

Relatório 13 - Bruna Meinberg

Schedules

- Qual *scheduler* apresentou o menor tempo médio?

O *Schedule* que apresentou menor tempo médio foi o **auto**.

- Algun *scheduler* teve variações significativas entre as execuções? Se sim, por quê?

O *scheduler* **default** teve variações significativas durante as execuções, afinal ele é o que tem maior concorrência entre os outros processos que estão rodando no sistema. Quando utilizamos outros schedules, pode-se ter definições de parâmetros que nos ajudam a ter os processos melhor distribuídos e mais otimizados.

- Alguma característica específica do trabalho (como carga de dados, balanceamento) parece ter influenciado o comportamento de um **scheduler** em particular?

A utilização de **dynamic** pareceu bastante benéfica

Cálculo do PI

- Qual abordagem (**parallel for** ou **tasks**) apresentou melhor desempenho?

A abordagem **parallel** foi visivelmente melhor

- O valor de **MIN_BLK** ou o número de tarefas influenciou significativamente o tempo de execução?

O valor de **MIN_BLK** ou o número de tarefas não influenciou significativamente o tempo de execução

- Alguma abordagem teve variação maior entre execuções? Por quê?

No *parallel* não teve nenhuma abordagem que teve maior variação, já no *tasks* teve uma leve variação devido a forma que as *threads* são distribuídas.

Manipulação de Efeitos Colaterais no Vetor

- Qual abordagem teve melhor desempenho: `omp critical` ou pré-alocação de memória?

A abordagem de pré-alocação de memória teve um melhor desempenho

- O uso de `omp critical` adicionou muito *overhead*? Como você pode justificar isso?

Com `omp critical` é adicionado muito *overhead*. Provavelmente o fato do vetor ser compartilhado entre as threads acaba atrapalhando a inserção de valores.

- A ordem dos dados no vetor foi mantida em ambas as abordagens?

Sim, a ordem foi mantida em ambas as abordagens

Conclusão

- Resuma as principais conclusões com base nos resultados obtidos nos testes.
 - O *scheduler default* teve as maiores variações, enquanto *auto* e *dynamic* tiveram melhor desempenho;
 - No cálculo do pi, *parallel* foi a melhor abordagem em comparação com *tasks*;
 - Na manipulação de efeitos colaterais do vetor, a abordagem de pré-alocação desempenhou melhor enquanto *omp critical* causou um *overhead*;
 - Cada caso deve ser analisado particularmente
- Qual abordagem geral você considera mais eficiente para problemas recursivos e com efeitos colaterais?
 - A abordagem mais eficiente para esse caso foi *parallel*.

- Alguma técnica apresentou resultados inesperados? O que poderia explicar isso?
 - No geral, não.