

Relatório da Tarefa: Software Caching

Victor Wichmann Raposo 9298020

Abril 2016

1 Introdução

Essa tarefa tinha o intuito de comparar a velocidade das diferentes versões propostas, a tabela de símbolos original do livro ST.java, a árvore rubro-negra esquerdista sem caching e a com caching.

Para isso calculei o tempo de execução de cada estrutura com dois arquivos de texto diferentes: tale.txt usado pelo professor em aula e leipzig1M.txt também usado em aula; com vários thresholds diferentes.

Para calcular o tempo de execução usei o comando "time" e levei o valor "real" que ele devolve.

2 Testes

2.1 Minlen = 1

2.1.1 tale.txt

- RedBlackBST com Caching: 0.740s
- RedBlackBST sem Caching: 0.775s
- ST original: 0.540s

2.1.2 leipzig1M.txt

- RedBlackBST com Caching: 37.608s
- RedBlackBST sem Caching: 50.913s
- ST original: 37.530s

2.2 Minlen = 5

2.2.1 tale.txt

- RedBlackBST com Caching: 0.731s

- RedBlackBST sem Caching: 0.745s
- ST original: 0.476s

2.2.2 leipzig1M.txt

- RedBlackBST com Caching: 31.248s
- RedBlackBST sem Caching: 42.878s
- ST original: 28.947s

2.3 Minlen = 10

2.3.1 tale.txt

- RedBlackBST com Caching: 0.660s
- RedBlackBST sem Caching: 0.622s
- ST original: 0.515s

2.3.2 leipzig1M.txt

- RedBlackBST com Caching: 23.079s
- RedBlackBST sem Caching: 24.866s
- ST original: 15.065s

2.4 Minlen = 20

2.4.1 tale.txt

- RedBlackBST com Caching: 0.568s
- RedBlackBST sem Caching: 0.580s
- ST original: 0.431s

2.4.2 leipzig1M.txt

- RedBlackBST com Caching: 20.870s
- RedBlackBST sem Caching: 23.120s
- ST original: 11.825s

2.5 Conclusão

Analisando os testes feitos, foi possível concluir que a tabela de símbolos original (ST.java) é a mais eficiente para todos os casos. O porquê disso é o uso da estrutura TreeMap, que é uma árvore rubro negra porém com algumas modificações para aumentar a eficiência.

Em relação às árvores rubro negras esquerdistas implementadas, com caching e sem caching, concluí que para o tale.txt os desempenhos foram muito parecidos. No entanto, ao usar o leipzig1M.txt notei uma discrepância, a eficiência da estrutura com cache foi substancialmente maior para a maioria dos "thresholds". Essa diferença se deu porque o tale.txt é um texto menor comparado ao leipzig1M.txt, dessa forma haverá menos casos em que o cache será útil deixando os desempenhos mais próximos.