

Resumo de Benchmarks JMH

5 de novembro de 2025

0.1 Introdução

Este documento apresenta um resumo dos benchmarks realizados com a ferramenta *Java Microbenchmark Harness (JMH)*, descrevendo os tipos de testes e as variáveis utilizadas.

0.2 Código de Execução do Benchmark

```
package org.sample;

import org.openjdk.jmh.runner.Runner;
import org.openjdk.jmh.runner.options.Options;
import org.openjdk.jmh.runner.options.OptionsBuilder;

public class Main {
    public static void main(String[] args) throws Exception {
        Options opt = new OptionsBuilder()
            .include("BlackholeBench")
            .forks(1)
            .build();

        new Runner(opt).run();
    }
}
```

0.3 Tipos de Benchmarks

0.3.1 BlackholeBench

Declara variáveis: `byte`, `boolean`, `char`, `short`, `int`, `long`, `float`, `double`, `Object`.

Possui dois tipos de benchmark:

- **Implicit:** mede o tempo de acessar e retornar a variável. A JVM pode otimizar e eliminar o retorno se ele não for utilizado.
- **Explicit:** garante que o valor seja utilizado de forma que a JVM não possa otimizar, medindo o custo real de acesso.

0.3.2 BlackholeConsecutiveBench

Mede o tempo médio da JVM para executar operações simples.

Variáveis: `boolean`, `byte`, `short`, `char`, `int`, `float`, `long`, `double`, `Object`.

Cada variável possui três métodos:

- 1: consome uma operação;
- 4: consome quatro operações consecutivas;
- 8: consome oito operações consecutivas.

Observação: garanta que o valor seja usado de forma que a JVM não possa otimizar. Os métodos 4 e 8 medem o custo de múltiplas operações consecutivas.

0.3.3 BlackholeConsumeCPUBench

```
public class BlackholeConsumeCPUBench {  
  
    '''  
    @Param("0")  
    private int delay;  
  
    @Benchmark  
    public void consume() {  
        Blackhole.consumeCPU(delay);  
    }  
    '''  
  
}
```

Descrição:

- Executa um loop vazio com `delay` iterações;
- Gasta CPU para simular trabalho;
- Mede o tempo médio gasto por chamada de `consumeCPU`.

0.3.4 BlackholePipelineBench

Mede o custo de processar sequências de elementos (arrays).

Variáveis: `boolean`, `byte`, `short`, `char`, `int`, `float`, `long`, `double`, `Object`, `Array` de objetos.

Mede o custo de percorrer arrays de elementos consecutivos.

0.4 Resumo em Tabela

Benchmark	Tipo de Variáveis	Descrição
-----------	-------------------	-----------

BlackholeBench	byte, boolean, char, short, int, long, float, double, Object	Implicit: acesso otimizado; Explicit: acesso real sem otimização
BlackholeConsecutiveBench	boolean, byte, short, char, int, float, long, double, Object	Mede custo de 1, 4 ou 8 operações consecutivas, garantindo uso de valores
BlackholeConsumeCPU- Bench		Executa loop vazio para simular CPU Mede tempo médio de consume CPU
BlackholePipelineBench	boolean, byte, short, char, int, float, long, double, Object, Array	Mede custo de percorrer arrays de elementos consecutivos