

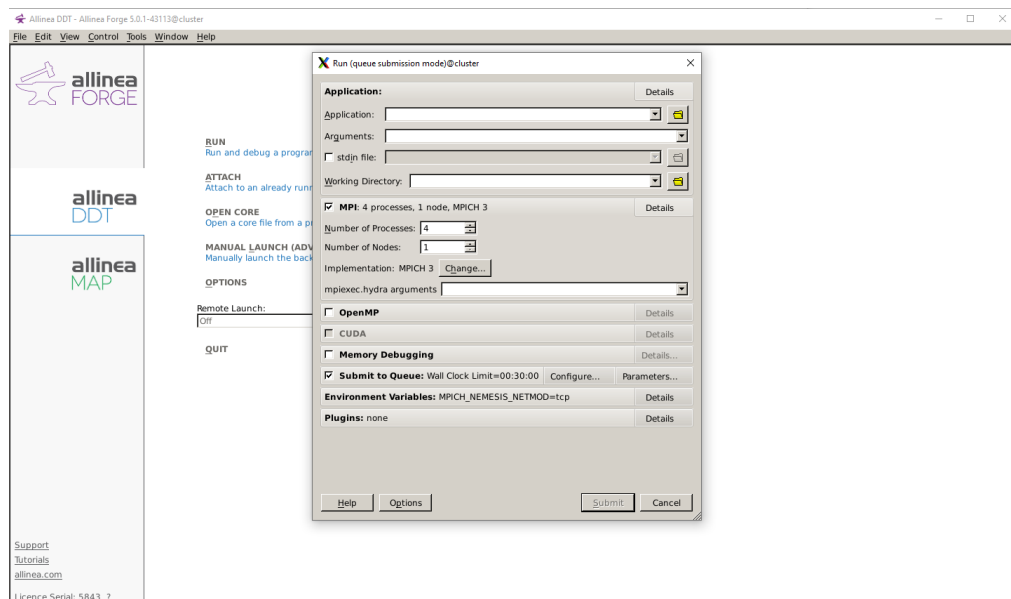
Aufgabe 3: Paralleles Debugging mit DDT

16.11.2019

1. Wie kann man in DDT die Programmparameter angeben? Gibt es dafür auch andere Möglichkeiten? Wenn ja, welche? Eine Möglichkeit Programmparameter anzugeben, ist das Programm mit seinen Parametern beim Aufruf über die Kommandozeile zu übergeben:

```
Usage: ddt [OPTION...] [PROGRAM [PROGRAM_ARGS]]
       ddt [OPTION...] (mpirun|mpiexec|aprun|...) [MPI_ARGS] PROGRAM [PROGRAM_ARGS]
```

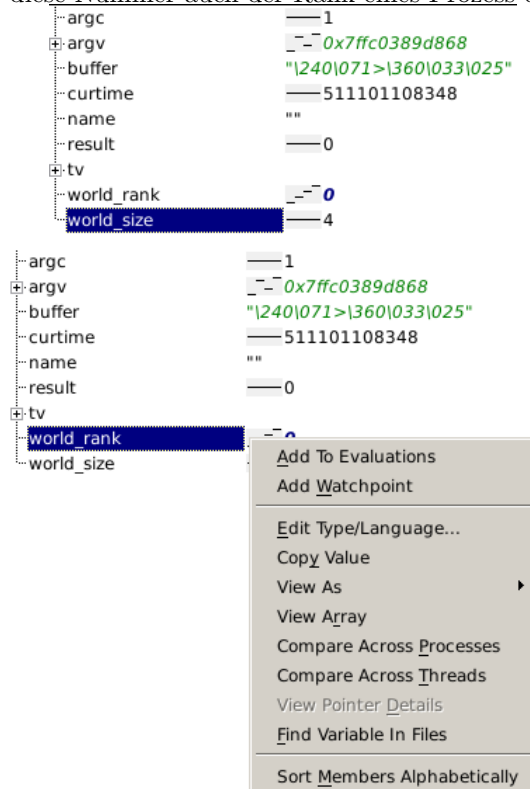
Beim Starten der GUI von ddt kann auch im Fenster vor dem Starten des Programmes verschiedene Parameter, darunter auch die Programmparameter im Feld 'Arguments', angegeben werden:

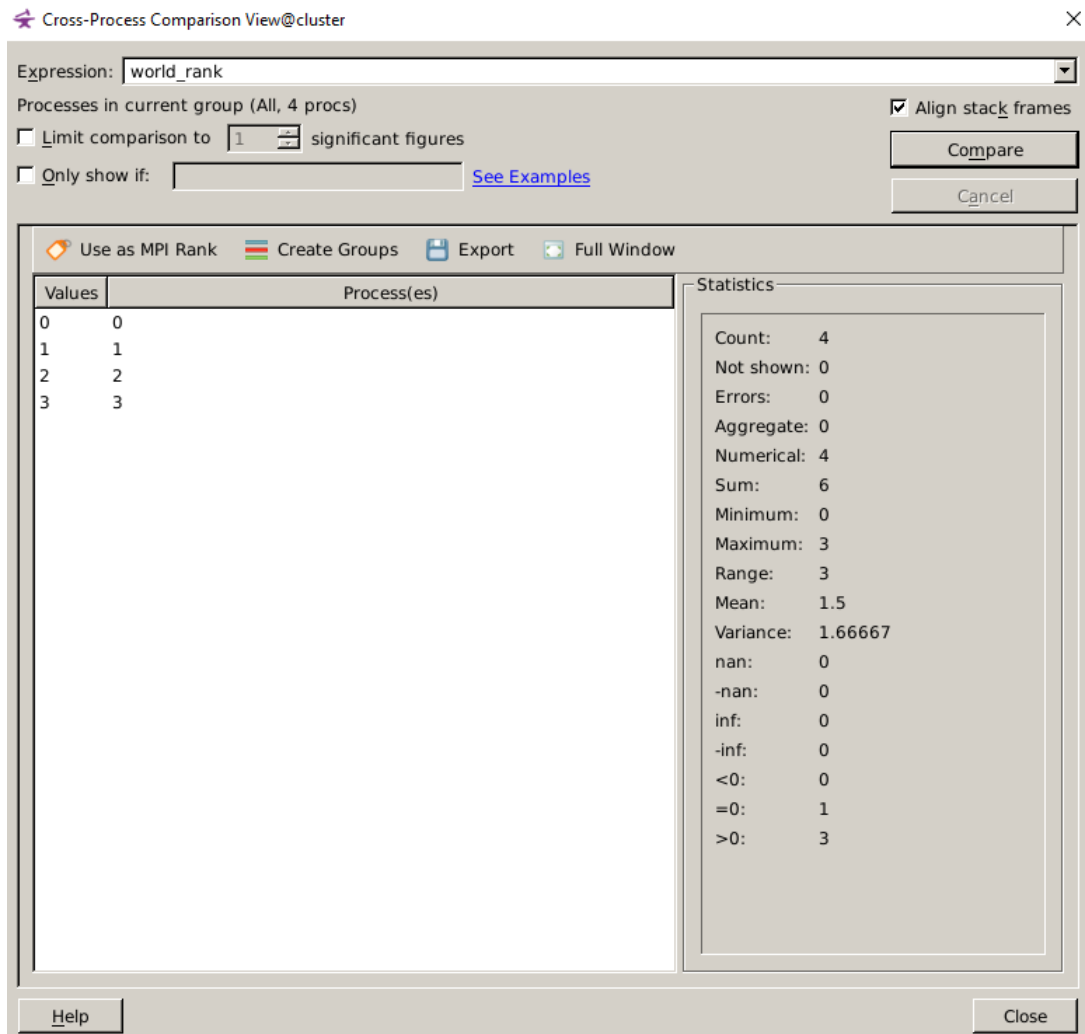


2. Setzen Sie in einer Zeile einen Breakpoint. Welche Step-Möglichkeiten gibt es und wie unterscheiden sich diese? Es gibt die drei Step-Möglichkeiten 'SStep Into', 'SStep Out' und 'SStep Over'. 'SStep Into' setzt die Ausführung fort, stoppt aber vor der Ausführung der ersten Zeile in der Funktion, falls eben jene Funktion davor aufgerufen wurde. 'SStep Out' setzt die Ausführung fort bis die momentane Funktion zurückkehrt und stoppt in der Zeile wo die Funktion aufgerufen wurde.

SStep Overführt die momentane Zeile aus und stoppt an der nächsten ausführbaren Zeile.

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. Einige Lokale Variablen haben eine Grafik vor dem eigentlichen Wert, welches anzeigt, welche Prozesse/Threads ungefähr dieselben oder verschiedene Werte haben. Dabei werden diese Unterschiede durch einen Strich pro Prozess dargestellt, die auf eine relative Höhe zueinanderstehen, je nachdem ob der Wert größer oder kleiner ist. Im Kontextmenu kann ausgewählt werden ob die Werte über Prozesse oder Threads verglichen werden sollen. In dem Fenster wo die Werte für die Variable pro Prozess angezeigt werden, kann man sehen das die Prozesse mit einer Nummer identifiziert werden. Da diese Nummern mit den jeweiligen Werten von world_rank übereinstimmen, scheinen diese Nummer auch der Rank eines Prozess darzustellen.





4. Machen Sie sich mit der Funktion des Evaluate-Fensters in der rechten unteren Ecke vertraut. In dem Evaluate-Fenster können wie erwartet nur Ausdrücke und keine Statements verwendet werden (also nichts was mit ";" endet). Es kann dadurch auch nichts deklariert oder Makros verwendet werden. Zur Verfügung stehen nur Konstrukte und Variablen, die im momentanen Kontext (der jetzigen Position im Stack) auch verfügbar sind. Es werden die Ausdrücke für alle Prozesse berechnet und auch wie im Fenster Locals ein Vergleich über die Prozesse mit Strichen sichtbar gemacht. Eingaben die nicht validen Ausdrücken in dem Kontext entsprechen, bringen das Programm wie erwartet nicht zum Absturz, die Fehlermeldungen werden dafür in dem Feld ausgegeben, wo sonst der Wert stehen würde.

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. Beim Wechseln

der Prozesse werden wie erwartet die Werte neu berechnet.

Evaluate		Evaluate	
Expression	Value	Expression	Value
-1+1	2	-1+1	2
-1+1;	<Invalid character	-1+1;	<Invalid chara
"asd"	"asd"	"asd"	"asd"
char *stu[100]	<A syntax error in i	char *stu[100]	<A syntax errc
char stuff_with_colon[100];	<A syntax error in i	char stuff_with_colon[100];	<A syntax errc
gettimeofday(&tv, 0)	0	gettimeofday(&tv, 0)	0
gettimeofday(&tv, NULL)	<No symbol "NULL	gettimeofday(&tv, NULL)	<No symbol "t
int i = 0;	<A syntax error in i	int i = 0;	<A syntax errc
world_rank	2	world_rank	3
world_rank + 100	102	world_rank + 100	103

6. Erweitern Sie Ihr Programm um ein Array und initialisieren Sie es mit beliebigen Zahlenwerten. Lassen Sie sich die Werte anzeigen. Welche sonstigen Visualisierungsmöglichkeiten bietet DDT? (Hinweis: Das Array wird nur für diese Aufgabe benötigt und soll in den abgegebenen Programmen timempi bzw. timempi2 nicht enthalten sein.) Bei globalen Variablen kann nur über das evaluate-Fenster als Ausdruck, bei lokalen Variablen aber dazu auch noch über das LocalsFenster darauf zugegriffen werden. Die Inhalt des Arrays kann durch Klicken auf das Kreuz-Symbol neben den Namen der Variabel angezeigt werden oder über das Kontextmenü-„View Arraymit dem Multi-Dimensional Array Viewer“.

```
#define _DEFAULT_SOURCE
#include <stdio.h>
#include <string.h>
#include <mpi.h>
#include <unistd.h>
#include <sys/time.h>
#include <time.h>

int main(int argc, char** argv) {
    int array[20] = {1,2,3,4,5,6,7,8,9,11,12};
    // Initialize the MPI environment
    MPI_Init(&argc, &argv);

    // Get the number of processes
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    // Get the rank of the process
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    char buffer[30];
    struct timeval tv;

    time_t curtime;

    gettimeofday(&tv, NULL);
    curtime=tv.tv_sec;

    // format time_val struct into a c string and put it into buffer
}
```

Variable Name	Value
argc	1
argv	0x7ffe08e9cc18
array	{[0] = 1, [1] = 2, [2] = 3, [3] = 4, [4] = 5, [5] = 6, [6] = 7, [7] = 8, [8] = 9, [9] = 11, [10] = 12, [11] = 0, [12] = 0, [13] = 0, [14] = 0, [15] = 0, [16] = 0, [17] = 0, [18] = 0, [19] = 0}
buffer	"003"
curtime	7
name	"p"
result	7

Expression	Value
-1+1	2
-1+1;	<Invalid character ':' in expression.>
"asd"	
char *stu[100]	<A syntax error in expression, near 'stu[100]':>
char stuff_with_colon[100];	<A syntax error in expression, near 'stuff_with_colon[100]':>
gettimeofday(&tv, 0)	0
gettimeofday(&tv, NULL)	<No symbol "NULL" in current context.>
int i = 0;	<A syntax error in expression, near 'int i = 0;':>
world_rank	0
world_rank + 100	100

Multi-Dimensional Array Viewer@cluster

Array Expression:

Distributed Array Dimensions:
[How do I view distributed arrays?](#)

☐ Staggered Array
 [What does this do?](#)

Range of \$i

From:

To:

Display:

☒ Align Stack Frames
☐ Auto-update

☐ Only show if:
[See Examples](#)

Data Table

0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	11
10	12
11	0
12	0