

Explain how divide and conquer algorithms can be parallelized with tasks in OpenMP.

Each division can be handled by a thread by defining a new task for each of the subsets.

(Beispiel: Partition von Daten in 3 Teile → eine neue task für jeden dieser Teile → paralleles berechnen der Zwischenlösungen → 'taskwait' → Zusammenführen der Teillösungen)

Describe some ways to speed up merge sort.

Merge sort ist ein rekursiver Algorithmus, bei dem bei jedem rekursiven Schritt 2 Teilbäume entstehen. Deshalb kann man merge sort optimieren, indem man z.B. ganz am Anfang beim ersten Split eine task für die linke und eine task für die rechte Seite erzeugt oder bei jedem Split 2 neue tasks erzeugt. (Neue tasks zu erzeugen lohnt sich nur bei Teilarrays, die groß genug sind)

Vor der merge Phase muss man dann ein taskwait einbauen.

In der merge Phase kann man multithreaded merging benutzen.

Man kann insertion sort in der ersten Phase verwenden (nur für kleine n).

Man kann einen heap in der zweiten Phase verwenden (nur für große n).

What is the idea behind multithreaded merging?

Problem: Das native merge läuft immer nur auf einem Thread.

Lösungsidee: Partition Algorithmus anwenden

- ➔ Wir partitionieren die Teilarrays abhängig von einem Median Wert x in 3 Teile, $<x$, $=x$, $>x$
- ➔ Für diese drei Mengen kann dann jeweils ein Thread benutzt werden (eigentlich zwei Mengen, da die Gruppe $=x$ nicht sortiert, sondern nur kopiert werden muss).

Ergebnis: Die divide Phase (also die eigentliche Rekursion) ist sehr schnell, weil wir für jedes Arrayelement nur einen Vergleich mit dem Median x brauchen. Die conquer Phase kann auf zwei Threads laufen; dies war vorher nicht möglich.

Do the coding warmup on slide 17.

-

Read What every systems programmer should know about concurrency.

<https://assets.bitbashing.io/papers/concurrency-primer.pdf>

Discuss two things you find particularly interesting.

- ➔ Splitting concurrency tools into blocking and lockless:
 - Habe nie davon gehört/ daran gedacht, dass man nicht unbedingt blocking braucht
- ➔ Memory Orderings:
 - Die verschiedenen Arten von Orderings und deren Bedeutungen/Anwendungen sind sehr interessant