

Análise e Teste de Software

Green Profiling

Implementação de um método energia em c--

David Alves a53791
Nelson Torres pg31063
Pedro Lopes a32652

RAPL

Biblioteca da Intel que mede os
consumo de energia em joules

- DRAM
- CPU
- Package

Fase 1

Introdução da Instrução na Gramática

Gramática(1)

- Conseguir detectar o método energia() no c--
- Gom:

```
Expressao = ExpNum(Exp1:Expressao,c1:LComentarios,op:OpNum,c2:LComentarios,Exp2:Expressao)
| Id(Id:String)
| Pos(Expressao:Expressao)
| Neg(Expressao:Expressao)
| Nao(Expressao:Expressao)
| Call(c1:LComentarios,Id:String,c2:LComentarios,c3:LComentarios,Parametros:Parametros,c4:LComentarios,
c5:LComentarios)
| IncAntes(OpInc:OpInc,Id:String)
| IncDepois(OpInc:OpInc,Id:String)
| Condicional(Condicao:Expressao,c1:LComentarios,c2:LComentarios,Exp1:Expressao,c3:LComentarios,
c4:LComentarios,Exp2:Expressao)
| Int(Int:int) | Char(Char:String) | True() | False() | Float(num:int)
| Ou(Cond1:Expressao,c1:LComentarios,c2:LComentarios,Cond2:Expressao)
| E(Cond1:Expressao,c1:LComentarios,c2:LComentarios,Cond2:Expressao)
| Comp(Exp1:Expressao,c1:LComentarios,OpComp:OpComp,c2:LComentarios,Exp2:Expressao)
| Input(c1:LComentarios,c2:LComentarios,c3:LComentarios,Tipo:DefTipo,c4:LComentarios,c5:LComentarios)
| Print(c1:LComentarios,c2:LComentarios,c3:LComentarios,Expressao:Expressao,c4:LComentarios,
c5:LComentarios)
| DEnergia(c1:LComentarios,c2:LComentarios,c3:LComentarios,c4:LComentarios)
| Expressoes(Expressao*)
| Empty()
```

Gramática(2)

Antlr:

```
instrucao :  
    (if_ -> if_ | for_ -> for_ | while_ -> while_ | return_ ';' ->  
    return_ | call ';' -> ^(Exp call) | print_ ';' -> print_ |  
    energia_ ';' -> energia_ )  
    ;
```

```
energia_ :  
    c1=comentarios 'energia' c2=comentarios '(' c3=comentarios ')' '  
    c4=comentarios -> ^(Exp ^(DEnergia ^(Comentarios $c1?) ^(Comentarios $c2?) (Comentarios $c3?) ^(Comentarios $c4?)))  
    ;
```



GOM

Gramática(3)

- Tom:

```
Expressoes(exp1, exp*) -> {  
    String genExp = `compileAnnotExpressoes(exp1, numInstrucao);  
    String exps = genExp.concat(`compileAnnotExpressoes(exp*, numInstrucao));  
  
    return exps;  
}  
DEnergia(_,_,_,_) -> {  
    return "Energy,";  
}  
Empty() -> { return ""; }  
}  
return "";  
}
```

Implementação Thread(1)

- energia() no MSP:

```
public void getEnergia() throws IOException{
    try{
        final Process x = Runtime.getRuntime().exec("sudo java EnergiaExample");
        new Thread(new Runnable() {
            public void run() {
                DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
                Calendar cal = Calendar.getInstance();
                //out.println(dateFormat.format(cal.getTime()));
                String hora= dateFormat.format(cal.getTime());
                PrintWriter out = null;
                try {
                    out = new PrintWriter(new FileWriter("../logs/Energia "+hora+".log"));
                    BufferedReader input = new BufferedReader(new InputStreamReader(x.getInputStream()));
                    String line = null;
                    try {
                        while ((line = input.readLine()) != null)
                            out.println(line);
                        out.close();
                    } catch (IOException e) {}
                } catch (IOException ex) {
                    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
                } finally {
                    out.close();
                }
            }
        }).start();
        x.waitFor();
    } catch (InterruptedException e) {}
}
```

Implementação Thread(2)

- Quando a Thread acaba escreve no ficheiro a quantidade de energia lida:

```
dram: 2141.540131 cpu: 24900.20079 package: 43571.289398
```


Fase 2

Tratamento de Informação/Redução de Overhead

Melhorias da Thread

- Melhoria na implementação da Thread
 - Se programa acabasse e estivesse a medir não guardava a medição;
- Espera-se sempre que ela acabe;
- Só no final de executar o c-- é que escreve no ficheiro;

Fase 3

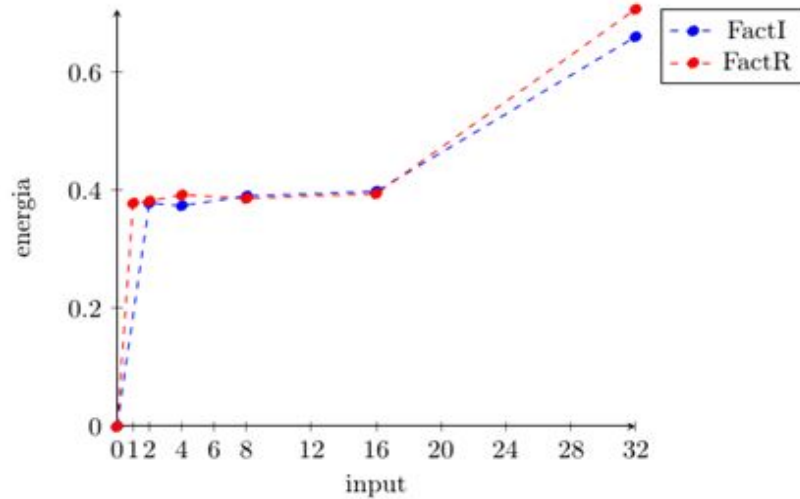
Implementação de novas funcionalidades

Novas Implementações

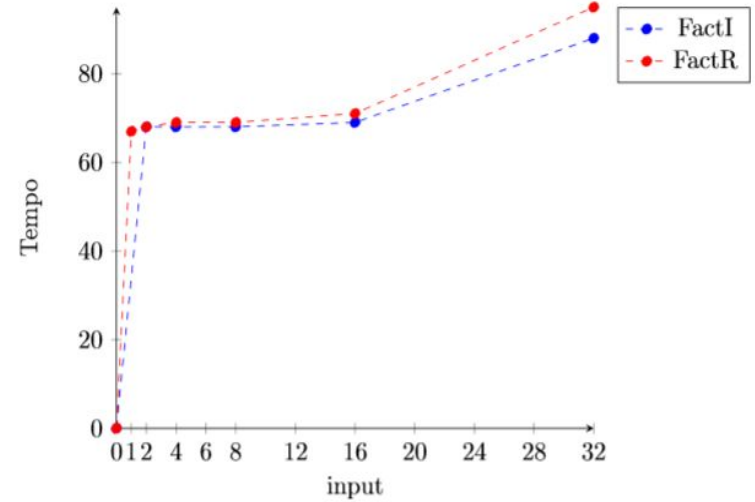
- Medições de energia automáticas
 - Sempre que se deteta uma chamada a uma função energia() mede a energia;
- Tempo de execução quando faz o energia();

Resultados

1 CPU



3 Time



Análise e Teste de Software

Green Profiling

Implementação de um método energia em c--

David Alves a53791
Nelson Torre pg31063
Pedro Lopes a32652