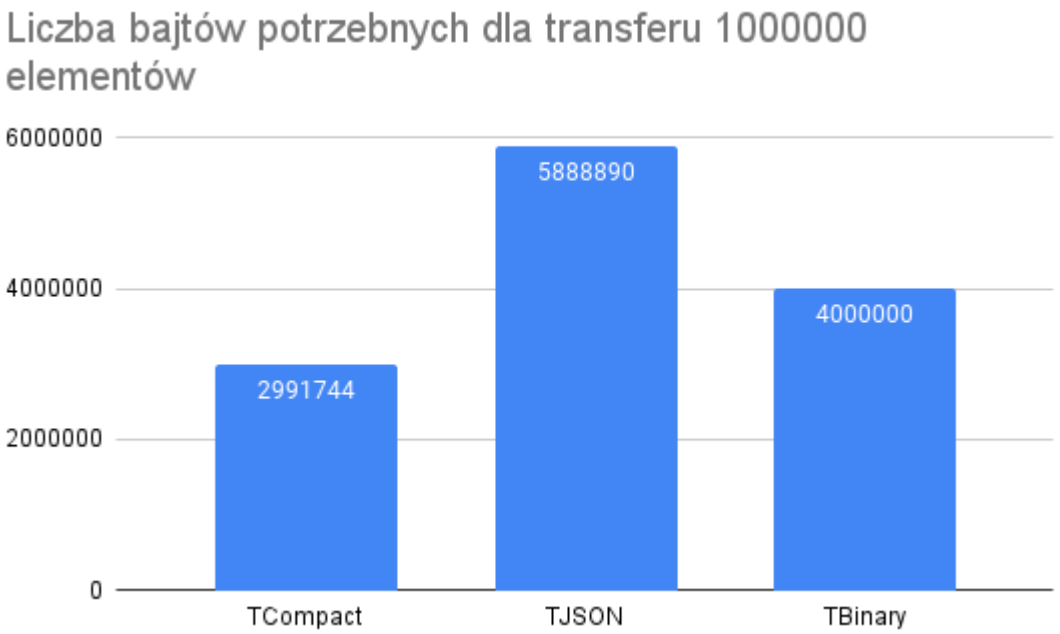
Środowisko testowe

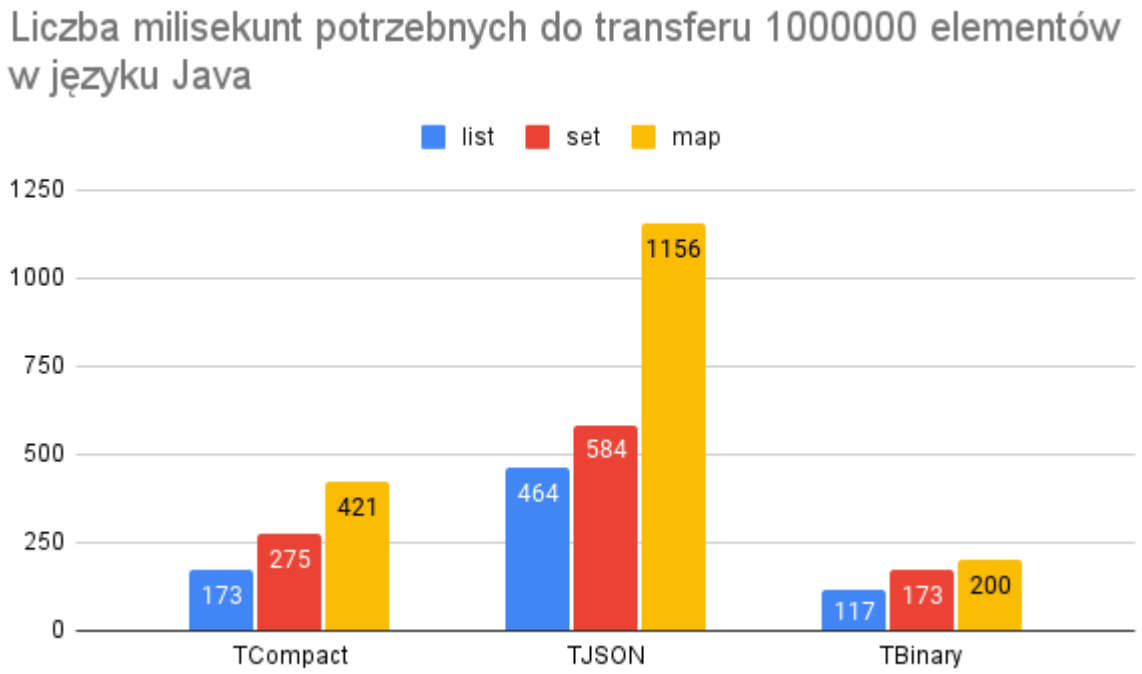
W celu zbadania różnych sposobów serializacji oferowanych przez Thrift (protokoły TBinary, TJSON i TCompact) stworzono dwie analogiczne aplikacje klient-serwer w językach Java i Python. W ramach testów klienci przesyłali 1000000 elementów przechowywanych w różnych strukturach danych (lista, set, mapa) oraz mierzyli czas transferu tych danych do serwera.

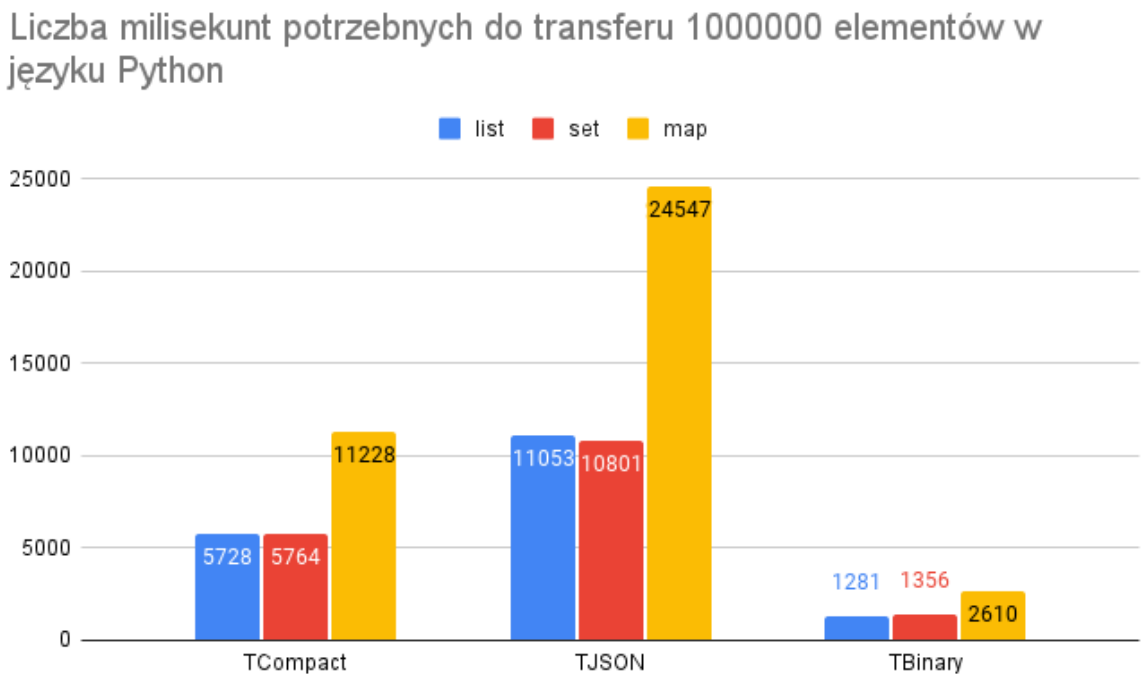
Dla mniejszej liczby elementów różnica pomiędzy strukturami danych jak i językami była nieistotna, natomiast większą różnicę wykazała większa liczba elementów.

Sprawdzono również liczbę przesyłanych bajtów dla każdego z protokołów.

Wyniki







Wnioski

Z przedstawionych rezultatów testów wynika, że protokół TCompact zużywa najmniej pamięci, czyli jest najbardziej efektywny. Najmniej efektywny jest protokół TJSON. Natomiast najbardziej wydajny (czyli zużywający najmniej czasu) jest protokół TBinary. Najmniej wydajny jest protokół TJSON.

Zgodnie z oczekiwaniami czas transferu map w każdym z przypadków zajął prawie dwukrotnie więcej czasu niż w przypadku list i setów. To jest związane z tym, że mapa przechowuje dwa razy więcej elementów (1000000 par klucz-wartość). Porównywając sety i listy widzimy, że w języku Python czas ich transferu jest niemal taki sam, ale w przypadku języka Java transfer list zajął mniej czasu w każdym z protokołów.

Widzimy również, że dla każdej struktury danych oraz dla każdego protokołu język Java jest o wiele bardziej wydajny niż Python.

Efektywność transferu danych (czyli liczba przekazanych bajtów) w obu językach była taka sama.

Podsumowując można zauważyć, że najbardziej efektywne rozwiązanie jest nieco mniej wydajne, a najbardziej wydajne rozwiązanie jest nieco mniej efektywne. Wybierając konkretny protokół musimy brać pod uwagę co jest dla nas najbardziej ważniejsze - mniejsze zużycie pamięci, mniejsze zużycie czasu albo może czytelność przekazywanych danych? Dla każdego przypadku lepiej będzie pasował tylko jeden protokół; nie ma takiego protokołu, który byłby bezwzględnie lepszy albo bezwzględnie gorszy, ponieważ każdy jest dobry dla różnych zastosowań.