

O uso computacional de processadores com múltiplos núcleos de processamento, também conhecidos como multi-core, demonstra a nova aposta da indústria de semicondutores, que propõe o uso de chips multiprocessados para computação de alto desempenho. A principal vantagem destes processadores é a possibilidade do programador poder dividir tarefas em partes menores a serem executadas em paralelo, melhorando o desempenho do sistema. Nestes processadores de propósito geral, o uso de hierarquia de memória, em especial as memórias cache, é bastante comum, e tem o objetivo de esconder as altas latências que o processador deve enfrentar ao trabalhar com dados que estejam em memórias de grande porte. Além disso, alguns níveis dessa hierarquia de memória cache podem estar compartilhados por dois ou mais núcleos, dependendo do modelo do processador. Porém, este compartilhamento pode prejudicar o desempenho quando cada núcleo opera sobre dados diferentes, pois os dois núcleos estariam concorrendo pelo mesmo recurso, causando conflitos na memória cache. Mapear os processos que utilizam um mesmo subconjunto de dados nos núcleos que compartilham a mesma memória cache faria com que problemas de conflitos e concorrência fossem evitados. Neste trabalho, utilizamos funções MPI e técnicas de afinidade manual para analisar empiricamente a forma com que o sistema operacional Linux escalona os processos entre os núcleos. A técnica consiste em medir o tempo de execução de um benchmark através do mapeamento feito pelo programador e pelo sistema operacional. Após as medições, é feita a comparação dos resultados a fim de definir qual é a estratégia adotada pelo sistema operacional, e quais os possíveis ganhos que outros mapeamentos de processos poderiam trazer ao desempenho final do sistema.