Programmazione Concorrente e Distribuita - Assigment 02

Andrea Biagini
Filippo Gurioli
Leonardo Randacio

Università di Bologna
Scienze e Ingegneria Informatiche

April 20, 2024

Contents

List of Figures

Chapter 1

Macropart 1

Chapter 2

Macropart 2

2.1 Task Decomposition

The main task can be rappresented in pseudocode with a recursive function `getWordOccurrences(str URL, str WORD, int DEPTH)`

The function identifies two tasks:

- countTask
- linkTask

The countTask finds all occurrences of the defined word in the website, while the linkTask finds all the links and for every link starts the getWordOccurrences on the new url found. Both subtasks are expressed below:

Algorithm 2.1: countTask expressed in pseudocode

```
1  input: str SITE_CONTENTS, str WORD
2  output: int
3  begin
4      int COUNT ← 0
5      foreach CURRENT_WORD in SITE_CONTENTS
6          if (CURRENT_WORD == WORD)
7              COUNT ← COUNT + 1
8          endif
9      endforeach
10     return COUNT
11 end
```

Algorithm 2.2: linkTask expressed in pseudocode

```
1  input: str SITE_CONTENTS, str WORD, int DEPTH
2  begin
3      if (DEPTH != 0)
4          foreach CURRENT_WORD in SITE_CONTENTS
5              if (is_url(CURRENT_WORD))
6                  getWordOccurrences(CURRENT_WORD, WORD, DEPTH - 1)
7              endif
8          endforeach
9      endif
10 end
```

2.2 Event-loop

Source code: TODO ADD THE SOURCE CODE DIR The tasks are rappresented by events. The first function call creates an event that parses through the contents of the given website. If a url is found and the depth value is not 0 a new event is created, that executes the function on the new website, with the depth value decreased by one.

2.3 Reactive

Source code: TODO ADD THE SOURCE CODE DIR TO BE DONE

2.4 Virtual Threads

Source code: TODO ADD THE SOURCE CODE DIR The function creates a virtual thread that executes the following pseudocode:

```
input: str URL, str WORD, int DEPTH
begin
    str SITE_CONTENTS ← get_site_contents(URL)
    int COUNT ← 0
    if (DEPTH != 0)
        foreach CURRENT_WORD in SITE_CONTENTS
            if (CURRENT_WORD == WORD)
                COUNT ← COUNT + 1
            endif
            else if (is_url(CURRENT_WORD))
                VT = create_new_vt(() ->
                    getWordOccurrences(CURRENT_WORD, WORD, DEPTH - 1))
                join(VT)
        endforeach
    endif
end
```