

”

E-fólio A | Folha de resolução para E-fólio



UNIDADE CURRICULAR: Sistemas Computacionais

CÓDIGO: 21174

DOCENTE: Nelson Russo

A preencher pelo estudante

NOME: Diogo Manuel Miguel Gomes

N.º DE ESTUDANTE: 2305343

CURSO: Engenharia informática

DATA DE ENTREGA: 11 de novembro de 2024

TRABALHO / RESOLUÇÃO:

1) A evolução do desempenho dos computadores, prevista pela Lei de Moore, foi suportada por várias técnicas de otimização em arquitetura de computadores, entre elas o paralelismo, o pipelining e a hierarquia de memórias. Segundo a Lei de Moore, formulada em 1965 por Gordon Moore, cofundador da Intel, o número de transístores em um circuito integrado dobraria aproximadamente a cada dois anos. Esse aumento exponencial na densidade de transístores permitiu que a capacidade de processamento aumentasse significativamente sem que o tamanho físico dos chips precisasse crescer.

Com mais transístores disponíveis, foi possível implementar o paralelismo nos processadores, dividindo as tarefas em subtarefas que podem ser executadas simultaneamente em diferentes núcleos. Essa abordagem permite que múltiplas operações ocorram ao mesmo tempo, o que acelera o processamento e melhora a eficiência. Em vez de executar tarefas uma de cada vez, como ocorre na computação sequencial, o paralelismo usa múltiplos núcleos de processamento, aumentando a velocidade total do sistema. Essa técnica é essencial para lidar com cargas de trabalho complexas e intensivas, como em gráficos e simulações, onde a capacidade de executar várias operações ao mesmo tempo é vital.

Além do paralelismo, o pipelining é outra técnica que contribui para o aumento de desempenho. No pipelining, o processamento de uma tarefa é dividido em várias etapas ou estágios que funcionam em sequência, de forma que várias instruções possam estar em processamento ao mesmo tempo, cada uma em um estágio diferente. Isso permite que o processador trabalhe de forma contínua, aproveitando ao máximo sua capacidade ao minimizar o tempo ocioso entre as operações. Com essa técnica, enquanto uma instrução está em um estágio inicial, outra já está sendo processada nos estágios seguintes, aumentando a eficiência e a taxa de processamento de instruções.

Por fim, a hierarquia de memórias é fundamental para otimizar o acesso aos dados e melhorar o desempenho geral. As memórias são organizadas em níveis, conforme sua velocidade, capacidade e custo, de forma que os dados mais utilizados fiquem armazenados em memórias rápidas e de menor capacidade, como registradores e cache, enquanto os dados menos usados ficam em memórias mais lentas e de maior capacidade, como RAM e discos de armazenamento (HDD/SSD). Isso garante que as operações mais frequentes sejam atendidas com rapidez, otimizando o desempenho e mantendo o custo baixo, já que a memória de alta velocidade é mais cara e limitada.

Assim, a evolução prevista pela Lei de Moore só foi possível com o desenvolvimento de técnicas como o paralelismo, pipelining e a hierarquia de memórias, que ajudaram a aumentar a eficiência e o desempenho dos sistemas sem exigir um aumento proporcional nos custos ou no tamanho físico dos componentes. Essas técnicas maximizam o aproveitamento dos recursos disponíveis, garantindo que o desempenho dos computadores continue a crescer em linha com o aumento do número de transístores, proporcionando dispositivos cada vez mais rápidos e eficientes para os utilizadores.

2) O tempo total de execução do programa é de 200 segundos

Tempo de execução de operações de leitura e escrita de memória: 120 segundos

Tempo gasto em outros tipos de operações: 80 segundos

Para melhorar o desempenho das operações de leitura e escrita de memória, tornando-as 4 vezes mais rápidas, podemos calcular o novo tempo para essa parte:

$$\text{Novo tempo para leitura e escrita} = \frac{120 \text{ segundos}}{4} = 30 \text{ segundos}$$

Tempo de Execução Após a Melhoria:

O novo tempo total de execução do programa será a soma do tempo da operação que foi melhorada e do tempo que não foi afetado:

Tempo de execução após melhoria = Tempo de leitura e escrita após melhoria + Tempo de outras operações

Tempo de execução após melhoria = 30 segundos + 80 segundos = 110 segundos

Resumo - Após a melhoria nas operações de leitura e escrita de memória, o tempo total de execução do programa foi reduzido para 110 segundos. Esse resultado mostra que, embora tenhamos otimizado a parte de leitura e escrita, o desempenho total do programa não melhorou proporcionalmente à melhoria, que era suposto ser 4 vezes mais rápida. Isso deve-se ao fato de que as operações não afetadas, que levam 80 segundos, permanecem inalteradas.

Assim, mesmo que uma parte significativa do tempo de execução tenha sido otimizada, a totalidade do desempenho do programa é limitada pelo tempo das operações que não sofreram melhorias. Portanto, a Lei de Amdahl demonstra bem essa realidade,

evidenciando que melhorias em uma seção do código não resultam em ganhos equivalentes em desempenho geral, especialmente quando outras partes do programa permanecem inalteradas.

Leu Amdahl: O aumento de desempenho possível com uma determinada melhoria no sistema é sempre limitado pelo grau de uso dessa melhoria.

3)

