq 10^6 \\ 2.33 + 1.788 * 10^{-6} * (n - 10^6), & 10^6 < n \leq 10^7 \\ 18.42 + 2.184 * 10^{-6} * (n - 10^7), & 10^7 < n \leq 10^8 \end{cases} \quad (1)$$

де t – час серіалізації та десеріалізації, n – кількість об'єктів.

Для візуалізації отриманих результатів на рисунку 3 побудовані графіки на основі отриманих рівнянь.

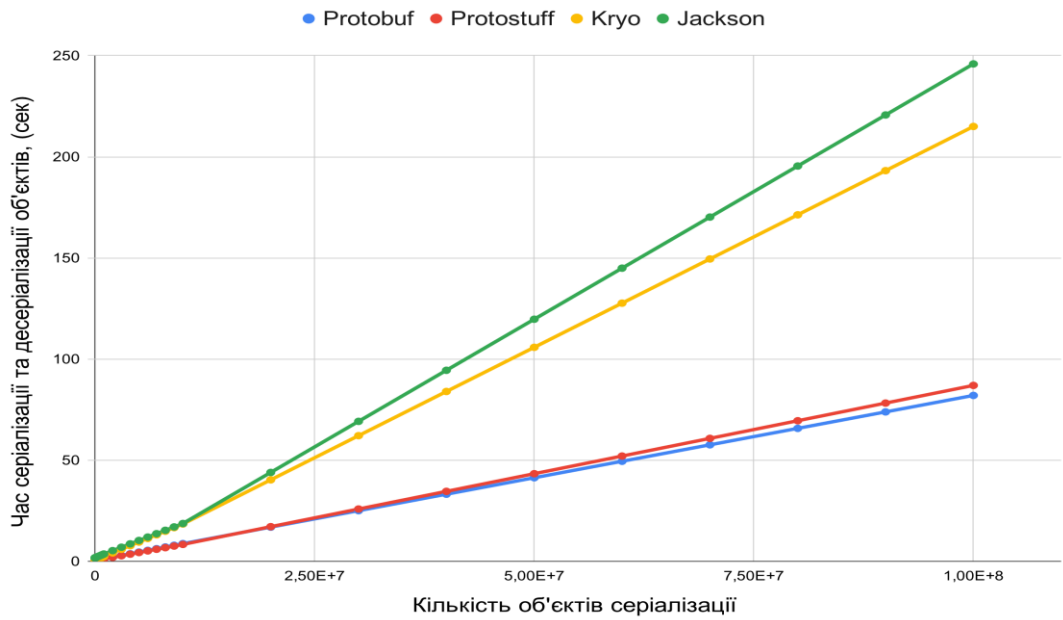


Рис.3. Час серіалізації чотирьох методів для різної кількості об'єктів обробки.

В даній роботі розглянуто та проаналізовано чотири методи серіалізації: Kryo, Protostuff, Protobuf, Jackson. Отримано системи рівнянь для кожного методу. На основі їх аналізу визначено найефективніший метод в залежності від кількості об'єктів серіалізації за критерієм часу обробки.

Найкращий результат отримано використовуючи Protobuf. В подальшій роботі при проектуванні платформи великих промислових даних отримані системи рівнянь можна використовувати для визначення часу, який витрачається на серіалізації та десеріалізації структур даних.

Література

1. State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating [Електронний ресурс] // IoT Analytics. – 2018. – Режим доступу до ресурсу: <https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>.
2. Serialization with Jackson [Електронний ресурс] // Akka Documentation – Режим доступу до ресурсу: <https://doc.akka.io/docs/akka/current/serialization-jackson.html#serialization-with-jackson>.
3. Protostuff - java serialization library, proto compiler, code generator [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/protostuff/protostuff>.
4. Protocol Buffers - Google's data interchange format [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/protocolbuffers/protobuf>.
5. Introduction To Kryo [Електронний ресурс] // Baeldung. – 2019. – Режим доступу до ресурсу: <https://www.baeldung.com/kryo>.
6. Big Data-Based Improved Data Acquisition and Storage System for Designing Industrial Data Platform [Електронний ресурс] // IEEE. – 2019. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/8681030>.
7. Linear interpolation [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Linear_interpolation.