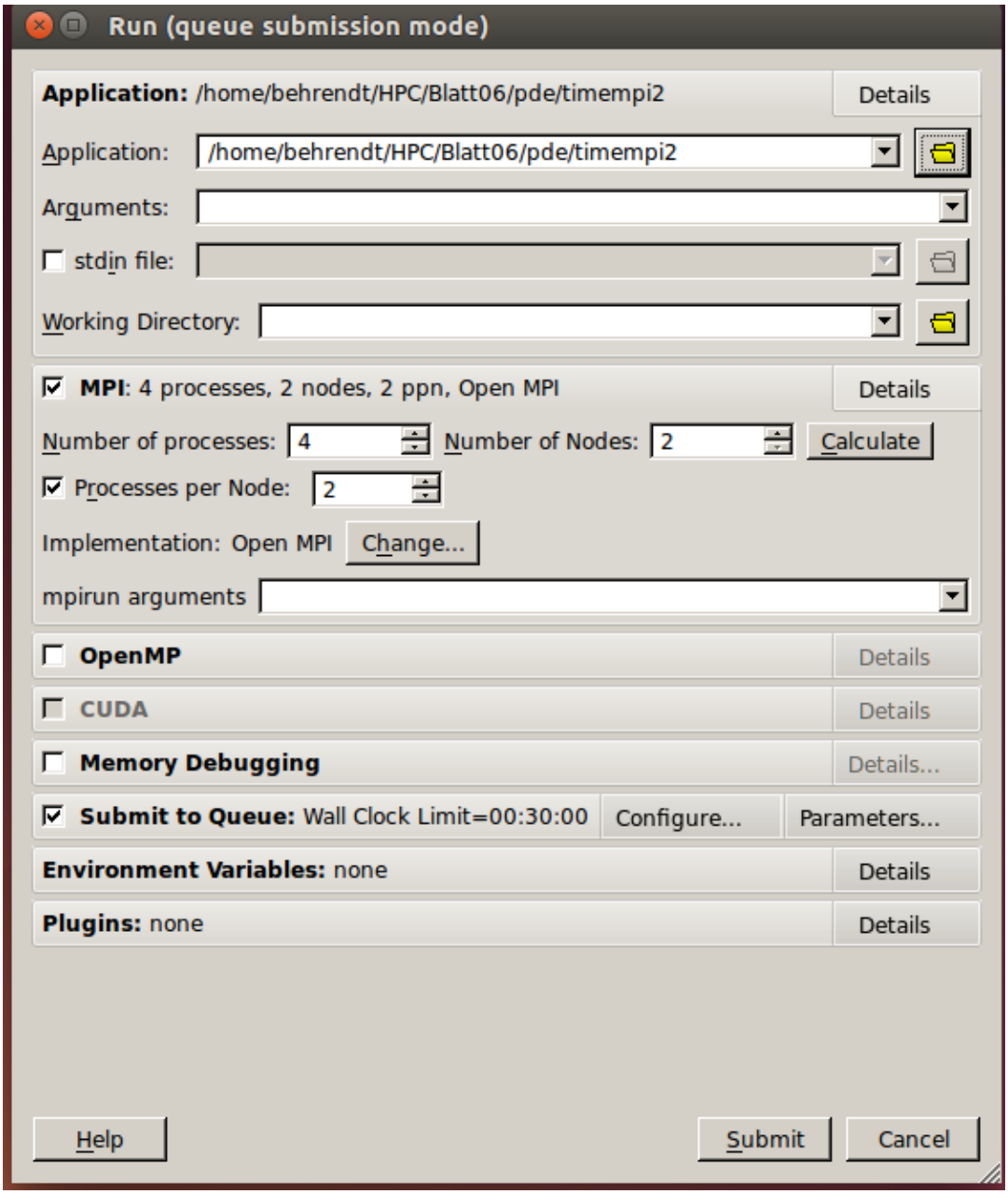
Benutzung von ddt

Start

Nach dem Programmstart kann per „Run“ ein Programm ausgewählt werden und die nötigen Parameter für das Programm angegeben werden. In unserem Fall soll das Programm timempi2 auf 2 Nodes mit 2 Prozessen pro Node laufen:

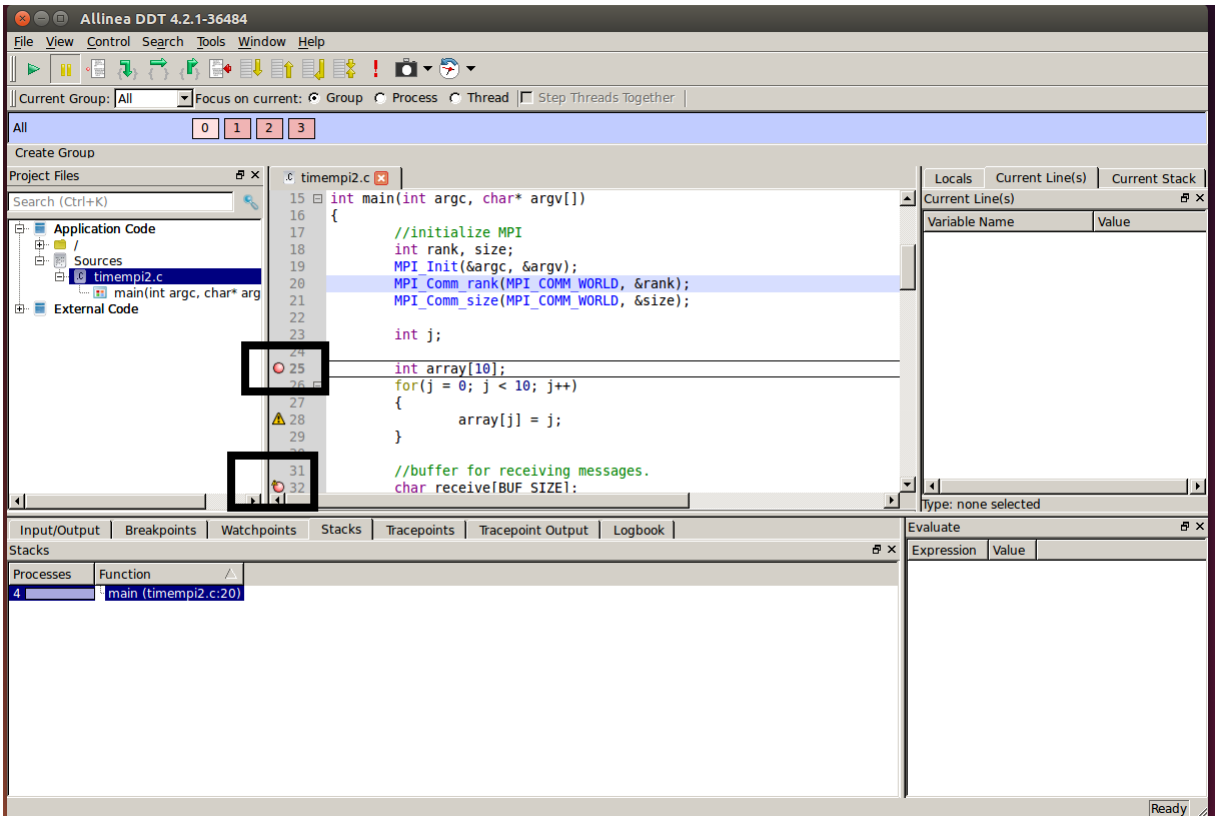


Damit das Programm sinnvoll beobachtet werden kann, muss es mit der Option –g kompiliert worden sein.

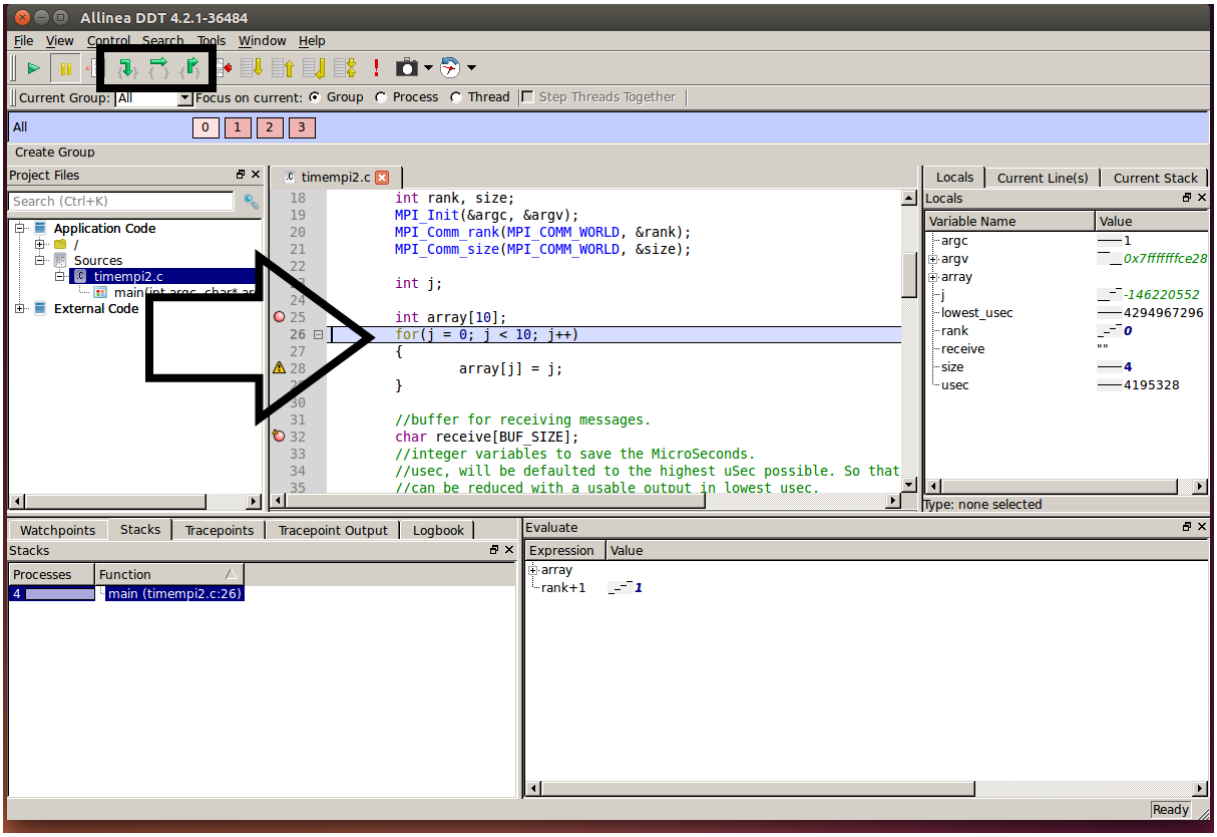
Optional kann dem Programm ddt per Kommandozeile das zu untersuchende Programm mit samt Parametern mitgeteilt werden. Auch Breakpoints können so gesetzten. An dieser Stelle sei an den Aufruf ddt –help verwiesen.

Breakpoint

Nun gibt es die Möglichkeit per Klick neben die Zeilenzahlen der geöffneten Datei Breakpoints zu setzten, die sich mit roten Punkten visualisieren. In diesem Beispiel gibt es 2 Breakpoints, einen in Zeile 23 und einen in Zeile 29.



Wenn das Programm nun gestartet wird, hält es am ersten Breakpoint an, sobald es den erreicht und wir bestätigen, dass dort angehalten werden soll. Die Oberfläche sieht dann wie folgt aus. Mit dem Pfeil ist markiert, wo sich das Programm zurzeit im Quelltext befindet.

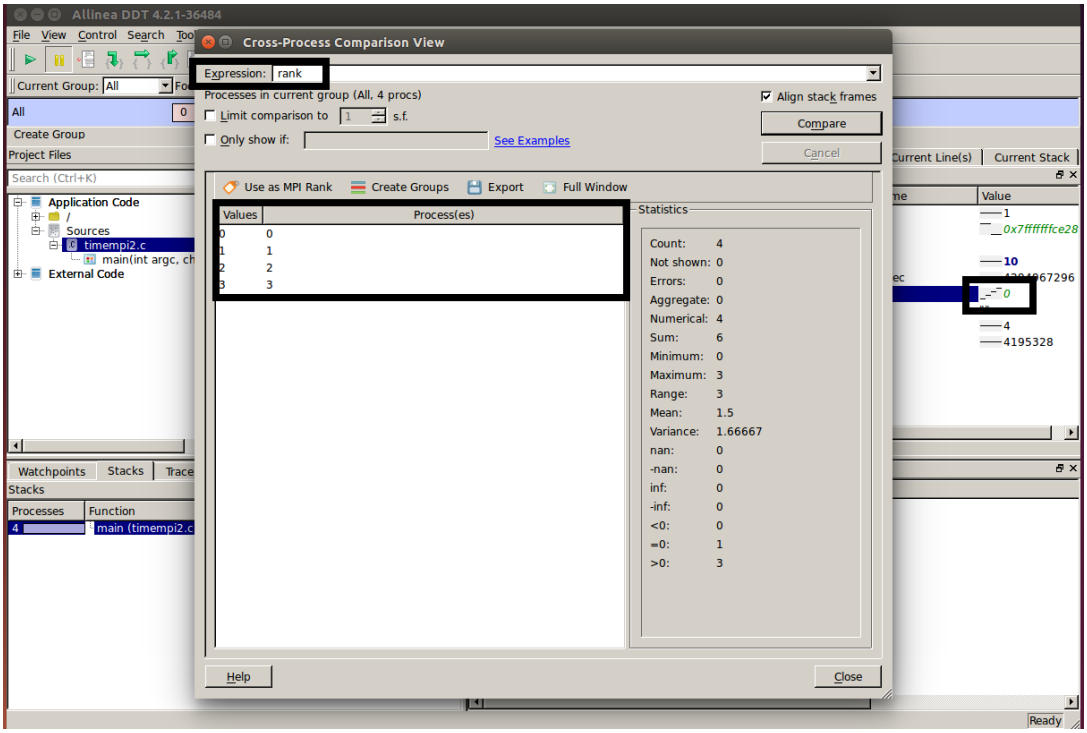


In der Box oben sind die „Step“-Möglichkeiten hervorgehoben. Diese sind die Standard „Step-Into“, „Step-Over“ und „Step-Out“ Funktionen mit denen noch tiefer, höher oder auf gleicher Ebene im Quelltext weitergelaufen werden kann.

Variable-Inspector

Im Fester rechts werden die lokalen Variablen angezeigt. Es fällt auf, dass vor dem Wert noch eine kleine Grafik[1] auftaucht, welche visualisiert, ob der Wert in den einzelnen Prozessen unterschiedlich ist. Etwas genauer schauen wir uns jetzt den rank an, der bereits in Zeile 20 initialisiert wurde (rechter Mausklick auf rank -> Compare Processes).

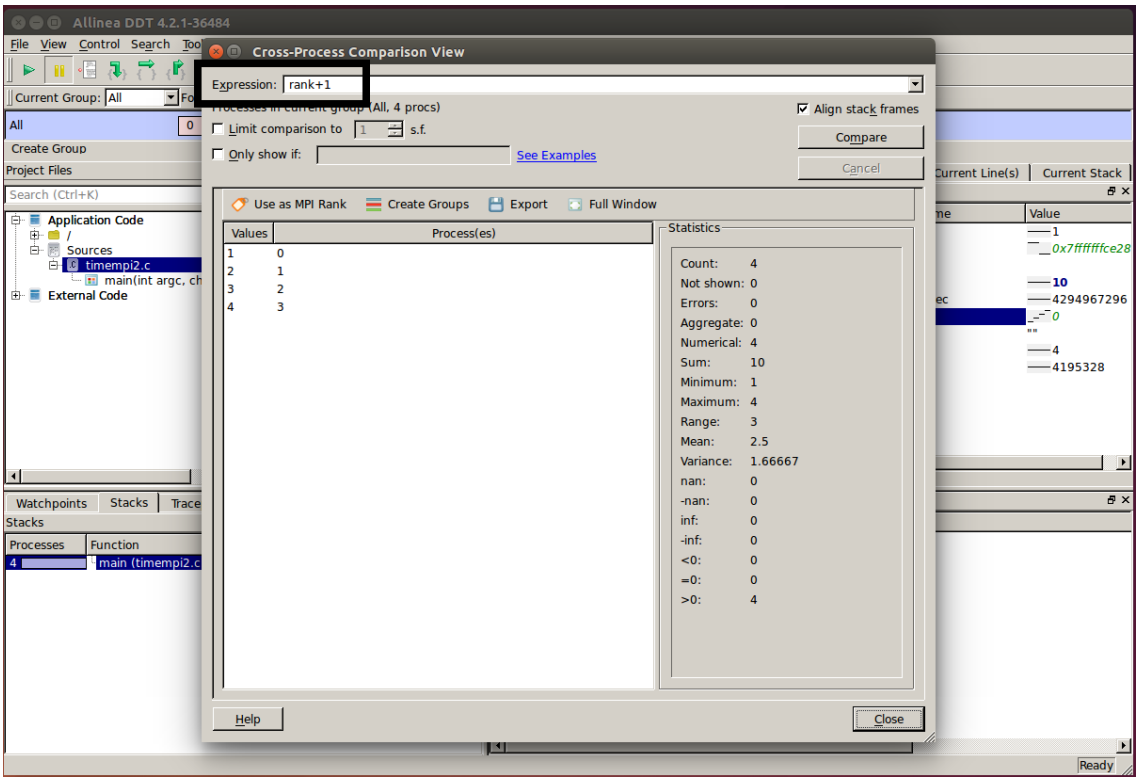
[1] Das graphische Icon zeichnet sich durch Linien aus. Wenn der Wert in allen Prozessen gleich ist, so existiert nur eine horizontale Linie. Bei unterschiedlichen Werten in den einzelnen Prozessen, so existiert eine obere und untere Linie.



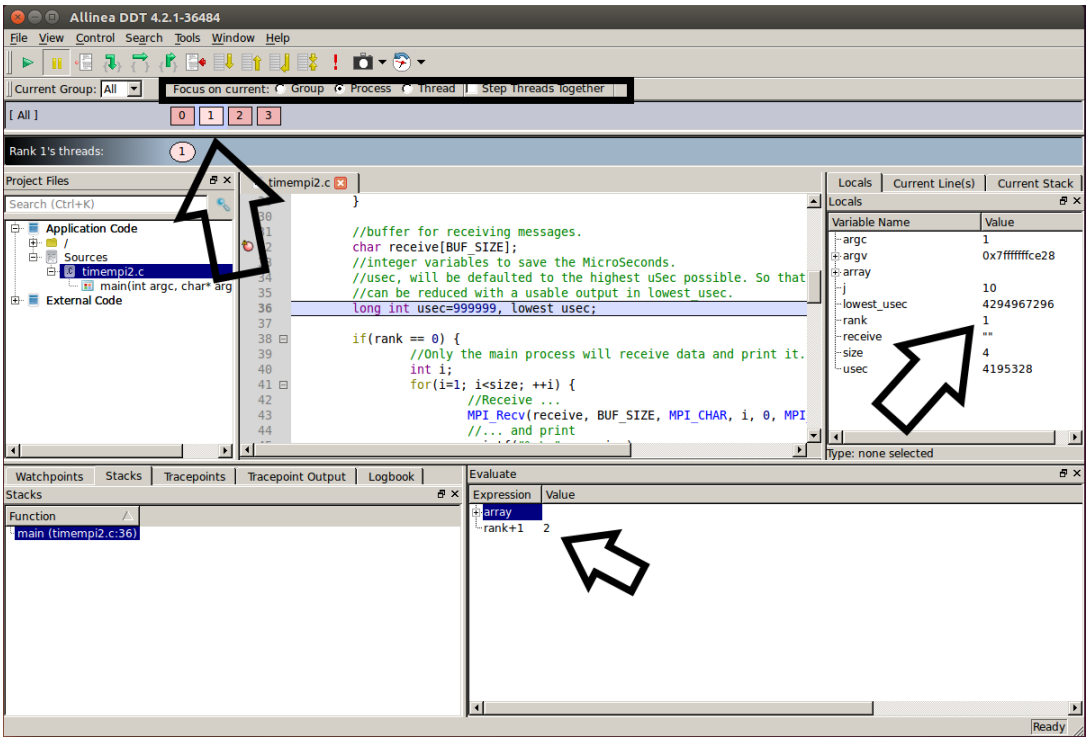
Dort zeigt sich im großen Fenster, dass im Prozess 0, der Rank 0 ist und im Prozess 1 der Rank 1 ist. Somit lassen sich hiermit die Variablen über verschiedene Prozesse (und auch Threads) beobachten.

Evaluate

Das Evaluate-Fenster unten rechts, erlaubt es dynamisch zur Laufzeit Ausdrücke auszuwerten unter Beachtung der aktuellen Variablenbelegung. Ein ganz einfaches Beispiel ist der Ausdruck rank+1, der sich ebenfalls per Variable-Inspector betrachten lässt.



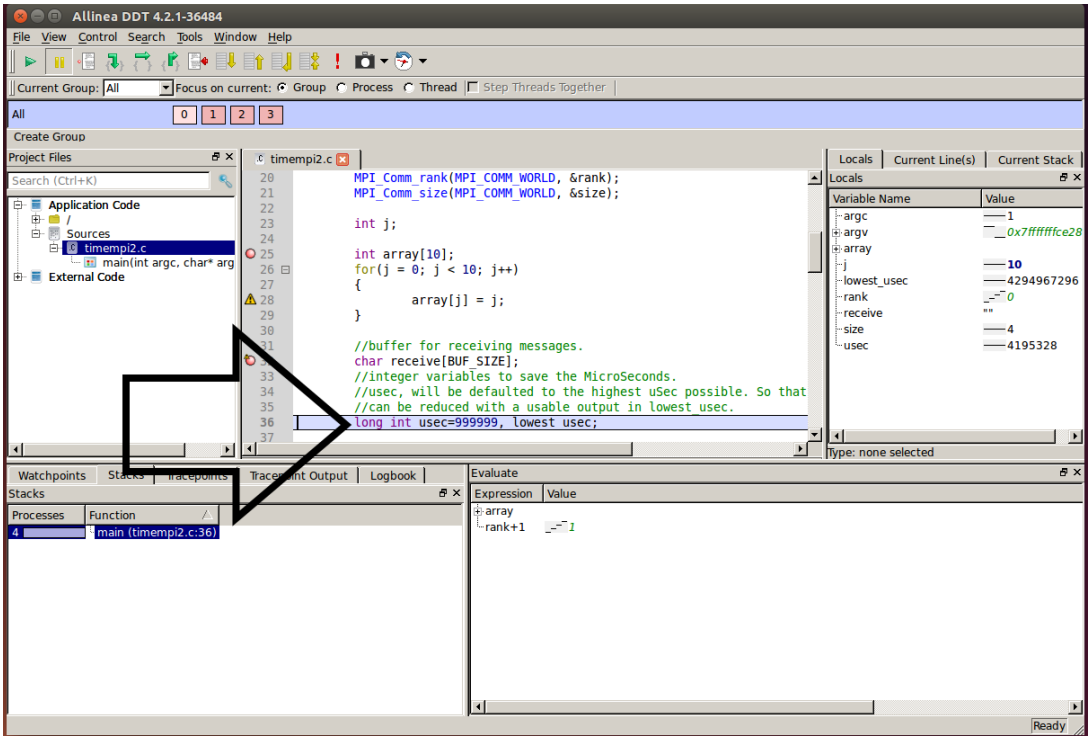
Evaluate in einem bestimmten Prozess



Über die Toolbar lassen sich auch einzelne Prozesse (jetzt Prozessnummer 1) beobachten. Dadurch verändert sich der Variable-Inspector und das Evaluate-Fenster. Diese zeigen nun die Werte für diesen einen Prozess an und nicht mehr für alle. Dadurch verschwindet das kleine grafische Icon und der Wert wird exakt für diesen Prozess angezeigt (Rank ist im Prozess 1 gleich 1. Somit ist der Ausdruck rank+1 gleich 2).

Nächster Breakpoint

Nun lief das Programm weiter und das Array „array“ wurde mit Werten belegt. Im Variable-Inspector werden die einzelnen Werte angezeigt, aber es gibt auch noch eine 2. Ansicht.



View Array

Per Rechtsklick auf „array“ im Variable-Inspector und dann „View Array“ lässt sich das Array wie folgt betrachten. Es muss allerdings die Array Expression erst ausgewertet werden.