

1) O artigo apresenta os seguintes princípios:

Shared Memory, threads, computação distribuída, MPI, memória distribuída computacionalmente.

2) Devemos estudar a possibilidade se ele pode ser paralelizado, se pode haver dependência de dados entre as threads, caso haja, um mecanismo de controle deve ser adotado para gerenciar o acesso às áreas compartilhadas, garantindo a consistência dos dados. Também é válido estudar se a escalabilidade do processo será otimizada mesmo.

3) Concorrência, falso compartilhamento de dados. Ele pode causar inconsistência de dados no cache, pois uma thread pode alterar o valor de alguma variável ou dado no cache, ao mesmo tempo que outra thread altera o mesmo valor, tornando o mesmo inconsistente e desatualizado.

4) Fazer com que o dado no cache se mantenha atualizado independente das alterações que foram feitas nesse valor pelas outras threads.

Para chegar nesse estado, é necessário implementar um semáforo, que nada mais é que uma maneira de controlar o uso de uma região de memória, ou seja, caso uma thread acesse uma variável/dado, nenhuma outra poderá alterar ou ler o valor enquanto a primeira não terminar sua tarefa.