

1) Wie kann man in DDT die Programmparameter angeben? Gibt es dafür auch andere Möglichkeiten? Wenn ja, welche?

Die Programmparameter können in einem extra Fenster angegeben werden, bevor das Programm ausgeführt wird.

The screenshot shows the 'Run (queue submission mode) (on cluster)' dialog box. The 'Application' field is set to '/home/papenfuss/mpi/timempi2'. The 'Arguments' field is empty and highlighted with a red circle. The 'std in file' field is empty. The 'Working Directory' field is empty. The 'MPI' section is checked, showing '4 processes, 1 node, MPICH 3'. The 'Number of Processes' is 4 and 'Number of Nodes' is 1. The 'Implementation' is MPICH 3. The 'mpiexec.hydra arguments' field is empty. The 'OpenMP', 'CUDA', and 'Memory Debugging' sections are unchecked. The 'Submit to Queue' section is checked, showing 'Wall Clock Limit=00:30:00'. The 'Environment Variables' field is set to 'MPICH_NEMESIS_NETMOD=tcp'. The 'Plugins' field is set to 'none'. The 'Help', 'Options', 'Submit', and 'Cancel' buttons are at the bottom.

Run (queue submission mode) (on cluster)

Application: /home/papenfuss/mpi/timempi2 Details

Application: /home/papenfuss/mpi/timempi2 Folder icon

Arguments: (highlighted with red circle)

☐ **std in file:** Folder icon

Working Directory: Folder icon

☒ **MPI:** 4 processes, 1 node, MPICH 3 Details

Number of Processes: 4 Spinner

Number of Nodes: 1 Spinner

Implementation: MPICH 3 Change...

mpiexec.hydra arguments Dropdown

☐ **OpenMP** Details

☐ **CUDA** Details

☐ **Memory Debugging** Details...

☒ **Submit to Queue:** Wall Clock Limit=00:30:00 Configure... Parameters...

Environment Variables: MPICH_NEMESIS_NETMOD=tcp Details

Plugins: none Details

Help Options Submit Cancel

2) Setzen Sie in einer Zeile einen Breakpoint. Welche Step-Möglichkeiten gibt es und wie unterscheiden sich diese?

Step-Möglichkeiten:

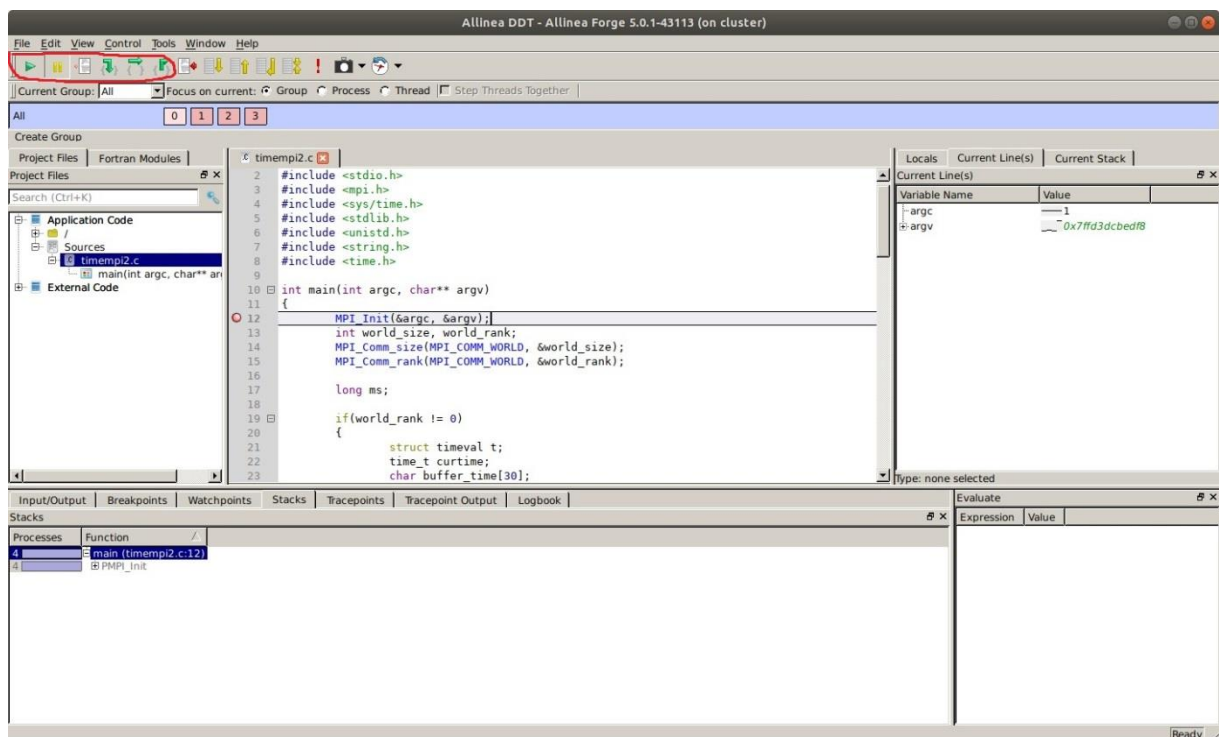
Step-Into: Action um z.B. einen Methodenaufruf zu verfolgen, es wird in die Methode reingesprungen und man kann dann den Methodenaufruf Schritt für Schritt debuggen.

Enthält die aktuelle Zeile kein Aufruf zum reinspringen, verhält sich die Action wie Step-Over und springt zur nächsten Zeile.

Step-Over: Action zum Überspringen der aktuellen Zeile, enthält die Zeile eine Methode wird diese ausgeführt und das Ergebnis ist verfügbar, ohne jede einzelne Zeile debuggen zu müssen.

Step-Out: Zum Rauspringen von einem Step-Into, der Debugger springt zurück an den Zeile wo der Methodenaufruf stattgefunden hat.

Play/Continue: Debuggen Anhalten / Fortfahren

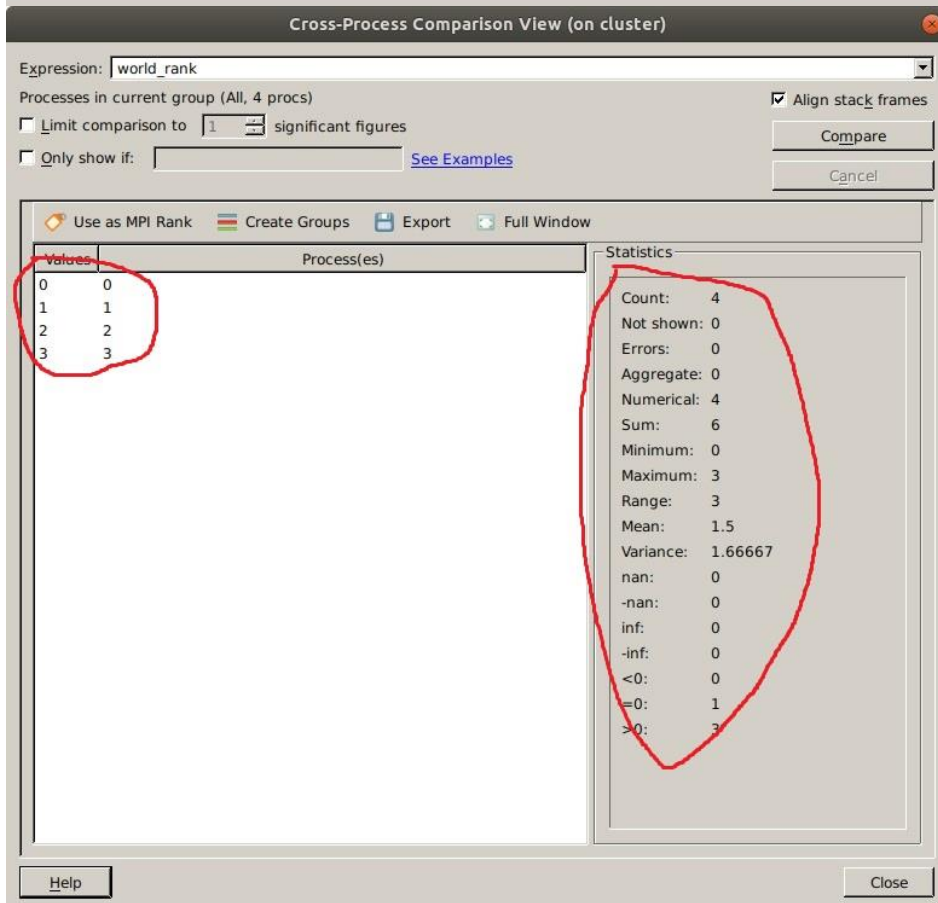
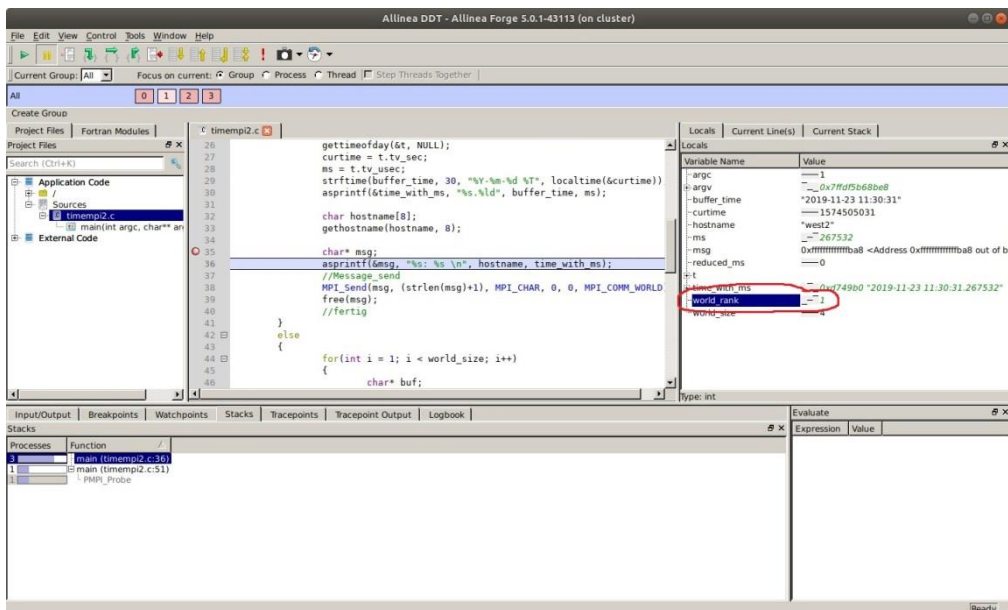


3) Schauen Sie sich die Werte der Variable an, in der Sie den Rang des aktuellen Prozesses gespeichert haben.

Was fällt Ihnen in der Darstellung auf? Vergleichen Sie die Werte aller Prozesse mit Hilfe des Rechtsklickmenüs.

Es fällt auf, dass Zahlenwerte komische Zeichen vor dem eigentlichen Zahlenwert stehen haben.

Vergleicht man jedoch die gesamten Werte der Variable, werden die Integer richtig dargestellt.



4) Machen Sie sich mit der Funktion des Evaluate-Fensters in der rechten unteren Ecke vertraut.

5) In der oberen Leiste finden Sie eine Übersicht aller Prozesse und Threads Ihres Programmes.

Wechseln Sie zwischen den einzelnen Prozessen und beobachten Sie das Evaluate-Fenster.

Die Evaluate-Fenster zeigt die Variablen/Umgebung des aktuellen Prozesses an, befinden wir uns z.B. in Prozess mit Rang 1, so wird dieses angezeigt. Es fällt ebenfalls auf, dass sich die Variablen in dem View Currentline verändern.

The image displays two screenshots of the Allinea DDT - Allinea Forge 5.0.1-43113 (on cluster) interface. The top screenshot shows the 'Evaluate' window with the 'Current Line(s)' tab selected, displaying the current state of the program. The bottom screenshot shows the 'Processes' window with the 'main (timempi2.c:36)' process selected, displaying the current state of the program.

Top Screenshot: Evaluate Window

The 'Evaluate' window shows the current state of the program. The 'Current Line(s)' tab is selected, displaying the current line of code. The 'Locals' tab is also visible, showing the current state of the program's local variables.

Bottom Screenshot: Processes Window

The 'Processes' window shows the current state of the program. The 'main (timempi2.c:36)' process is selected, displaying the current state of the program.