Projeto 3 – WebCrawler Distribuído

André De Marco Toyama

https://github.com/AToyama/MPI_Crawler

INTRODUÇÃO

O objetivo desse projeto foi utilizar o WebCrawler previamente desenvolvido na matéria e criar uma versão distribuida em processos, utilizando OpenMPI. Assim realizamos uma comparação de desempenho entre essa versão e a sequencial utilizada de modelo para a mesma.

Projeto utilizado como base para este: https://github.com/AToyama/WebCrawler

FUNCIONAMENTO

Esse projeto foi feito baseado no e-commerce https://www.submarino.com.br, portanto o uso de qualquer outro site resultara no mau funcionamento do projeto. O código recebe como entrada uma URL apontando alguma categoria de produtos do site, então lista a url de todas as páginas de produtos dessa categoria. Por fim ele faz o download e análise de cada uma das páginas de produto da respectiva categoria.

Na versão distribuida, o programa é divido em processos que realizam tarefas em paralelas e utilizam OpenMPI para comunicarem entre si. Um processo base é responsável por varrer todas as páginas e obter todas as urls dos produtos, ao mesmo tempo que vai obtendo as urls ele já vai enviando para os outros processos irem realizando o download e análise. Então por exemplo, o processo inicial vai listar todas as urls da página 1 e enviar para o processo 1, depois vai listar todas as urls da página 2 e enviar para o processo 2, etc… Como o processo de listar as urls é rápido, os outros processos não precisam ficar esperando muito, então realizam as análises em paralelo.

DADOS

Para a comparação final foram tiradas as seguintes medidas de tempo:

- Tempo para download e análise de cada produto
- Tempo médio para download e análise de produto
- Tempo total ocioso realizando download de páginas
- Tempo Total para rodas o programa

Foram executados os teste em 4 formatos diferentes do programa:

- Sequencial
- Paralelo com 2 processos
- Paralelo com 4 processos
- Paralelo com 8 processos

A Url utilizada para os testes foi a seguinte:

- https://www.submarino.com.br/categoria/automotivo/mini-geladeira-automotiva (114 produtos)

SISTEMA

CPU: Intel Core i5-4210U CPU @ 2.7GHz
GPU: Mesa DRI Intel(R) Haswell Mobile
RAM: 8GB DDR3

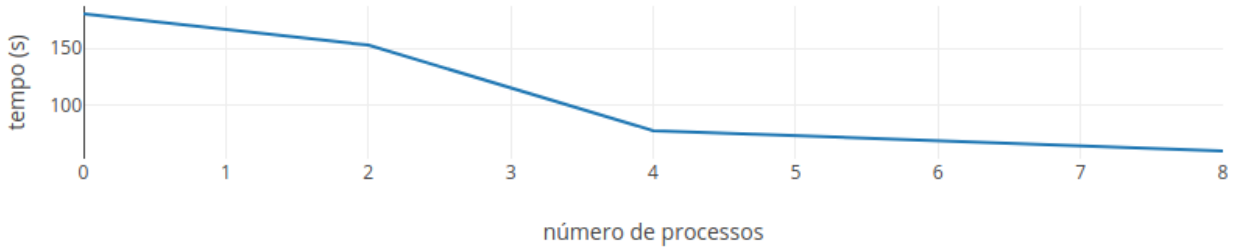
RESULTADOS

Nos gráficos a versão sequencial está representada como 0 processos.

Tempo Médio por Produto:

Tempo Total Ocioso:

Tempo Total do Programa:



No último gráfico já temos a confirmação de que ocorreu o esperado, quanto mais processos trabalhando simultâneamente, mais rápido o programa será executado, porém chega a um ponto que não adianta mais aumentar o número de processos. Pode parecer estranho o resultado se considerarmos que o tempo ocioso total aumentou, porém se pararmos para pensar, o tempo ocioso é o tempo que demora para realizar o download das páginas, então com 8 processos, realizamos o download de 8 páginas simultâneas. Cada página individualmente terá um tempo ocioso maior, pois várias páginas estão sendo baixadas, porém mesmo que o tempo individual seja maior, vários produtos são analisados em um tempo um pouco maior do que um produto só seria analisado no sequencial.

REPRODUTIBILIDADE

No README do repositório do projeto tem as instruções necessárias para rodar e realizar os teste por conta própria, basta acessar o link disponível no começo do relatório..