

How does the ordered clause in OpenMP work in conjunction with a parallel for loop?

Alle Befehle außerhalb der 'pragma omp ordered' clause werden parallel ausgeführt und alle Befehle innerhalb der Klausel werden sequentiell ausgeführt.

D.h. in einer Anwendung, in der es uns darauf ankommt, in welcher Reihenfolge gerechnet wird, ist die ordered Klausel nützlich; wenn wir jedoch eine Berechnung haben die einfach nur ausgeführt werden soll, dann brauchen wir die ordered Klausel nicht.

What is the collapse clause in OpenMP good for?

Mit collapse kann man geschachtelte Schleifen parallelisieren. Ohne collapse würde nur die äußere Schleife parallel laufen, mit collapse(#) kann eine beliebige # an Schleifen parallelisiert werden.

Explain how reductions work internally in OpenMP.

Eine reduction ist OpenMP's implementation der Reduktionsfunktion aus map/reduce. Es wird eine lokale Kopie von jeder Variable der Eingabeliste für jeden Thread angelegt (und je nach Operator initialisiert). Jeder Thread führt die Operation dann lokal auf seinem Datenblock aus → es entsteht eine Menge an lokalen 'Zwischenergebnissen', die dann zusammengeführt (reduziert) werden.

What is the purpose of a barrier in parallel computing?

Mit einer Barriere kann man ein paralleles Programm synchronisieren, indem die nächste Codezeile erst ausgeführt wird, sobald alle Threads mit ihrer Aufgabe fertig sind. Sehr nützlich z.B. für Operationen auf großen Datensätzen, (1) Wir wollen die Daten lesen oder transformieren und (2) danach eine Operation auf den transformierten Daten ausführen (eine Reduktion z.B.) → Zwischen (1) und (2) sollten wir eine Barriere einbauen.

Explain the differences between the library routines

`omp_get_num_threads()`, `omp_get_num_procs()` and `omp_get_max_threads()`.

`omp_get_num_threads()` sagt uns innerhalb einer parallelen Region, wie viele Threads aktiv sind.

`omp_get_num_procs()`, gibt # der logischen cores zurück (logische cores = #physische cores \* #threads pro phys. core)

`omp_get_max_threads()` wie viele Threads habe ich in einer parallelen Region? (wird außerhalb einer p. Region aufgerufen). Diese Zahl ist  $\neq$  der maximalen Anzahl an Threads (logischen cores), sondern  $\leq$  der Threadzahl, die wir mit `omp_set_num_threads()` definieren können.

Clarify how the storage attributes private and firstprivate differ from each other.

Private: nichtinitialisierte Kopie einer Variable für jeden Thread (die lokalen Kopien haben sehr wahrscheinlich nicht den gleichen Wert wie die globale V., die kopiert wird)

Firstprivate: initialisierte Kopie der Variable (one-to-one copy) für jeden Thread

Do the coding warmup on slide 18. Write in pseudo code how the computation of  $\pi$  can be parallelized with simple threads.