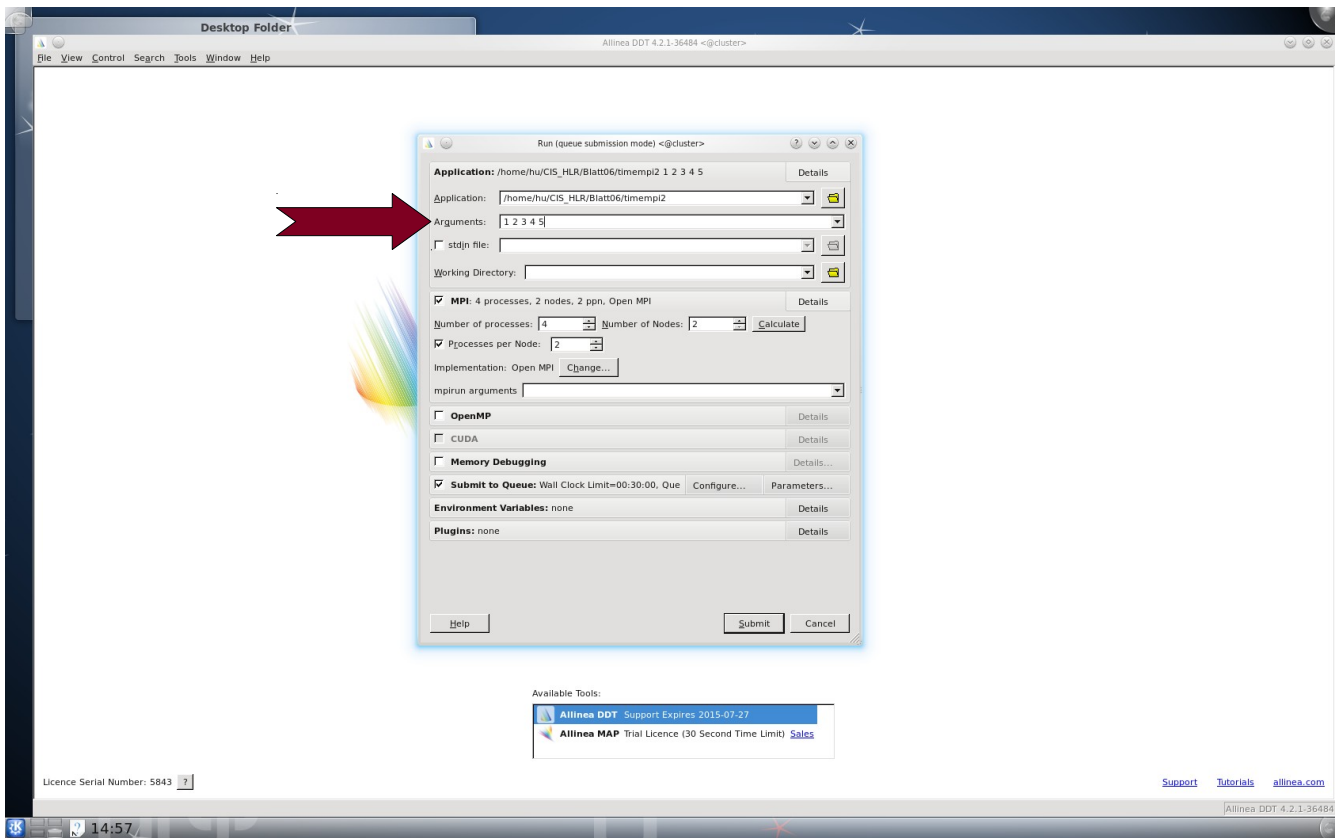
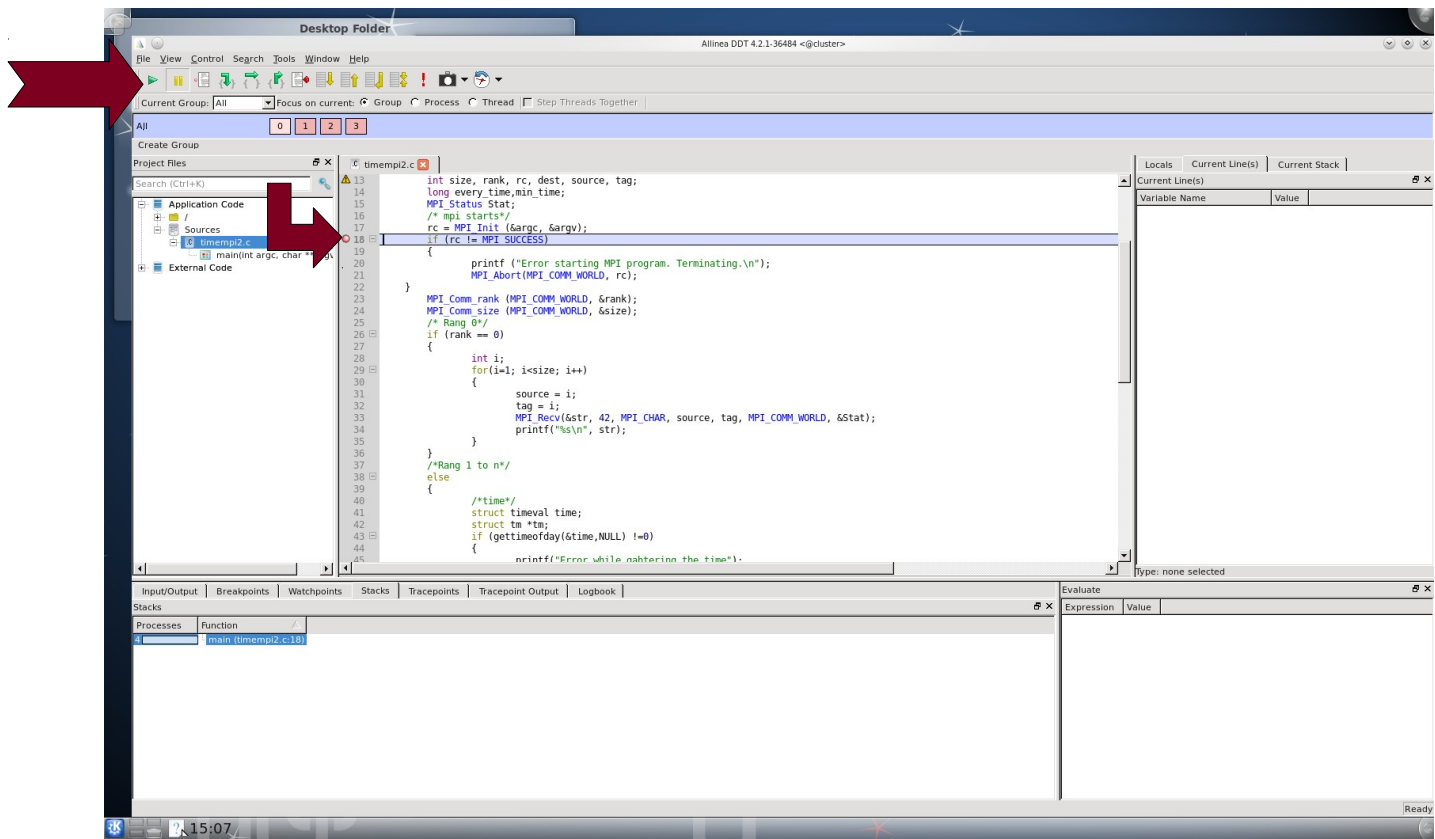


1. Es gibt zwei Möglichkeiten dem in DDT aufgerufenen Programm Argumente zu übergeben: in DDT Argument in der GUI oder in der Shell beim Aufruf von DDT z.B. mit `ddt ./timempi2 1 2 3 4 5`

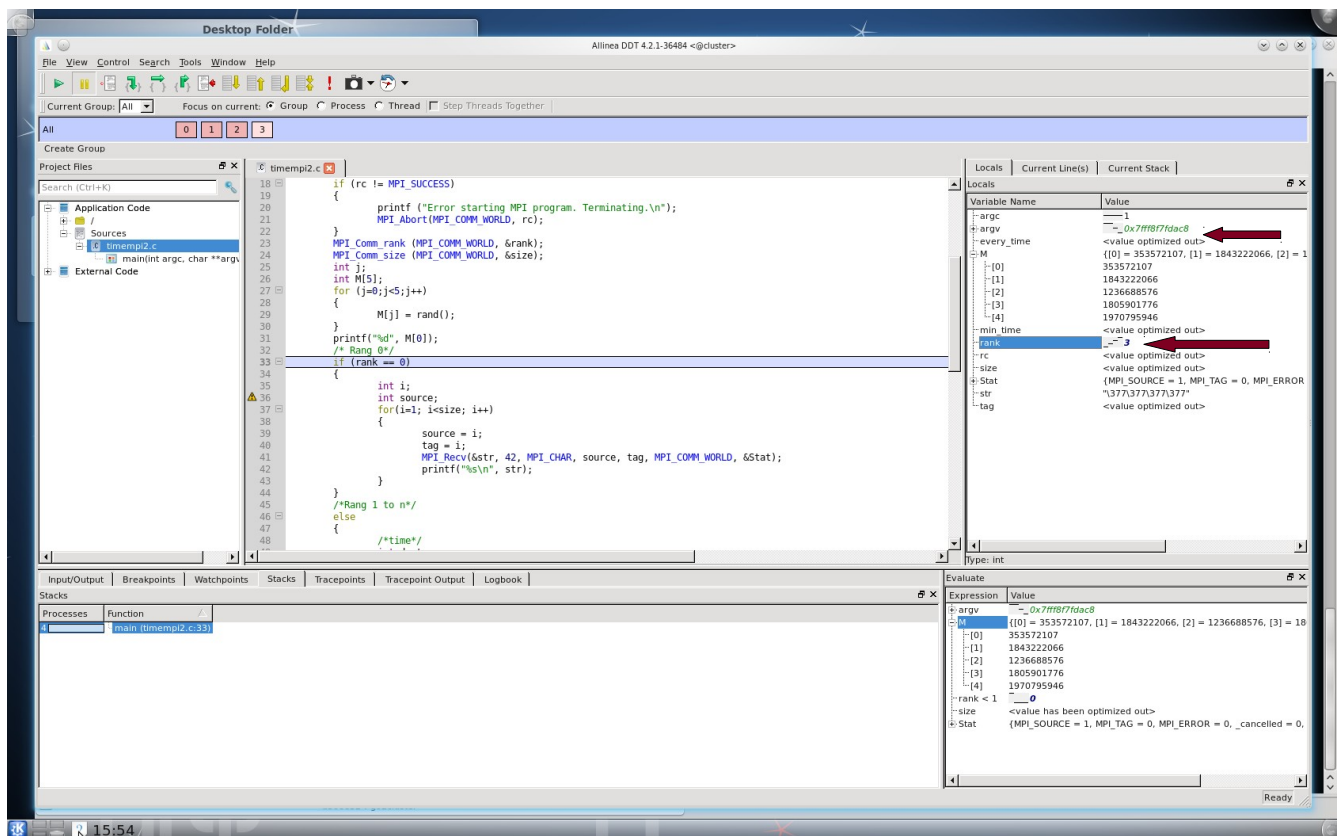


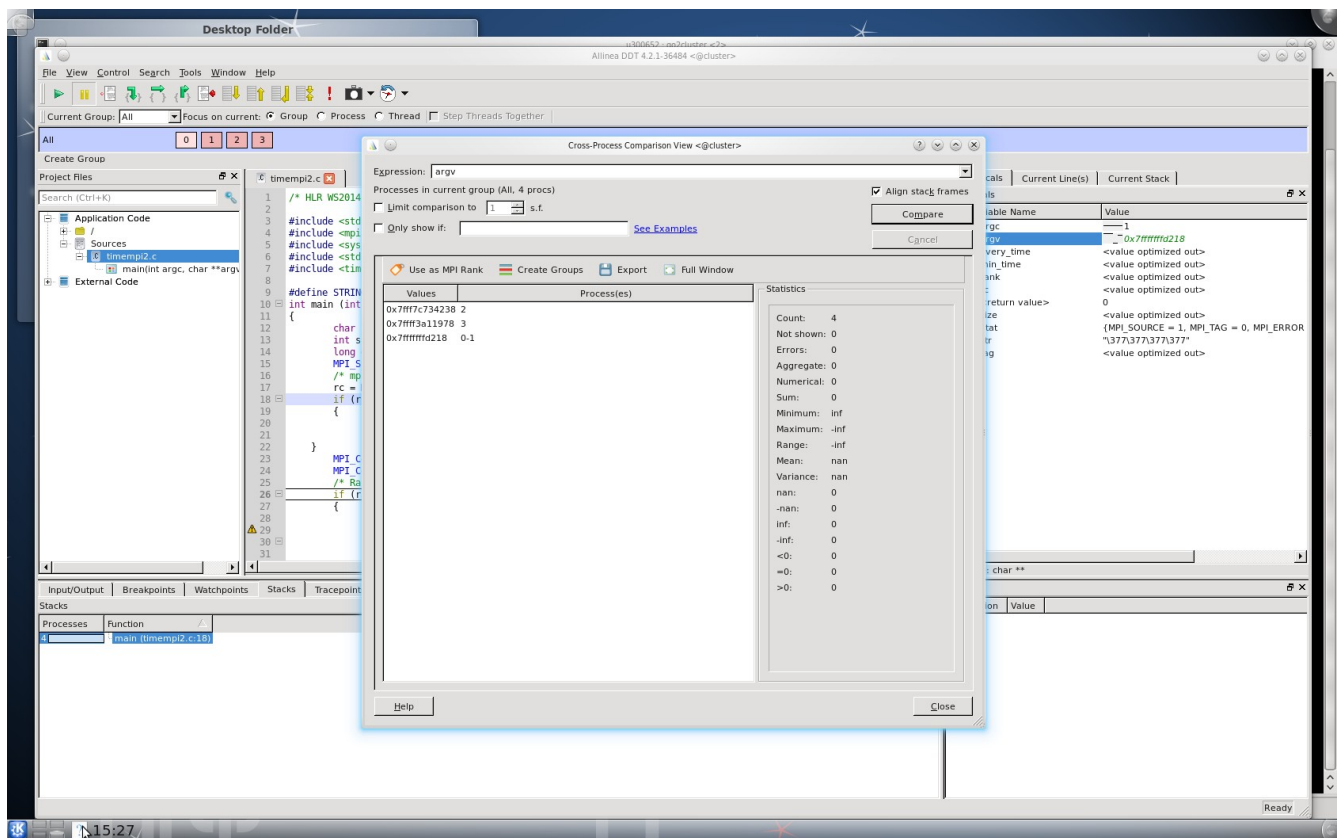
2. Mit einem Linksklick vor links vor die Zeilennummer kann ein Breakpoint gesetzt werden. Es



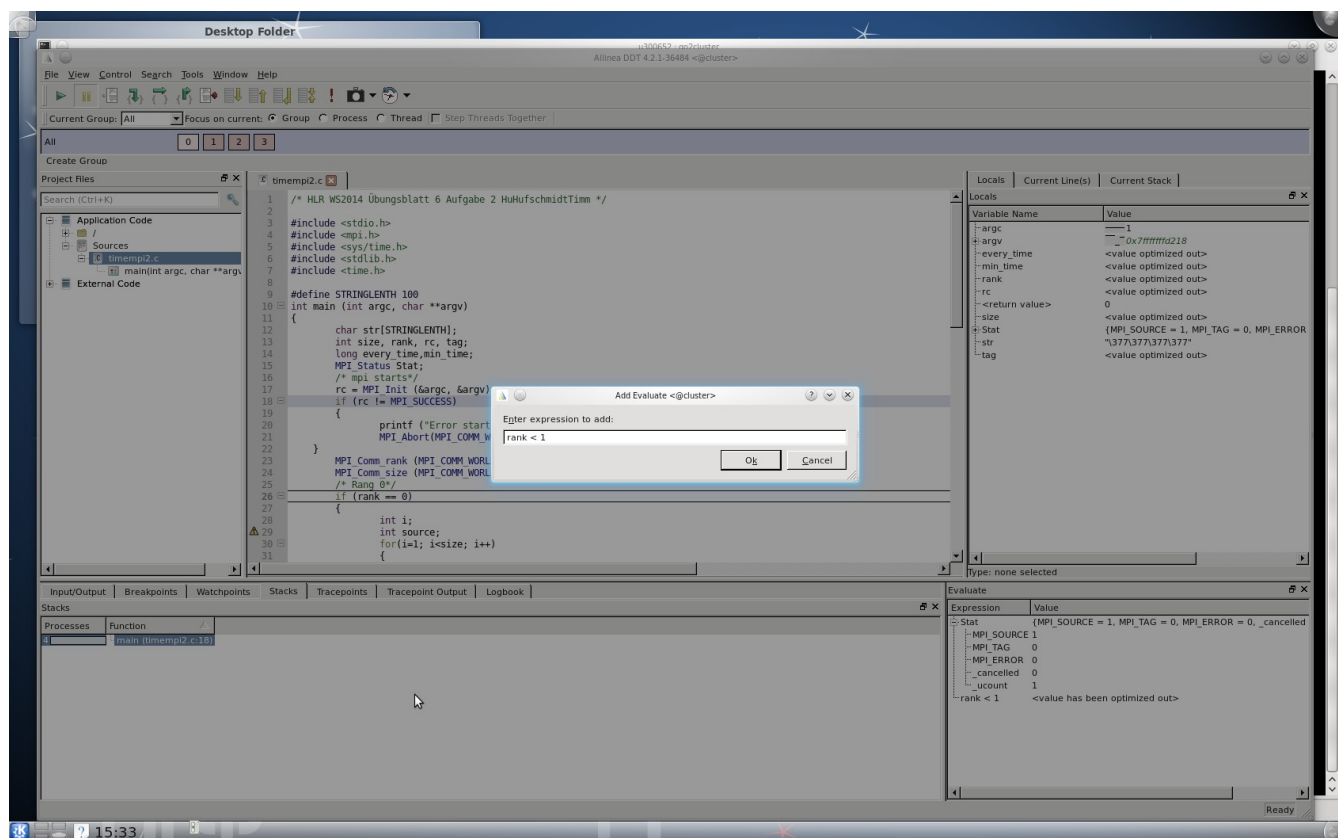
gibt die 3 Möglichkeiten des Steppings: Step into; Step out; Step over. Bei der Option Step into wird in die nächste Zeile gesprungen. (Bei Funktionsaufrufen in die erste Zeile der Funktion). Step Over springt in die nächste Zeile. Funktionsaufrufe werden ausgewertet und in die nächste Zeile gesprungen. Bei Step Out wird der Restliche Teil der Funktion ausgeführt und in die nächste Zeile der Aufrufenden Funktion gesprungen. Ein Step Out aus der Main() Funktion beendet das Programm.

- Bei der Anzeige von Variablen die für alle Prozesse unterschiedlich sind gib es eine kleine Grafische Anzeige mit Stufen. Bei Auswahl eines bestimmten Prozesses wird der eigene Wert blau dargestellt. Beim darstellen mit dem Rechtsklick-Menü wird eine Liste der Werte mit den zugehörigen Prozessen ausgegeben.



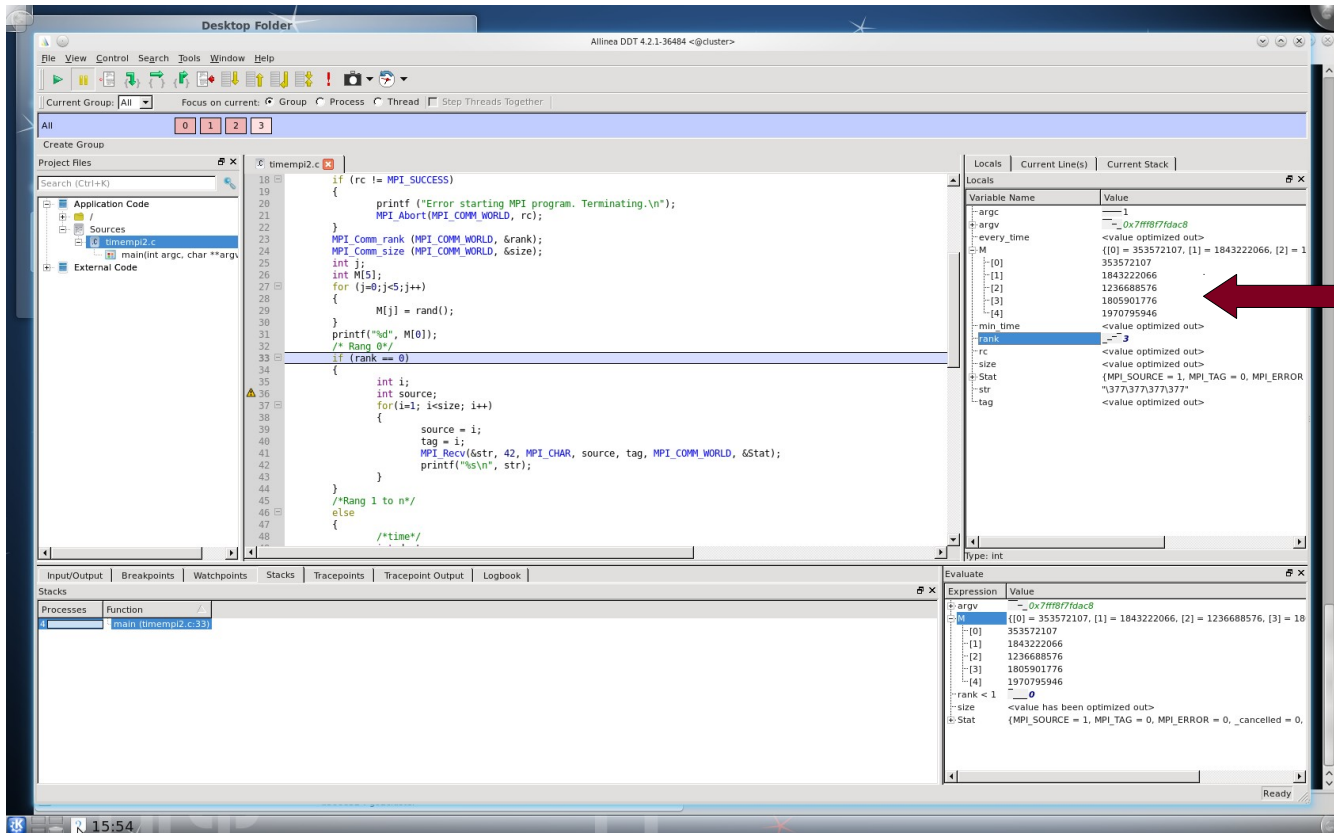


4. Das Evaluate Fenster in der Rechten unteren Ecke wertet Ausdrücke aus. Dies können



Variablen/Matrizen oder logische Ausdrücke sein. Richtige Klammerung ist dabei zu beachten. Durch Benutzung des Menüpunktes „Add Expression“ kann ein solcher Ausdruck hinzugefügt werden.

6. Eine Matrix wird in Form einer ausklappbaren Baum-Struktur dargestellt dabei werden die



einzelnen Werte ausgegeben. Mit dem MDA (Multi Dimensional Array Viewer) können Arrays von DDT auch grafisch ausgegeben werden. Dieser Viewer hat jedoch verschiedene Einschränkungen.

Allinea DDT 4.2.1-36484

File View Control Search Tools Window Help

Current Group: All Focus on current: Group Process Thread Step Threads Together

Create Group

Project Files

Search (Ctrl+K)

Application Code

Sources

timempi2.c

timempi2.c

```
10 #ifndef MODE
11 #define MODE 0 // 0 = sequentiell, 1 = Matrix
12 #endif
13 int main (int argc, char **argv)
14 {
15     char str[4096];
16     // MPI program Termination
```

Locals

Current Line(s)

Current Stack

Variable Name Value

-argc 1

-argv 0x7fffffd228

-dest <value optimized out>

-every\_time <value optimized out>

-i <value optimized out>

-min\_time <value optimized out>

-random {[0] = 0, [1] = 0, [2] = 0, [3] = 0, [4] = 0}

-[0] 0

-[1] 0

-[2] 46

-[3] 32767

-[4] -12000

Multi-Dimensional Array Viewer

Array Expression: random[\$i]

Distributed Array Dimensions: None [How do I view distributed arrays?](#)

Range of \$i

From: 0 To: 99 Display: Rows

Align Stack Frames

Auto-update

Only show if: [See Example](#)

Data Table

Statistics

Goto Visualize Export Full Window

0	0
1	0
2	46
3	32767
4	-12000
5	32767
6	-1
7	0
8	-146273612
9	32767
10	-146273112
11	32767
12	-12048
13	32767

Visualization

File View Viewpoint

Process 0

Help Close