Java binary serialization review

Sirenko A.

December 3, 2014

2014

Requirements

We use serialization in 2 cases:

1. to store;
2. to transfer;

Main features to consider:

1. serialize/deserialize speed;
2. serialized size;
3. stability to change;
4. laboriousness;
5. platform-dependence;

JDK tools

- Standard serialization: smallest effort / worst performance;
- Externalizable serialization: full control over serialization, define your binary format, no features;

serialization	speed	size	stability to change	laboriousness	platform dependence
standard	poor	big	average	low	only java
externalizable	fast	small	poor	high	independent

Table : Standard serialization vs Externalizable

Why?

Motivation:

- conveniently simple and fast approaches;
- better adopted for our data: optional params, variable-length numbers, aliases;
- platform-independent: same description of our data for different platforms.

Here we consider:

- protobuf - https://code.google.com/p/protobuf/;
- avro - http://avro.apache.org/;
- kryo - https://github.com/EsotericSoftware/kryo.

Protobuf

Protobuf info

Avro

Avro info

Kryo

Kryo info

Comparison

Comparison placeholder

Kryo vs. Std in queue

results placeholder

Kryo vs. Std in queue

results placeholder