

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

DDT ermöglicht die direkte Eingabe von Programmparametern in einer Zeile, oder das Verwenden einer Input-File für kompliziertere Programmaufrufe:

Run (queue submission mode) <@cluster>

Application: /home/hansen1/HLR/06-MPI/timempi 4 Details

Application: /home/hansen1/HLR/06-MPI/timempi

Arguments: 4

☐ stdin file:

Working Directory:

☒ **MPI:** 4 processes, 2 nodes, MVAPICH 2 Details

Number of processes: 4 Number of Nodes: 2 Calculate

☐ Processes per Node: 2

Implementation: MVAPICH 2 Change...

mpirun arguments

☐ **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 Submit Cancel

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

DDT bietet die von anderen Debuggern bekannten Single-Step, Step-Over, Step-Out und Step-To Funktionalitäten, es gilt zu beachten, dass sowohl einzelne, als auch mehrere Prozesse zur gleichen Zeit gestept werden können, wenn diese die gleichen Codesegmente bearbeiten.

The screenshot shows the Allinea DDT 4.2.1-36484 <@cluster> IDE. The main window displays the source code of 'timempi.c'. A breakpoint is set at line 32, which is highlighted with a green arrow. The code is as follows:

```

13
14 int root_process; /* id of root process */
15 int my_id; /* id of the current process */
16 int i; /* loop iterator */
17 int amount_procs; /* total amount of processes */
18
19 const int timestamplength = 64;
20 char timestamp[64]; /* process individual timestamp */
21
22 /* start processes*/
23 MPI_Init(&argc, &argv);
24
25 root_process = 0;
26
27 /* get process id and amount of total processes */
28 MPI_Comm_rank(MPI_COMM_WORLD, &my_id);
29 MPI_Comm_size(MPI_COMM_WORLD, &amount_procs);
30
31 /* root process: monitoring and printing */
32 if (my_id == root_process) {
33     for(i = 1; i < amount_procs; i++)
34     {
35         MPI_Recv(timestamp, timestamplength, MPI_CHAR, i, return_data_tag, MPI_COMM_WORLD, &status);
36         printf("%s\n", timestamp);
37     }
38 }
39 else
40 {
41     char hostname[10]; /* hostname of current process */
42
43     struct timeval tv; /* time struct */
44     time_t nowtime;
45     struct tm *nowtm;
46     char tmbuf[64];
47
48     /* gets hostname and time */
49     gethostname(hostname, 40);
50     gettimeofday(&tv, NULL);
51     nowtime = tv.tv_sec;
52     nowtm = localtime(&nowtime);

```

The bottom pane shows the 'Breakpoints' tab with the following table:

Processes	Threads	File	Line	Function	Condition	Start After	Trigger Every	Stop After	Full path
<input checked="" type="checkbox"/> All	all	timempi.c	56	main		0	1	Forever	/home/hansen1/HLR/06-MPI/timempi.c
<input checked="" type="checkbox"/> All	all	timempi.c	32	main		0	1	Forever	/home/hansen1/HLR/06-MPI/timempi.c

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.

DDT weist durch ein Icon (hier durch Cursor verdeckt) darauf hin, dass die Variable über mehrere Prozesse verteilt unterschiedliche Werte annimmt. Über das Rechtsklickmenü wird eine Vergleichsübersicht geöffnet, welche den Wert der Variable in allen Prozessen anzeigt, sowie direkt weitere Statistiken berechnet.

The screenshot displays the 'Cross-Process Comparison View' in a development environment. The 'Expression' field is set to 'my_id'. The 'Processes in current group' are 'All, 4 procs'. The 'Values' table shows the following data:

Values	Process(es)
0	0
1	1
2	2
3	3

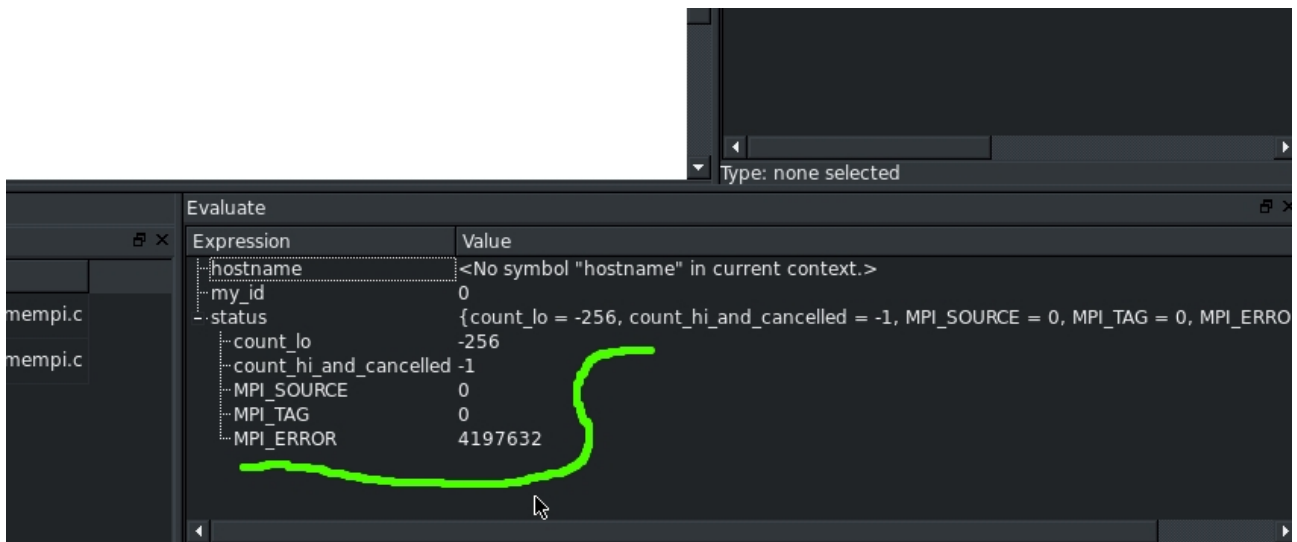
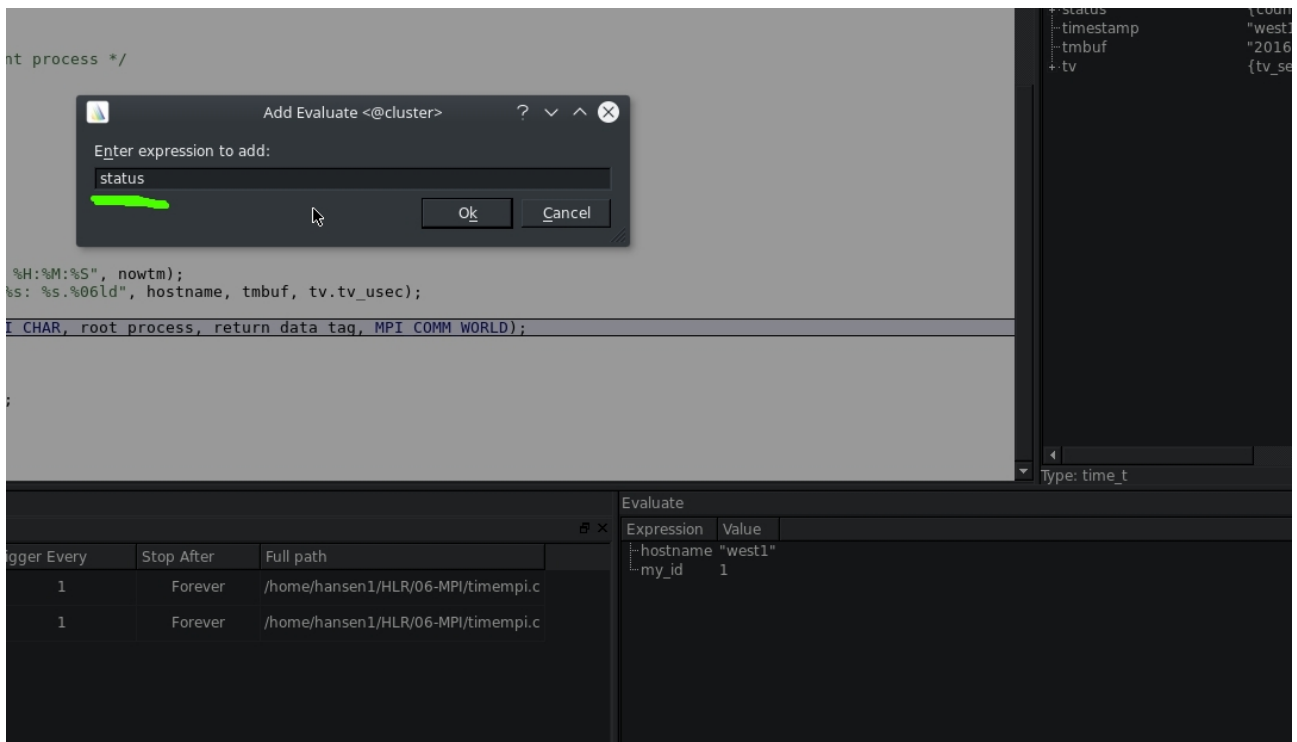
The 'Statistics' section provides the following summary:

- Count: 4
- Not shown: 0
- Errors: 0
- Aggregate: 0
- Numerical: 4
- Sum: 6
- Minimum: 0
- Maximum: 3
- Range: 3
- Mean: 1.5
- Variance: 1.66667
- nan: 0
- nan: 0
- inf: 0
- inf: 0
- <0: 0
- =0: 1
- >0: 3

The 'Locals' panel on the right shows the variable 'my_id' with a value of 0. A red arrow points to the 'my_id' variable in the 'Locals' panel, and a red circle highlights the 'Values' table in the 'Cross-Process Comparison View'.

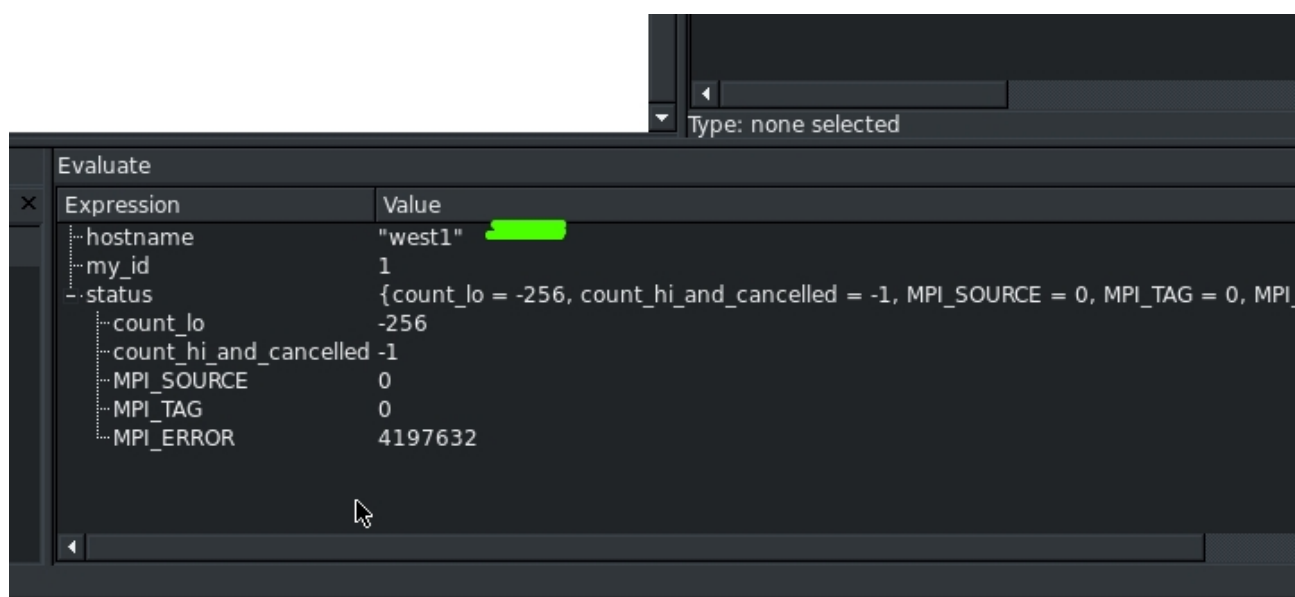
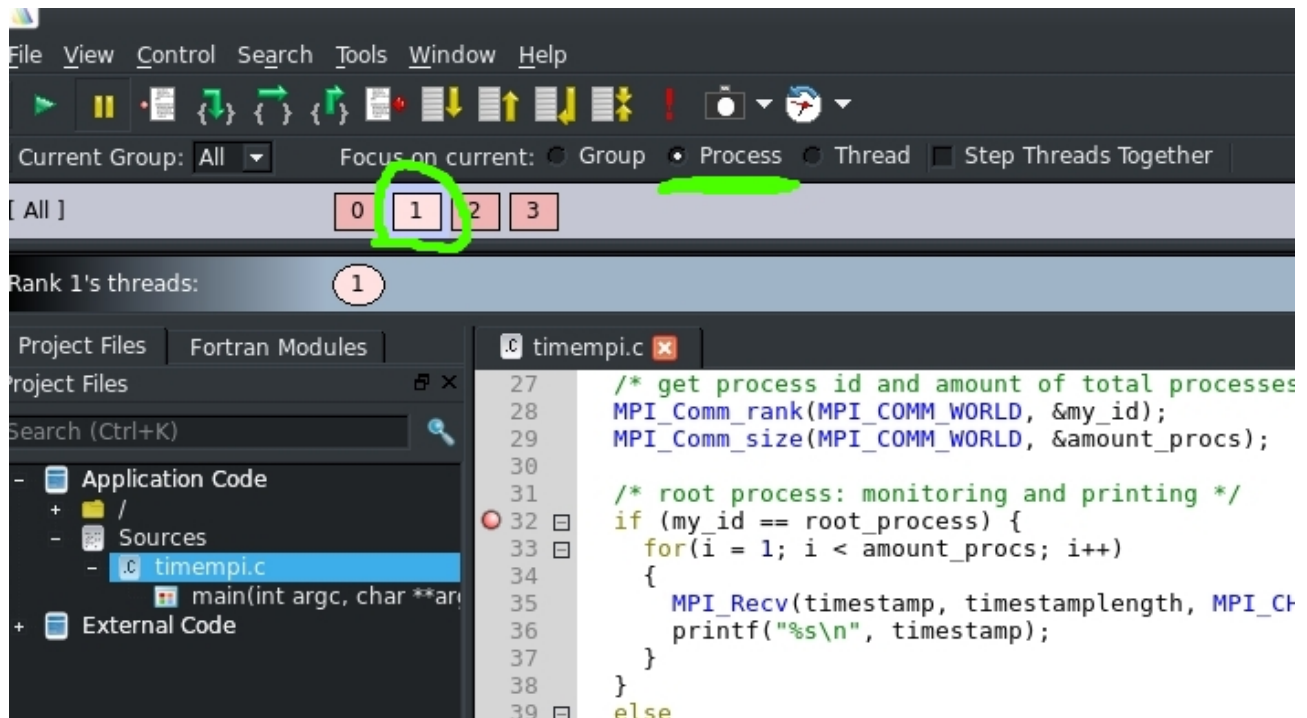
Machen Sie sich mit der Funktion des Evaluate-Fensters in der rechten unteren Ecke vertraut

Das Evaluate Fenster zeigt ausgewählte Variablen des aktuellen Prozesses an

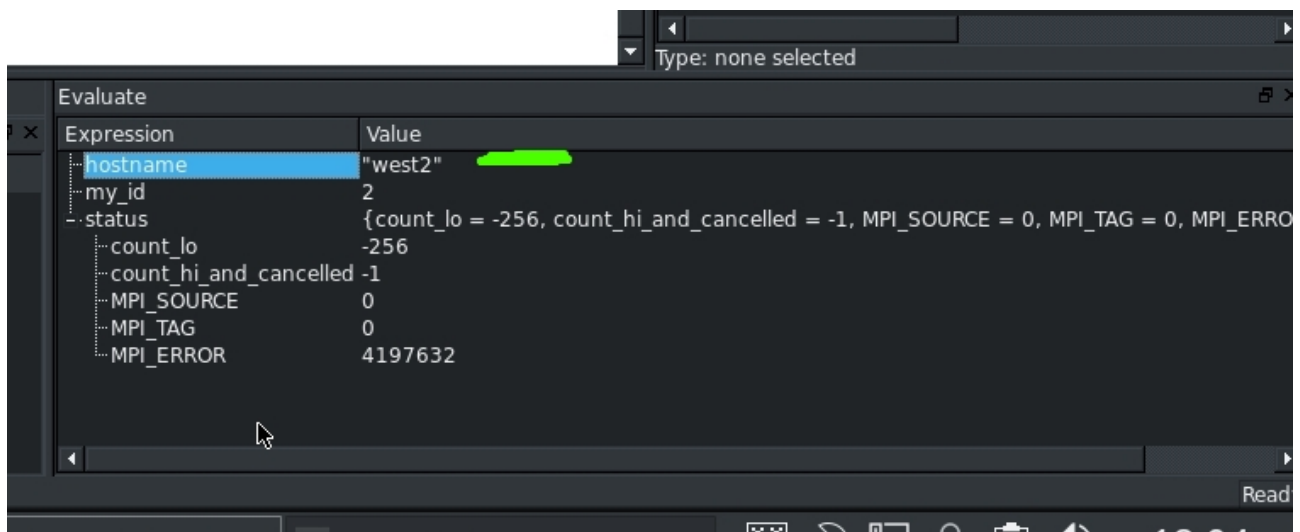
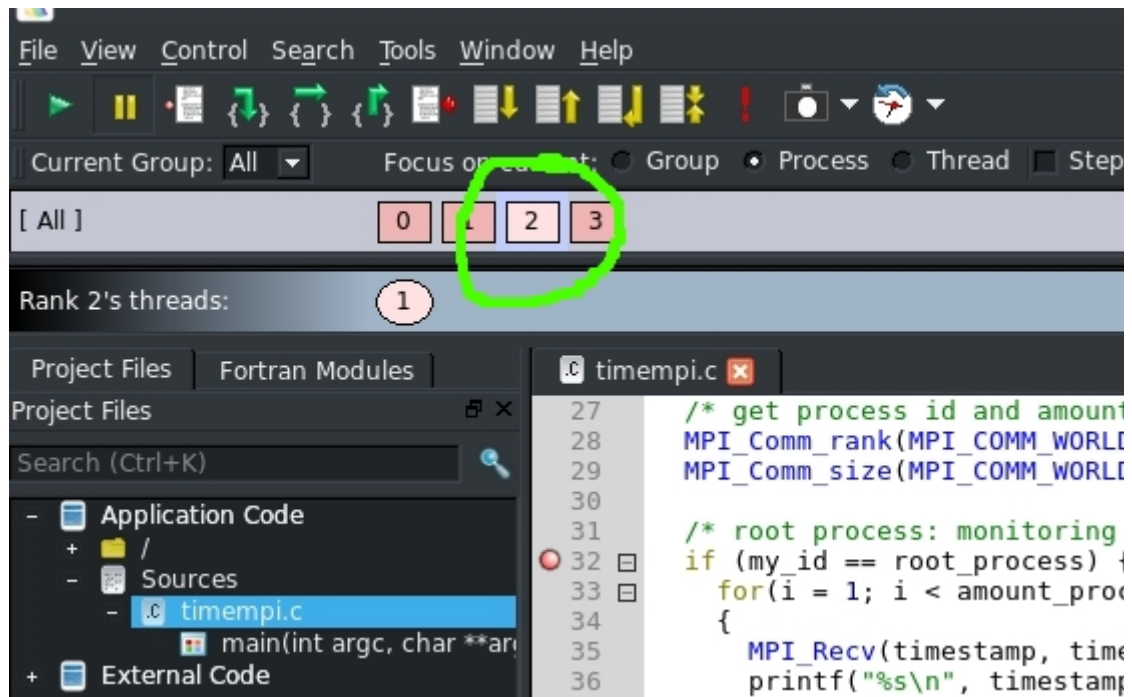


In der oberen Leiste nden Sie eine Übersicht aller Prozesse und Threads Ihres Programmes. Wechseln Sie zwischen den einzelnen Prozessen und beobachten Sie das Evaluate-Fenster.

Mit dem Wechsel zwischen Prozessen verändert sich auch der Scope der Variablen, dies wird im Evaluate-Fenster reflektiert.

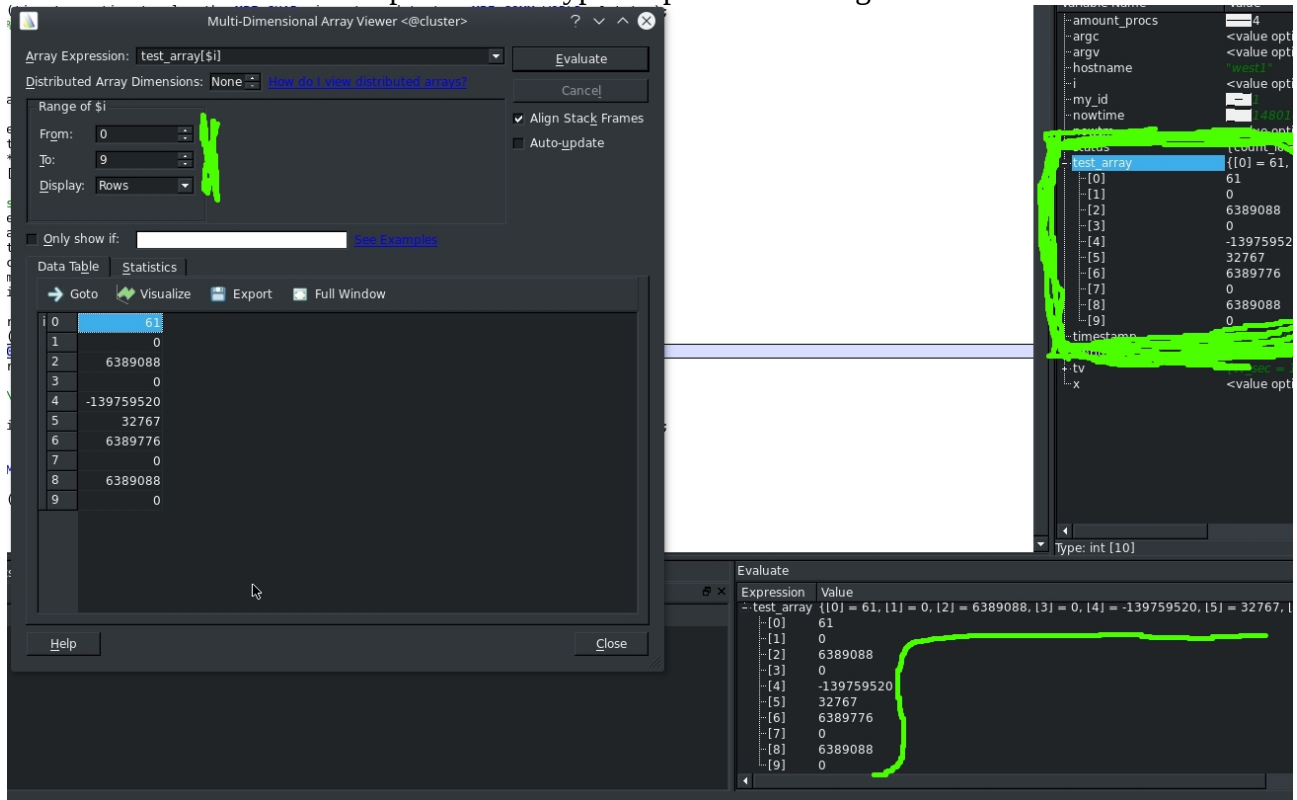


Nach dem Prozesswechsel:



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?

Das Array ist sowohl in der Übersicht der lokalen Variablen, als auch im Evaluate Fenster ähnlich einer Struct oder anderer komplexer Datentypen expandierbar dargestellt.



Die Detailansicht ermöglicht das anzeigen der Arraywerte über einen beliebigen Bereich, der Vergleich über mehrere Prozesse oder eine Visualisierung als 3D-Plot:

