

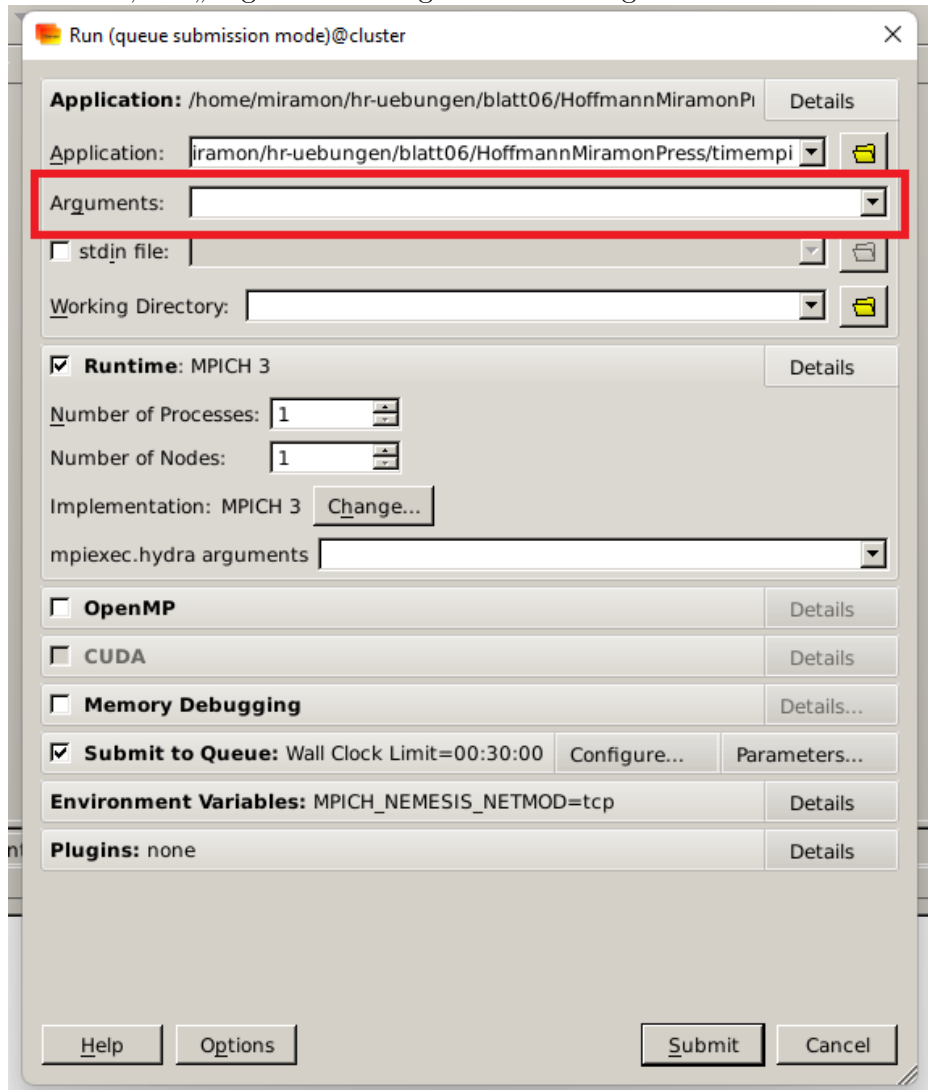
# HLR - Hochleistungsrechnen

## Aufgabenblatt 6

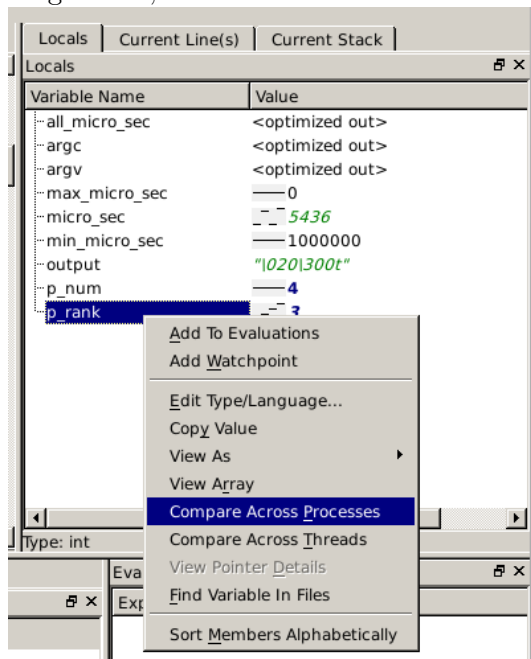
Merle Hoffmann, Joël Miramon, Max Press

### 3. Paralleles Debugging mit DDT:

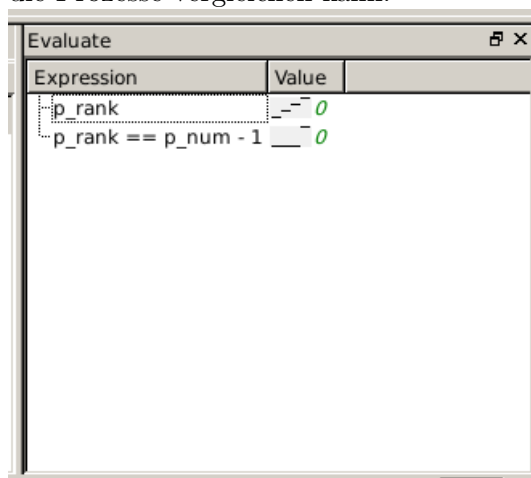
- Entweder man startet das Programm direkt von der Kommandozeile mit den gewünschten Argumenten mit `ddt -n <Anzahl Prozesse> -start <executable> <Argumente>` (z.B.: `ddt -n 4 -start ./timempi <Argumente>`), oder man startet den Debugger mit `ddt <executable>` und gibt danach in dem Fenster, das auftaucht, bei „Arguments“ die gewünschten Argumente an:



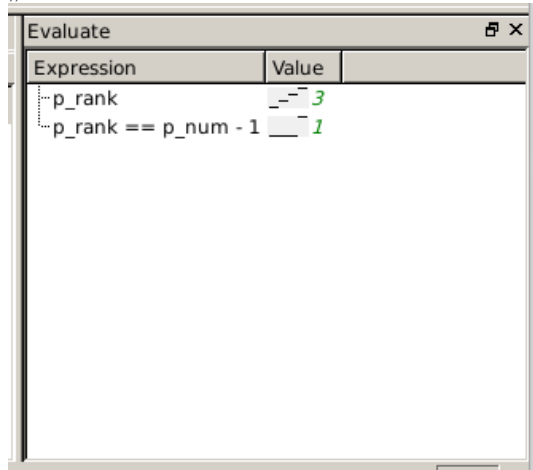
- Es gibt die Step-Möglichkeiten
  - Step Into: Das Programm läuft bis in die Funktion hinein, die in der aktuellen Zeile aufgerufen wird.
  - Step Over: Das Programm läuft über den aktuellen Funktionsaufruf bis zur nächsten Zeile.
  - Step Out: Das Programm läuft bis die aktuelle Funktion fertig gelaufen ist, also zu der Zeile, vor der die aktuelle Funktion aufgerufen wurde.
- Der Rang der Prozesse ist gleich der Prozesszahl, das heißt Prozess 0 hat Rang 0, Prozess 1 hat Rang 1 usw. Die Linien sind eine Art Diagramm, das graphisch die Werte in den Prozessen anzeigt. Die Höhe der Linie gibt den Wert wider und die horizontale Position auf der Linie steht für den Prozess. Man kann die Werte vergleichen, indem man im kontextmenü „Compare Across Threads “ klickt:



- Im „Evaluate“-Menü kann man Ausdrücke eingeben, deren Werte man dann über die Prozesse vergleichen kann:



- Ändert man den Prozess, dann ändern sich dementsprechend auch die Werte im „Evaluate“-Menü:



- Wir initialisieren `array` als ein zweidimensionales 10x10 int-Array, dessen Werte jeweils das Produkt der Indizes ist. Im Kotextmenü kann man sich mit „View Array“ das Array als eine Wertetabelle ausgeben lassen, mit dem Button „Visualize“ kann man eine hübsche Visualisierung bekommen, die man, wenn man will, auch mit anderen Prozessen vergleichen kann.

