Sr. Estagiário:

Sobre *#pragma omp parallel for schedule(dynamic, 1)*

*#pragma omp parallel* é a construção fundamental do OpenMP, ele indica a região paralela, que indica um bloco de código que será executado por múltiplas threads. Tudo certo por enquanto.

Como estamos tratando de paralelizar um *for loop* a derivativa *for* utilizada também está correto.

Sobre o cláusula *schedule(dynamic, 1)*, suponho que tenhas testado na sua máquina diferentes valores para *dynamic* e também testado usando *static* e *guided* e percebesse que *schedule(dynamic, 1)* acelerou seu programa. Se for esse o caso, então tudo bem.

Sobre *#pragma omp parallel for firstprivate(i, j) schedule(guided)*

Há muitas threads sendo criadas. No seu caso estás usando paralelismo dentro de paralelismo, ou seja, cada thread do for externo irá criar uma thread no for interno, gerando um grande número de threads que competiriam pelos recursos.

*Firstprivate(i, j)* cria uma cópia local das variáveis em cada *thread*. Essas cópias são inicializadas com o valor que as variáveis possuíam antes da região paralela. Essa cláusula não tem tanta relevância.

O mesmo que foi dito anteriormente sobre a cláusula *schedule* vale para essa situação.

Em resumo, acabasse criando mais threads do que o necessário.

Aliás, na próxima comenta teu código pois facilita o entendimento das pessoas que podem vir a ler seu código. Além disso, pode te ajuda a entender melhor o problema também.

Att.,

O cara que manteve o emprego