



Medição do Custo Energético de Execução de Software

Orientador: Lucio Mauro Duarte

Aluno: Bruno Ramos Toresan

Objetivo:

Habilitar a utilização do framework jRAPL para medir a energia de componentes de software na linguagem Java. O resultado é obtido pela subtração dos valores de potência consumida antes e depois do trecho de código alvo com o auxílio do método `EnergyCheckUtils.getEnergyStats()` da ferramenta.

```
EnergyCheckUtils.init();  
// ...  
double[] before = EnergyCheckUtils.getEnergyStats();  
// target code  
double[] after = EnergyCheckUtils.getEnergyStats();  
double consumption = ((after[0] - before[0]) + //DRAM  
                      (after[1] - before[1]) + //CPU  
                      (after[2] - before[2])) / 10.0; //Package  
System.out.println("Power consumption Total: " + consumption);
```

Figura 1: Exemplo de utilização da ferramenta que mede o consumo de potência de trechos de código em Java.

Aplicações:

Combinando a utilização dessa ferramenta com um LTS (Labelled Transition System) é possível construir um modelo que associa o comportamento de um software com seus custos de energia. Isso habilitaria os desenvolvedores a fazerem análises e otimizações baseadas no custo energético de seus códigos.

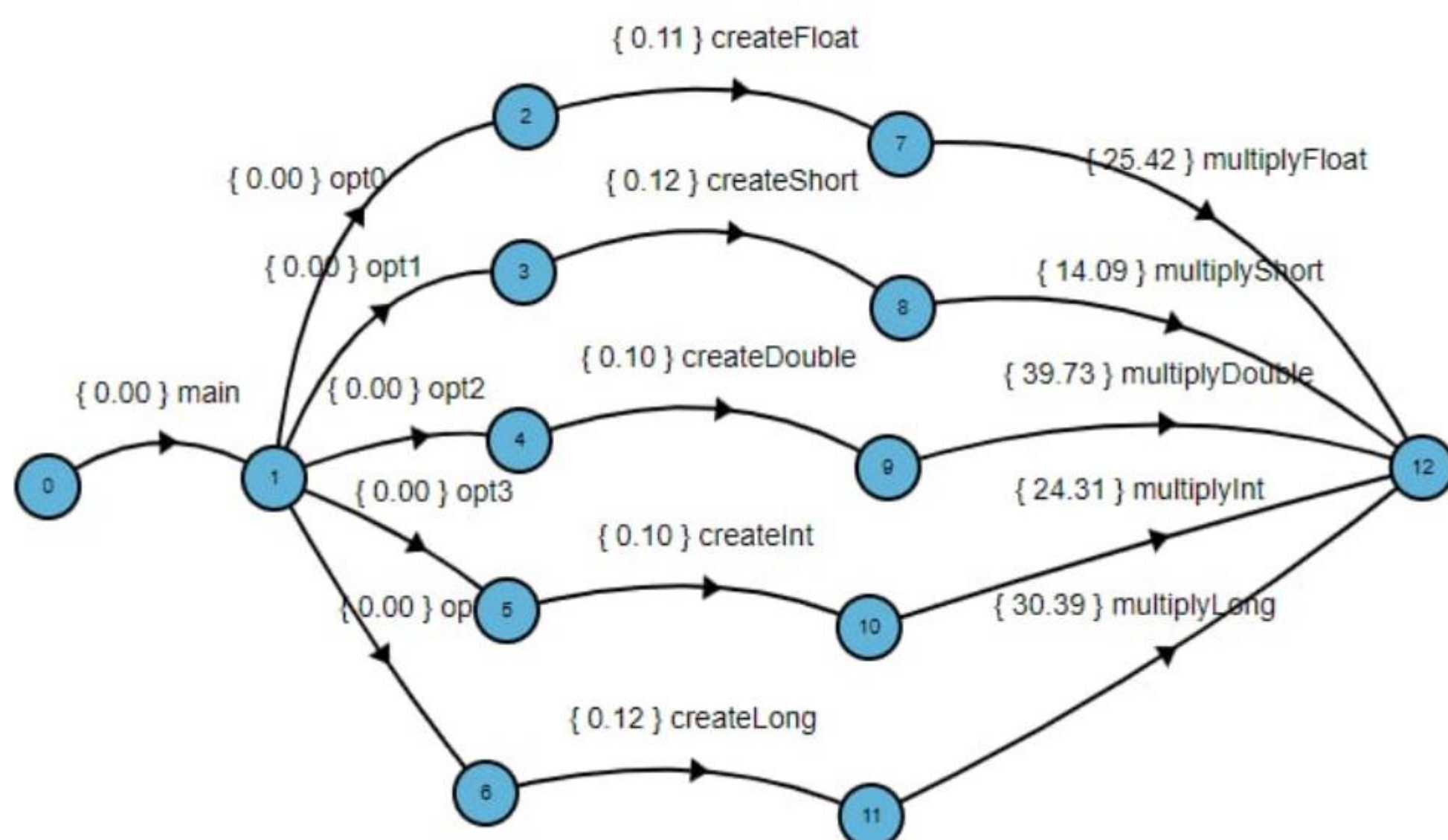


Figura 2: LTS de um algoritmo de multiplicação de matrizes em que cada matriz tem um tamanho 100x100 contendo os tipos Short, Int, Long, Float ou Double.

Conclusão:

O framework jRAPL acessa registradores de propósito especial que guardam informações sobre energia e potência consumida. Após algumas modificações no código fonte e integrações com uma IDE, habilitamos a medição do custo energético de códigos fonte em Java. Na figura abaixo temos um exemplo de execução mostrando a potência consumida em cada nodo do LTS ilustrado na figura 2, que representa um algoritmo de multiplicação de matrizes de diferentes tipos.

```
brtoresan@verites-desktop:  
sudo java -jar dist/MatMult.jar 3  
Power consumption 2: 0.13032839999978024  
Power consumption 7: 34.405674699999998  
Power consumption Total: 34.55126950000023  
brtoresan@verites-desktop:  
sudo java -jar dist/MatMult.jar 1  
Power consumption 3: 0.13645170000023654  
Power consumption 8: 18.583299399999873  
Power consumption Total: 18.731668299999466  
brtoresan@verites-desktop:  
sudo java -jar dist/MatMult.jar 0  
Power consumption 4: 0.13591920000042138  
Power consumption 9: 51.33267670000039  
Power consumption Total: 51.47957770000018  
brtoresan@verites-desktop:  
sudo java -jar dist/MatMult.jar 2  
Power consumption 5: 0.13708799999949406  
Power consumption 10: 32.41206960000024  
Power consumption Total: 32.565435699999966  
brtoresan@verites-desktop:  
sudo java -jar dist/MatMult.jar 4  
Power consumption 6: 0.14328310000055353  
Power consumption 11: 44.339299000000594  
Power consumption Total: 44.49114840000059
```

Figura 3: Exemplo de execução da ferramenta

Agradecimentos: