

Paralleles Debugging mit DDT

29.11.2024

Aufgaben

1. Programmparameter angeben

- Über Kommandozeile beim Aufruf
- Im DDT-Startfenster, Feld **Arguments**

```
Usage: ddt [OPTION...] [PROGRAM [PROGRAM_ARGS]]
```

Abbildung 1:

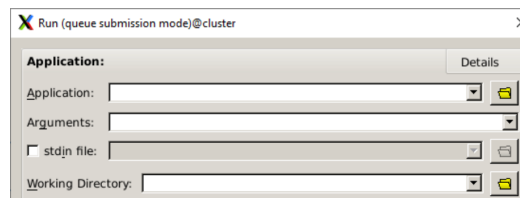


Abbildung 2:

2. Breakpoint setzen & Step-Möglichkeiten

- **Step Into:** Stop vor erster Zeile einer Funktion
- **Step Out:** Stop nach Rückkehr aus aktueller Funktion
- **Step Over:** Führt aktuelle Zeile aus, stoppt bei nächster

3. Variable mit Prozessrang prüfen

- Grafik zeigt Prozess-/Thread-Wertunterschiede
- Vergleich über Rechtsklickmenü wählbar

- Prozesse durch Nummern identifizierbar

```

argv          0x7ffc0389d868
buffer        "|240|071>|360|033|025"
curtime       511101108348
name          ""
result        0
tv
world_rank    0
world_size    4

```

Abbildung 3:

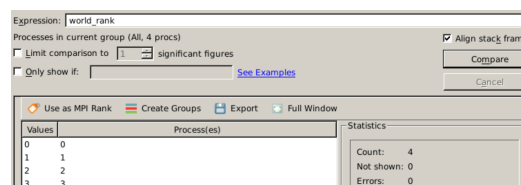


Abbildung 4:

4. Evaluate-Fenster nutzen

- Berechnet Ausdrücke für alle Prozesse
- Vergleich per Striche wie im Locals-Fenster
- Nur gültige Ausdrücke, Fehlermeldungen bei Fehlern

5. Prozess-/Thread-Wechsel beobachten

- Werte im Evaluate-Fenster neu berechnet

```

world_rank    2
world_rank + 100 102

```

Abbildung 5:

```

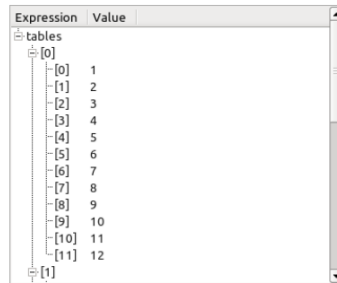
world_rank    3
world_rank + 100 103

```

Abbildung 6:

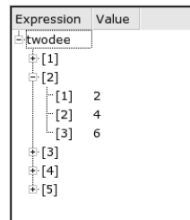
6. Array visualisieren

- Zugriff: global nur über Evaluate, lokal auch via Locals
- Anzeige: Kreuzsymbol oder View Array im Kontextmenü



Expression	Value
tables	
tables[0]	1
tables[1]	2
tables[2]	3
tables[3]	4
tables[4]	5
tables[5]	6
tables[6]	7
tables[7]	8
tables[8]	9
tables[9]	10
tables[10]	11
tables[11]	12

Fig 51: 2D Array In C: type of tables `int[12][12]`



Expression	Value
twodee	
twodee[1]	
twodee[2]	
twodee[2][1]	2
twodee[2][2]	4
twodee[2][3]	6
twodee[3]	
twodee[4]	
twodee[5]	

Fig 52: 2D Array In Fortran: type of twodee is `integer(3,5)`

Abbildung 7: