

Pipeline: Uma Técnica de Programação Paralela

Erick Modesto Campos

¹Instituto de Ciência Exatas e Naturais (ICEN) – Universidade Federal do Pará (UFPA)
Laboratório de Visualização, Interação e Sistemas Inteligentes (LabVis - UFPA)
Cep 66075110 – Belém – Pará – Brazil

erick.c.modesto@gmail.com / erickcampos@ufpa.br

Abstract. *With the evolution of microprocessors, the execution of sequential tasks has become increasingly unfeasible to be applied in scenarios where the issue of performance must be considered. Therefore, parallel programming techniques are essential to extract maximum performance in sequential task tasks. Therefore, parallel programming techniques are essential to extract maximum performance in serial tasks. In this sense, the so-called pipeline technique may be a good alternative for improving performance in applications.*

Resumo. *Com a evolução dos microprocessadores, a execução de tarefas sequenciais foi se tornando cada vez mais inviável de ser aplicado em cenários onde a questão do desempenho deve ser considerado. Por isso, técnicas de programação paralela são imprescindíveis para extrair o máximo de desempenho em tarefas de tarefas sequenciais. Nesse sentido, a técnica denominada de pipeline pode ser uma boa alternativa para a melhora de desempenho em aplicações.*

1. Introdução

Em algumas décadas atrás, era difícil prever que os computadores se tornariam tão populares no cotidiano de bilhões de pessoas. Os primeiros computadores, como o ENIAC (*Electronic Numerical Integrator and Computer*), por exemplo, foram desenvolvidos para realizar funções extremamente específicas que limitavam seu uso. A programação nesses computadores era realizada através de válvulas mecânicas que assumiam valores binários e precisava de uma grande quantidade de pessoas para realizar essa tarefa.

Foi com a chegada dos chamados computadores pessoais (PC, do inglês *personal computer*) que a população começou a ter acesso a essa tecnologia. O primeiro microcomputador surgiu em 1974, desenvolvido pela empresa MITS (*Micro Instrumentation Telemetry Systems*), e possuía um microprocessador de 8 bits operando a 2 MHz que já era capaz de realizar operações mais variadas, se comparado ao ENIAC, por exemplo, que foi desenvolvido para realizar cálculos balísticos.

Conforme os anos foram passando, o poder de processamento dos computadores foram crescendo significativamente. A Tabela 1 extraída de [Young 2016] mostra a evolução do poder de processamento dos microprocessadores.

Tabela 1. Evolução dos microprocessadores.

Processador	Ano	Transistores	Dados	Clock
8080	1974	6.000	8 bits	2 MHz
8085	1976	6.500	8 bits	5 MHz
8086	1978	29.000	16 bits	5 MHz
8088	1979	29.000	8 bits	5 MHz
80286	1982	134.000	16 bits	6 MHz
80386	1985	275.000	32 bits	16 MHz
80486	1989	1.200.000	32 bits	25 MHz
PENTIUM	1993	3.100.000	32/64 bits	60 MHz
PENTIUM II	1997	7.500.000	64 bits	233 MHz
PENTIUM III	1999	9.500.000	64 bits	450 MHz
PENTIUM IV	2000	42.000.000	64 bits	1.5 GHz

Com esse crescente poder computacional dos micropocessadores que possuem multiplos núcleos, a busca em extrair ao máximo a capacidade de processamento, a aplicação de técnicas de programação paralela são necessários nesse cenário. Nesse sentido, a utilização do pipelining é uma boa alternativa para melhorar o desempenho de tarefas seriais.

2. Pipelining

Pipeline é uma técnica bem antiga e bastante conhecida por ser empregada na linha de produção industrial. A ideia por trás dessa técnica é bastante simples. Um processo passa por vários estágios da linha de montagem, subdividindo em tarefas menores, e essa linha de montagem é continuamente alimentada de novos processos. Em cada estágio uma parte de um processo é finalizado e passado para o próximo estágio. Quando é chegado no ultimo estágio, cada subprocesso finalizado é reunido e então finalmente finalizado completamente [Bhujade 1995]. Um bom exemplo da aplicação dessa técnica é na indústria automobilística onde são construídos diversos carros simultaneamente. A Figura 1 mostra a ilustração da aplicação do pipeline na linha de montagem de um carro.

Na computação essa técnica foi também incorporada para o desenvolvimento de programas. Com a grande variedade de processadores multicore, houve a necessidade de aproveitar ao máximo o poder de processamento de cada núcleo do processador. A execução de tarefas sequenciais não é a melhor forma de obter alto desempenho de processamento. Nesse sentido, o pipelining se torna uma técnica alternativa para aumentar o desempenho de tarefas sequenciais de *software*.

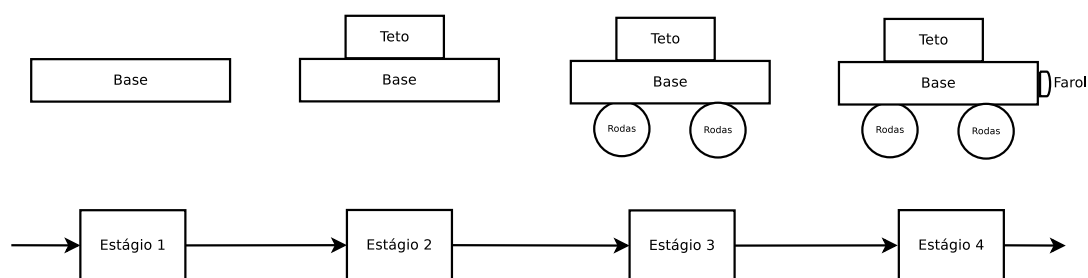


Figura 1. Processo de uma linha de montagem da indústria automobilística.

Ao utilizar a técnica de pipelining, um processo que normalmente seria executado de forma sequencial é dividido em estágios distintos que podem ser executados em um modelo de linha de montagem semelhante ao exemplo mostrado anteriormente, onde cada núcleo do processador fica encarregado de executar uma subtarefa.

Dessa forma, o pipelining melhora o desempenho, pois a subdivisão de uma tarefa em tarefas menores executadas cada uma por um certo processador é mais rápida para ser finalizada que se cada uma das subtarefas fossem executadas apenas por um único processador.

É importante ressaltar que para haver uma melhora significativa no desempenho, é sempre recomendável manter o balanceamento das subtarefas. A Figura 2 mostra a questão do balanceamento.

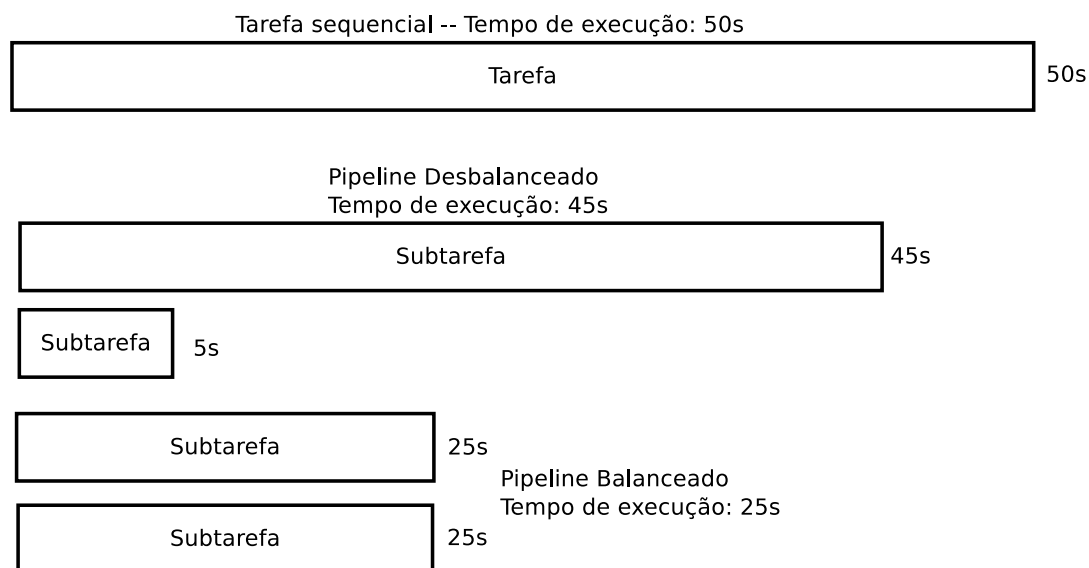


Figura 2. Balanceamento de subtarefas.

O primeiro caso da Figura 2 mostra o tempo de execução de uma tarefa sequencial. Aplicando técnica de pipelining sem levar em conta a questão do balanceamento, como mostra o segundo caso, o tempo para executar tarefa reduz pouco se levarmos em conta o tempo da tarefa sequencial. Contudo, se o pipelining for aplicado com as subtarefas

balanceados, o desempenho de execução melhora consideravelmente, sendo justificável a aplicação dessa técnica para este caso, visto que o tempo de execução é reduzido pela metade.

3. Aplicações de Pipelining

Apesar de simples, o pipelining é uma técnica bastante robusta e muito utilizada em aplicações onde a demanda de desempenho é requerida.

Em [Suleman et al. 2010], por exemplo, os autores propuseram o Feedback-Directed Pipelining (FDP) — pipelining direcionado por feedback —, um *framework* que escolhe a alocação de cada núcleo responsável por executar as subtarefas definidas. O FDP primeiro maximiza o desempenho da carga de trabalho e economiza energia reduzindo o número de núcleos ativos, sem afetar o desempenho. A avaliação em um sistema com dois processadores Core2Quad (totalizando 8 núcleos) mostrou que em média a técnica de pipeline modificada aplicada fornece um *speedup* de 4.2x, uma diferença significativamente maior que o *speedup* de 2.3x obtido através da técnica tradicional de pipeline. Além disso, o trabalho proposto é robusto a mudança de configuração do computador utilizado.

Já em [Xiao and Zhang 2010] é apresentado um modelo de processamento paralelo de imagens digitais baseado na técnica de pipeline. O objetivo do trabalho é atender as necessidades do processamento de imagens de sensoriamento remoto, pois com a crescente quantidade de dados a serem processados, execuções sequenciais dos algoritmos diminuem a eficiência do sensoriamento. A técnica de pipeline utilizada dividia uma imagem digital de $x \times y$ *pixels* em subimagens com uma quantidade menor de *pixel*, mantendo a questão do balanceamento (subimagens de tamanhos semelhantes). Cada subimagem dessa era executada em um determinado núcleo de processamento. Dessa forma, o desempenho do processamento das imagens aumentou significativamente considerando uma grande quantidade de dados (imagens a serem processadas).

4. Conclusão

Como descrito, a técnica denominada de pipeline é bastante simples e robusta que pode ser utilizada em aplicações que demandem melhoria de desempenho na execução dos processos. A questão do balanceamento sempre deve ser levada em consideração para que a aplicação dessa técnica seja eficaz, contudo nem sempre é fácil balancear as subtarefas. Portanto, se uma aplicação necessitar ser executada com um desempenho superior às tarefas seriais, o pipelining pode ser uma boa alternativa a ser aplicada nesse caso.

Referências

- Bhujade, M. (1995). *Parallel Computing*. New Age International (P) Limited.
- Suleman, M. A., Qureshi, M. K., Khubaib, and Patt, Y. N. (2010). Feedback-directed pipeline parallelism. In *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*, PACT '10, pages 147–156, New York, NY, USA. ACM.
- Xiao, Z. and Zhang, B. (2010). Parallel image processing based on pipeline. In *2010 18th International Conference on Geoinformatics*, pages 1–4.
- Young, W. N. (2016). Evolution and advancement of microprocessor - techyv.com. <https://www.techyv.com/article/evolution-and-advancement-microprocessor/>. (Acessado em 09/23/2019).