

Benchmarking Java Streams

Processamento de Dados com Streams de JAVA

Afonso Silva

Octávio Maia

8 de Janeiro de 2018

Conteúdo

I. Testes	2
1. Cálculo dos valores de transações registadas	3
Observações	3
Métodos a testar	3
Resultados	4
Análise e conclusões	4
2. Extração dos primeiros e últimos 20% de transações realizadas	5

Parte I.

Testes

1. Cálculo dos valores de transações registradas

Observações

Métodos a testar

```
public double sumValuesArray() {  
    double[] values = new double[transactions.size()];  
    int i = 0;  
  
    for (TransCaixa transaction : transactions) {  
        values[i++] = transaction.getValor();  
    }  
  
    double sum = 0.0;  
  
    for (i = 0; i < values.length; i++) {  
        sum += values[i];  
    }  
  
    return sum;  
}
```

Listing 1.1: Cálculo da soma dos valores das transações através de um array do tipo double

```
public double sumValuesDoubleStream() {  
    DoubleStream values = transactions.stream()  
        .mapToDouble(TransCaixa::getValor);  
    return values.sum();  
}
```

Listing 1.2: Cálculo da soma dos valores das transações através de uma DoubleStream

```
public double sumValuesDoubleStreamParallel() {  
    DoubleStream values = transactions.parallelStream()  
        .mapToDouble(TransCaixa::getValor);  
    return values.sum();  
}
```

Listing 1.3: Cálculo da soma dos valores das transações através de uma DoubleStream paralela

```

public double sumValuesStream() {
    Stream<Double> values = transactions.stream()
        .map(TransCaixa::getValor);
    return values.reduce(0.0, (a,b) -> a + b);
}

```

Listing 1.4: Cálculo da soma dos valores das transações através de Stream<Double>

```

public double sumValuesStreamParalell() {
    Stream<Double> values = transactions.parallelStream()
        .map(TransCaixa::getValor);
    return values.reduce(0.0, (a,b) -> a + b);
}

```

Listing 1.5: Cálculo da soma dos valores das transações através de uma Stream<Double> paralela

Resultados

Input	(1) double[]	(2) Double Stream	(3) Double Stream (parallel)	(4) Stream Stream <Double>	(5) Stream <Double> (parallel)
1000000 transactions	0,012632	0,015631	0,009870	0,031217	0,020908
2000000 transactions	0,022243	0,025128	0,018836	0,063035	0,036117

Análise e conclusões

2. Extração dos primeiros e últimos 20% de transações realizadas