The IBM High Performance Computing Toolkit

PeekPerf Documentation

Please send corrections to: David Klepacki
(klepacki@us.ibm.com)

PeekPerf has been developed as a viewer for data generated by:

- SiGMA
- HPM
- MPI/TurboSHMEM Trace Library
- MPI/TurboSHMEM Profiling Library

PeekPerf is a tool that will allow you to map your collected performance data back to the source code. So you will easier find bottlenecks and points for optimizations. PeekPerf is available for several UNIX derivations (AIX, Linux) and Microsoft® Windows®.

Content

The IBM High Performance Computing Toolkit	1
How to start PeekPerf	3
PeekPerf Main Window	4
PeekPerf Metrics View	6
PeekPerf Tracer	8
PeekPerf Setup	11
PeekPerf Command Line Options	12
PeekView Command Line Options	12
PeekPerf Configuration Files	13

How to start PeekPerf

PeekPerf is started from the command line with (where `tc>` is my prompt)

```
tc> peekperf
```
or
```
tc> peekperf <vizfiles>
```

You can specify more than one .viz file. PeekPerf will open all this files and combine the data from all the files. In this document we will refer to a MPI profile taken on a POWER4 running the ‘sweep3d’ benchmark with 16 tasks. PeekPerf has been called with all sixteen .viz files:

```
tc> peekperf sweep3d/mpi_profile*
```

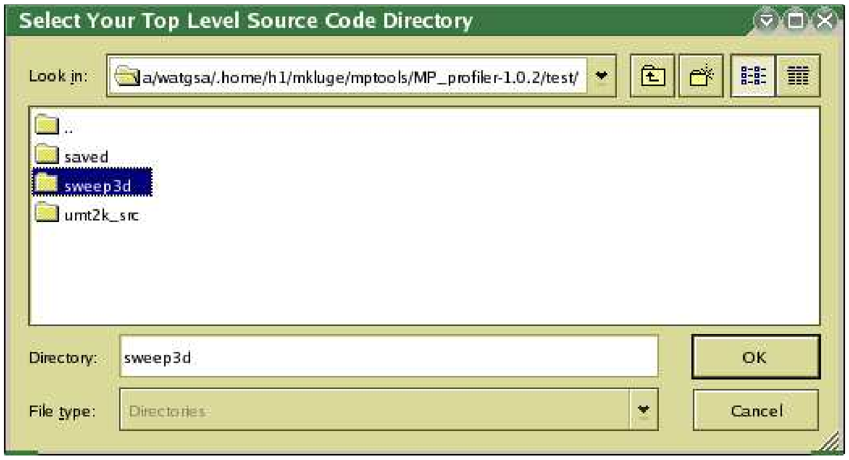
It is possible to start only the trace viewer by calling

```
tc> peekview sweep3d/single_trace
```

For a list of available options see the end of this document.

PeekPerf Main Window

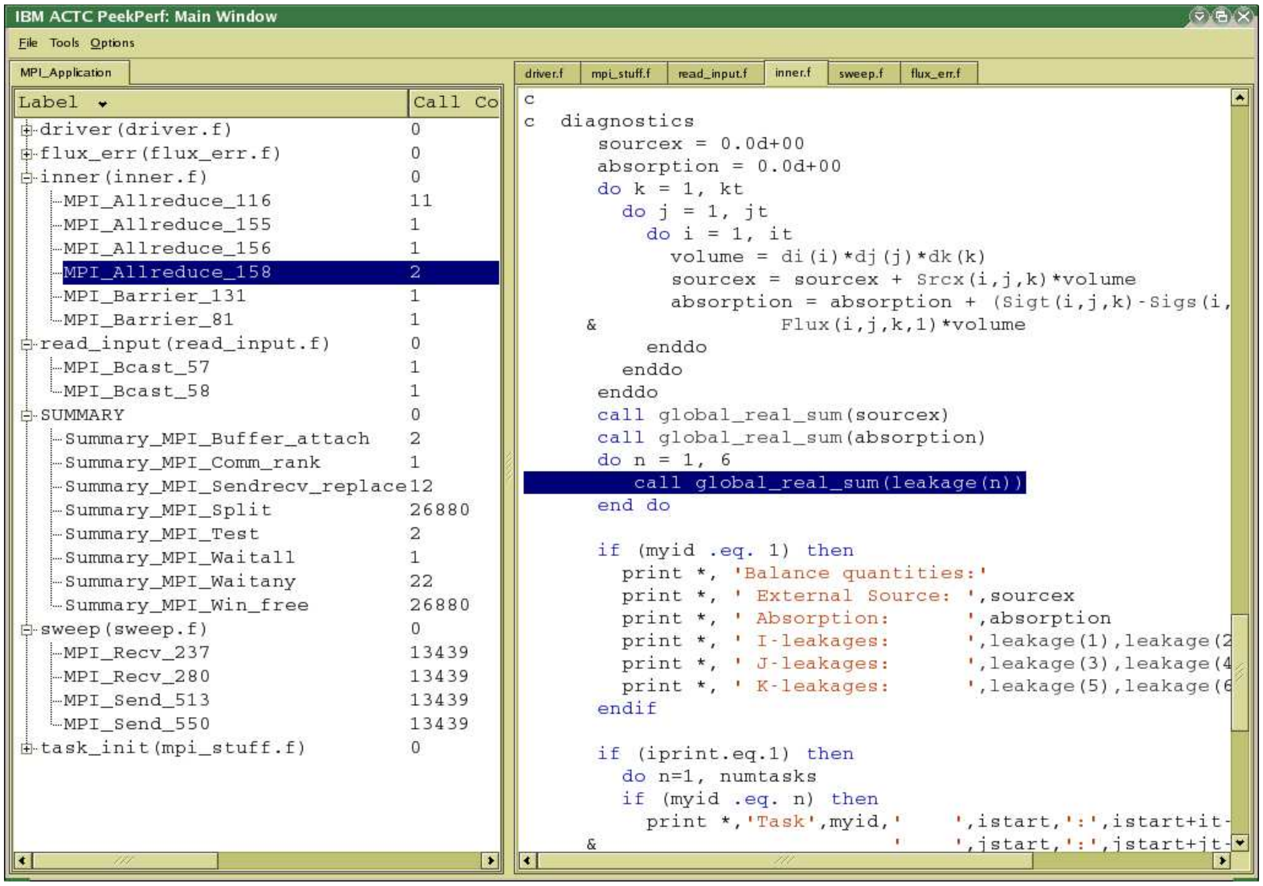
PeekPerf will try to find your source code first. Before the main window is opened, you have to select the top level directory for your source files (Note: In Microsoft® Windows® the Main Window is shown first and then you are asked for the source).


Select Your Top Level Source Code Directory
Look in:
a/watgsa/.home/h1/mkluge/mptools/MP_profiler-1.0.2/test/
..
saved
sweep3d
umt2k_src
Directory:
sweep3d
File type:
Directories
OK
Cancel

PeekPerf tries to find all the source files within this directory. If it finds more than one file with the same name, you will be asked to select the right one. If PeekPerf is unable to find a single file, it will ask for more source directories. You are able to cancel this search. But then the mapping from the data shown to the source code may be impossible. PeekPerf will then open all the source files. While loading the files, the Main Window is shown. If you it has to load at lot of source (lets say 20 MB or more) it will a few several minutes to parse all the files to make the source code highlighting. We have seen for the source for the GAMESS program, where about 32 MB of code has to be loaded, that it took about 2.5 minutes with an 1.8 Ghz desktop computer to load and to parse this code. However, you can use the program while it is still loading the sources. Later you will se how to disable the parsing of the source code to speed up the loading process with large and/or many source files.

The left side of the main window shows the collected performance data. If you click with the left mouse button on an item on the right side the corresponding line of source will be highlighted immideateely. In this example you see that the MPI_Bcast function at line 219 of the file mpi_stuff.f has been called only once.

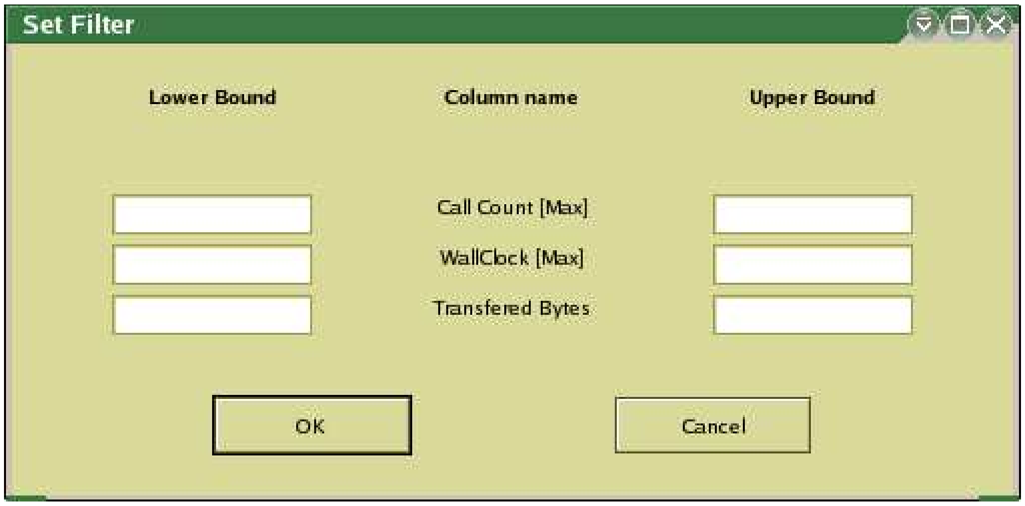
You can open additional data using the menu point File->Open. So it is possible to view the HPM data (recorded, because you had linked your program with the HPM library) and the MPI or SHMEM data (because you had linked with the trace library too) simulaneously.



```
Label                         Call Co
driver(driver.f)              0
flux_err(flux_err.f)          0
inner(inner.f)                0
  MPI_Allreduce_116           11
  MPI_Allreduce_155           1
  MPI_Allreduce_156           1
  MPI_Allreduce_158           2
  MPI_Barrier_131             1
  MPI_Barrier_81              1
read_input(read_input.f)      0
  MPI_Bcast_57                1
  MPI_Bcast_58                1
SUMMARY                       0
  Summary_MPI_Buffer_attach   2
  Summary_MPI_Comm_rank       1
  Summary_MPI_Sendrecv_replace12
  Summary_MPI_Split           26880
  Summary_MPI_Test            2
  Summary_MPI_Waitall         1
  Summary_MPI_Waitany         22
  Summary_MPI_Win_free        26880
sweep(sweep.f)                0
  MPI_Recv_237                13439
  MPI_Recv_280                13439
  MPI_Send_513                13439
  MPI_Send_550                13439
task_init(mpi_stuff.f)        0
```

```
c
c   diagnostics
      sourcex = 0.0d+00
      absorption = 0.0d+00
      do k = 1, kt
        do j = 1, jt
          do i = 1, it
            volume = di(i)*dj(j)*dk(k)
            sourcex = sourcex + Srcx(i,j,k)*volume
            absorption = absorption + (Sigt(i,j,k)-Sigs(i,
     &               Flux(i,j,k,1)*volume
          enddo
        enddo
      enddo
      call global_real_sum(sourcex)
      call global_real_sum(absorption)
      do n = 1, 6
         call global_real_sum(leakage(n))
      end do

      if (myid .eq. 1) then
        print *, 'Balance quantities:'
        print *, ' External Source: ',sourcex
        print *, ' Absorption:      ',absorption
        print *, ' I-leakages:      ',leakage(1),leakage(2
        print *, ' J-leakages:      ',leakage(3),leakage(4
        print *, ' K-leakages:      ',leakage(5),leakage(6
      endif

      if (iprint.eq.1) then
        do n=1, numtasks
        if (myid .eq. n) then
          print *,'Task',myid,'     ',istart,':',istart+it-
     &                          '     ',jstart,':',jstart+jt-
```

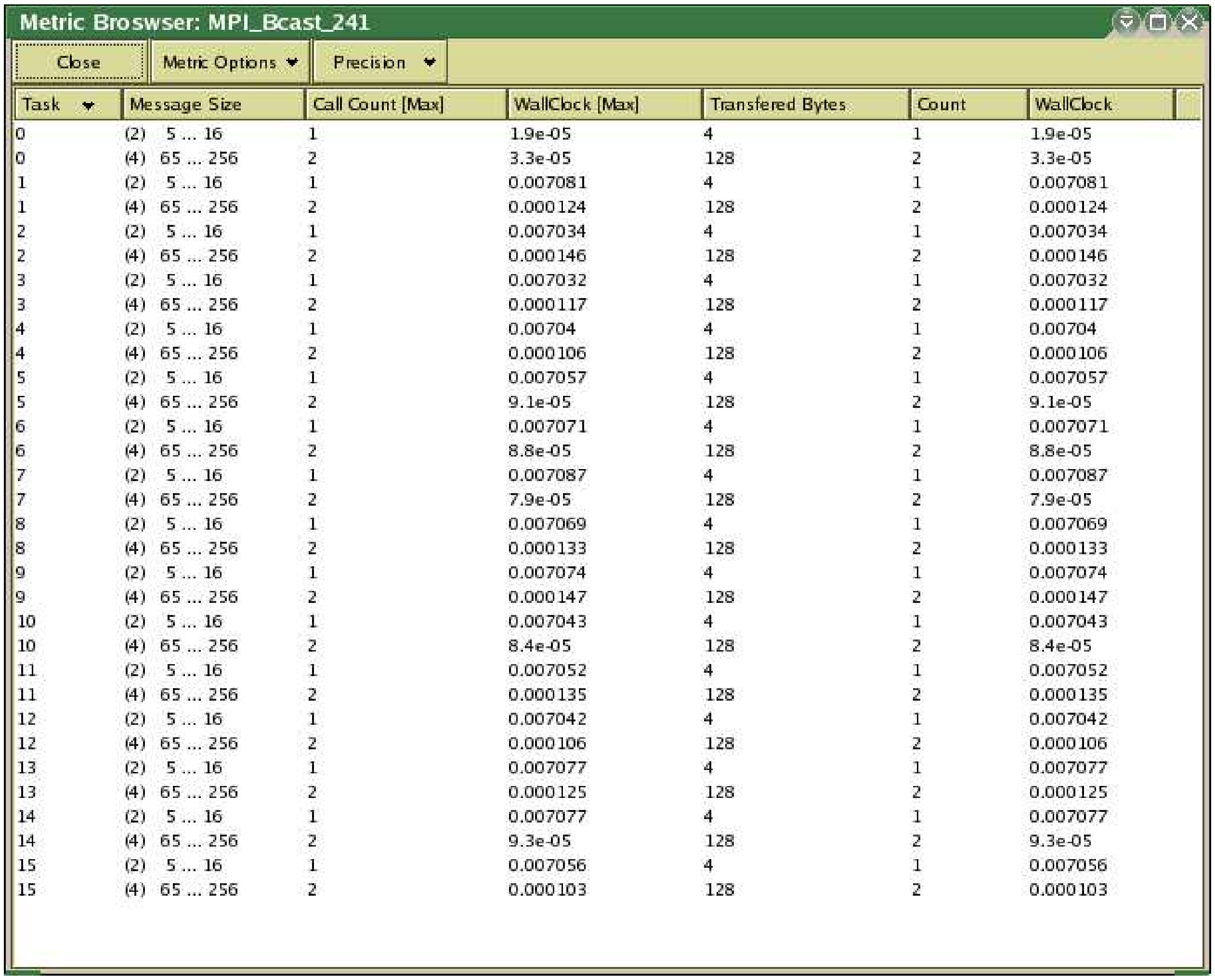
You can set filters to exclude some performance data from beeing shown by setting a filter. The Filter window can be opened via Tools->Filter.



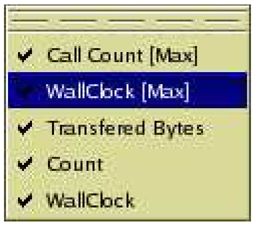
Lower Bound	Column name	Upper Bound
	Call Count [Max]	
	WallClock [Max]	
	Transfered Bytes	

PeekPerf Metrics View

If you click, in our example on the next line in the performance data view, you will see all metrics collected for this line of source. In this example we see the performance data for all the 16 tasks of this example for this single MPI call. We see how often MPI_Bcast from this line of source has been called and with which message sizes. We also see the aggregated number of transferred bytes and the aggregated time.



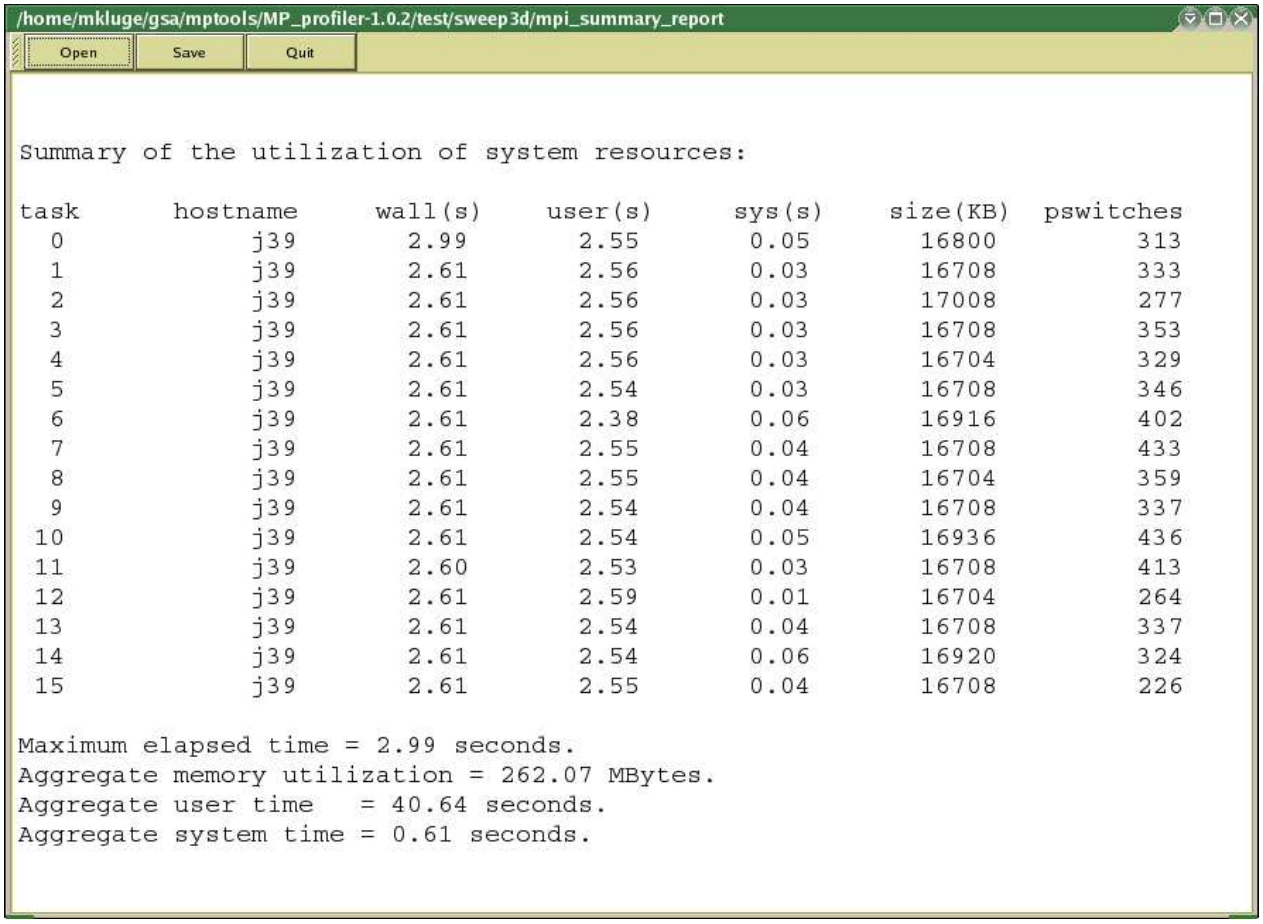
Task	Message Size	Call Count [Max]	WallClock [Max]	Transfered Bytes	Count	WallClock
0	(2) 5 ... 16	1	1.9e-05	4	1	1.9e-05
0	(4) 65 ... 256	2	3.3e-05	128	2	3.3e-05
1	(2) 5 ... 16	1	0.007081	4	1	0.007081
1	(4) 65 ... 256	2	0.000124	128	2	0.000124
2	(2) 5 ... 16	1	0.007034	4	1	0.007034
2	(4) 65 ... 256	2	0.000146	128	2	0.000146
3	(2) 5 ... 16	1	0.007032	4	1	0.007032
3	(4) 65 ... 256	2	0.000117	128	2	0.000117
4	(2) 5 ... 16	1	0.00704	4	1	0.00704
4	(4) 65 ... 256	2	0.000106	128	2	0.000106
5	(2) 5 ... 16	1	0.007057	4	1	0.007057
5	(4) 65 ... 256	2	9.1e-05	128	2	9.1e-05
6	(2) 5 ... 16	1	0.007071	4	1	0.007071
6	(4) 65 ... 256	2	8.8e-05	128	2	8.8e-05
7	(2) 5 ... 16	1	0.007087	4	1	0.007087
7	(4) 65 ... 256	2	7.9e-05	128	2	7.9e-05
8	(2) 5 ... 16	1	0.007069	4	1	0.007069
8	(4) 65 ... 256	2	0.000133	128	2	0.000133
9	(2) 5 ... 16	1	0.007074	4	1	0.007074
9	(4) 65 ... 256	2	0.000147	128	2	0.000147
10	(2) 5 ... 16	1	0.007043	4	1	0.007043
10	(4) 65 ... 256	2	8.4e-05	128	2	8.4e-05
11	(2) 5 ... 16	1	0.007052	4	1	0.007052
11	(4) 65 ... 256	2	0.000135	128	2	0.000135
12	(2) 5 ... 16	1	0.007042	4	1	0.007042
12	(4) 65 ... 256	2	0.000106	128	2	0.000106
13	(2) 5 ... 16	1	0.007077	4	1	0.007077
13	(4) 65 ... 256	2	0.000125	128	2	0.000125
14	(2) 5 ... 16	1	0.007077	4	1	0.007077
14	(4) 65 ... 256	2	9.3e-05	128	2	9.3e-05
15	(2) 5 ... 16	1	0.007056	4	1	0.007056
15	(4) 65 ... 256	2	0.000103	128	2	0.000103



When clicking on “Metric Options“ you can enable and disable metric options of your choice.

PeekPerf Summary View

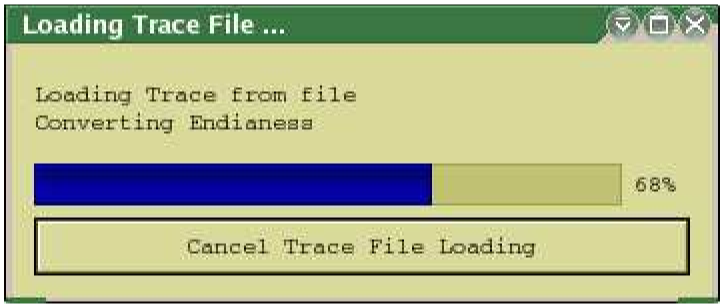
When you select Tools->Summary you will get a view of text file produced by the trace library describing a very brief view of the execution. So you are able to recognize the performance data easier and to see the execution from the OS view. The data shown are collected using the getrusage() system call.


/home/mkluge/gsa/mptools/MP_profiler-1.0.2/test/sweep3d/mpi_summary_report
Open
Save
Quit

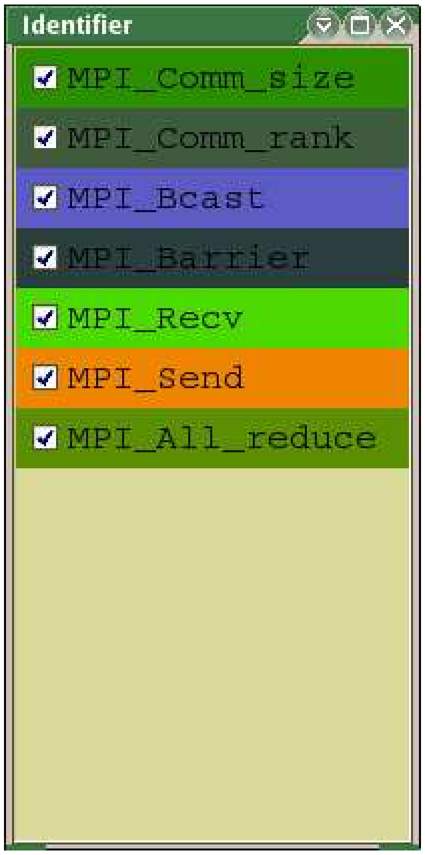
```
Summary of the utilization of system resources:

task      hostname     wall(s)     user(s)      sys(s)     size(KB)  pswitches
  0            j39       2.99        2.55        0.05       16800          313
  1            j39       2.61        2.56        0.03       16708          333
  2            j39       2.61        2.56        0.03       17008          277
  3            j39       2.61        2.56        0.03       16708          353
  4            j39       2.61        2.56        0.03       16704          329
  5            j39       2.61        2.54        0.03       16708          346
  6            j39       2.61        2.38        0.06       16916          402
  7            j39       2.61        2.55        0.04       16708          433
  8            j39       2.61        2.55        0.04       16704          359
  9            j39       2.61        2.54        0.04       16708          337
 10            j39       2.61        2.54        0.05       16936          436
 11            j39       2.60        2.53        0.03       16708          413
 12            j39       2.61        2.59        0.01       16704          264
 13            j39       2.61        2.54        0.04       16708          337
 14            j39       2.61        2.54        0.06       16920          324
 15            j39       2.61        2.55        0.04       16708          226

Maximum elapsed time = 2.99 seconds.
Aggregate memory utilization = 262.07 MBytes.
Aggregate user time   = 40.64 seconds.
Aggregate system time = 0.61 seconds.
```

PeekPerf Tracer

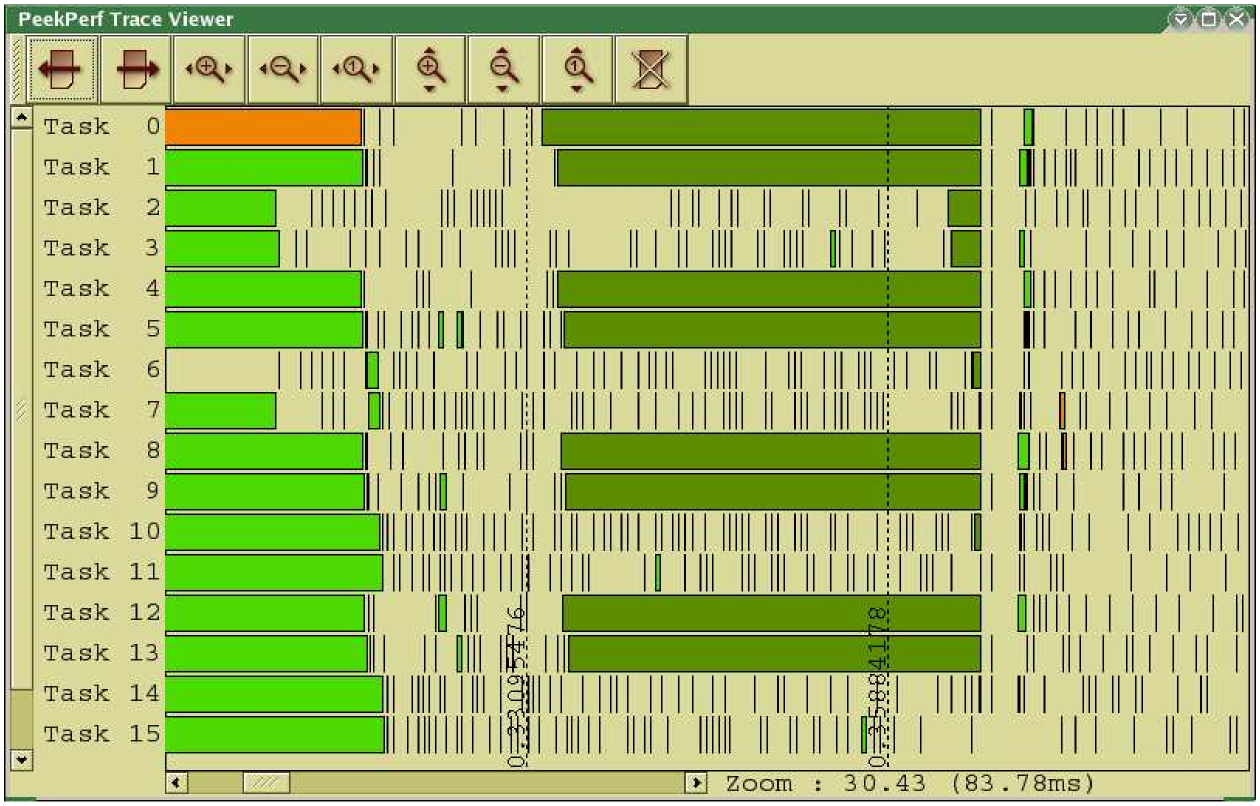
If you have used the trace libraries you will be able to use the trace viewer to see a detailed view of the execution of the program. When opening a trace file PeekPerf will automatically take care about the correct endianess and, if necessary, convert the trace file to the correct endianess. So there is no need to convert any of the files when moving them from the system where they have been generated (for example: POWER4 running AIX) to a system where they should be viewed (for example: Workstation with AMD processor running Linux or Microsoft® Windows®).

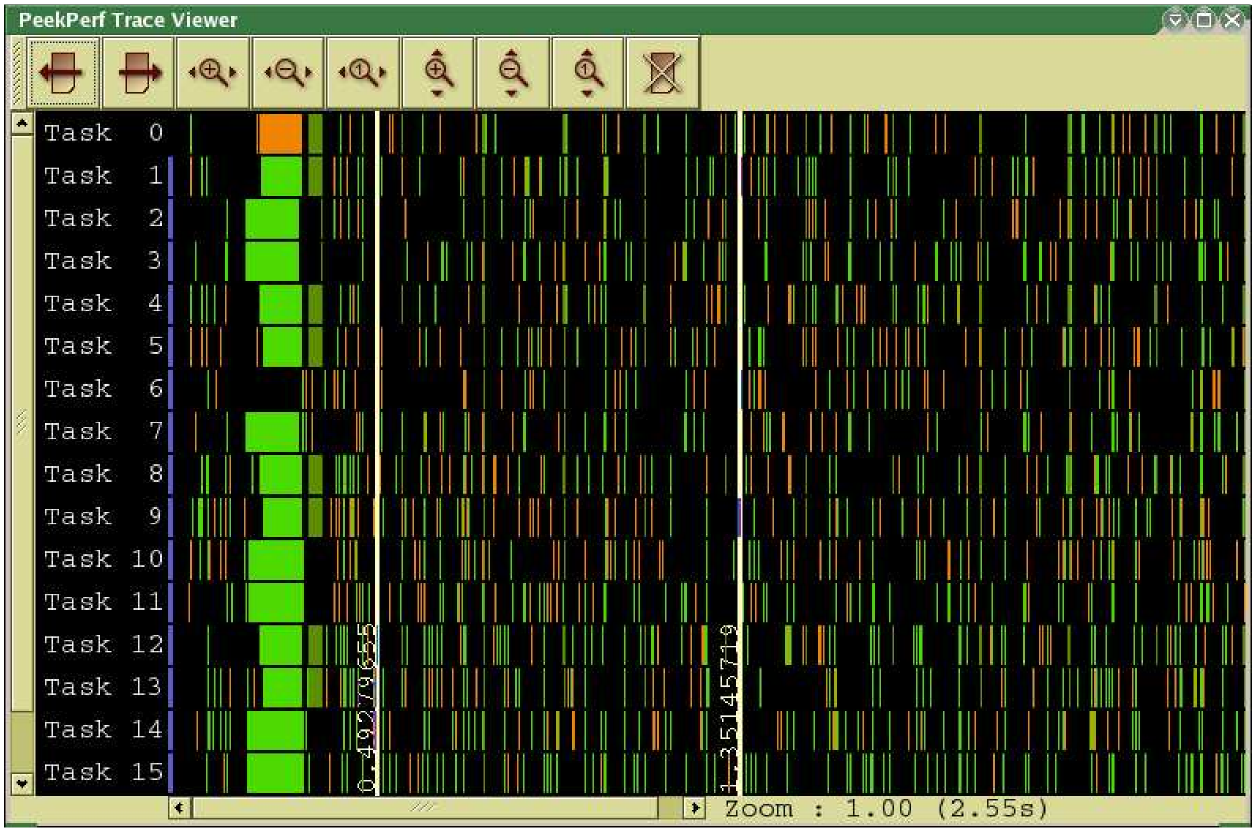


```
Loading Trace from file
Converting Endianess
68%
```



The trace viewer itself shows the tasks at the y axis and the time at the x axis. For every task you can see the timeline. Every MPI call is highlighted with an other color. The MPI traces can be viewed with a black or a bright background. When using the bright background every event is surrounded by a black rectangle which makes it easier to identify very short events. When viewing a lot of events all this black frames will create an very distorted view of the scene, so you are maybe happier with the black background. The picture at the left shows a screenshot of a window which is shown beside the trace viewer. So you are able to map easy from the timeline to a event within the timeline. You can supress some types of events from beeing shown simply be clicking on the event type in the 'Identifier' window.


PeekPerf Trace Viewer
Task 0
Task 1
Task 2
Task 3
Task 4
Task 5
Task 6
Task 7
Task 8
Task 9
Task 10
Task 11
Task 12
Task 13
Task 14
Task 15
0.3209554176
0.35884178
Zoom : 30.43 (83.78ms)


PeekPerf Trace Viewer
Task 0
Task 1
Task 2
Task 3
Task 4
Task 5
Task 6
Task 7
Task 8
Task 9
Task 10
Task 11
Task 12
Task 13
Task 14
Task 15
0.49279655
1.3514571
Zoom : 1.00 (2.55s)

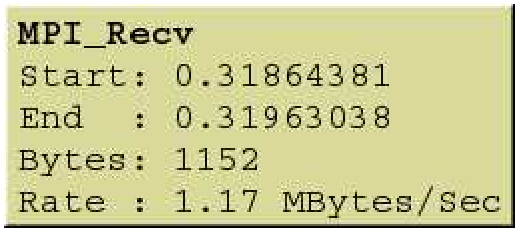
If you click on an event with in the tracer window you will see when clicking with the:

- Left Mouse Button

  That the correspondig line of source within the PeekPerf Main Window is highlighted. If you hold the left mouse button and move the mouse pointer to an other point you will see two lines. The first line is shown at the origin (where you pressed the mouse button). The second line is shown at the current position of the mouse. If you lift the mouse button now the timeline zooms automatically into this selected timeframe.

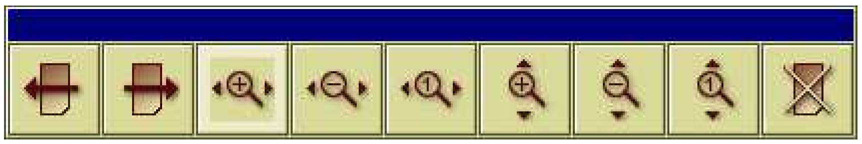
- Right Mouse Button

  You will see a summary of the collected data for this special event. Please note that when ever an event transmits some bytes we will try to calculate a transfer rate by simply dividing the number of transferred bytes by the elapsed time. In case of non-blocking calls (for example: MPI_Irecv()) these data will not show the associated physical transfer rate.



```
MPI_Recv
Start: 0.31864381
End  : 0.31963038
Bytes: 1152
Rate : 1.17 MBytes/Sec
```

You can move through the trace using the toolbar on top. This toolbar can be undocked from the window. You are able to zoom in and out vertically and horizontally.

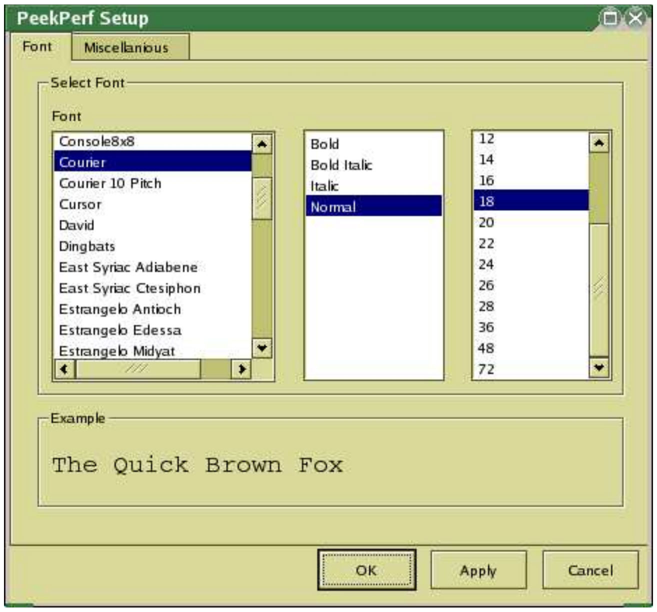


An other way to move around within the execution trace are the following keys:

Key	*Action*
←	move traces to the left
→	move trace to the right
↑	scroll up through tasks
↓	scroll down through tasks
Page Up	scroll trace faster to the left
Page Down	scroll trace faster to the right
'z' or 'y'	zoom time in
'x'	zoom time out
'a'	zoom tasks in
's'	zoom tasks out

PeekPerf Setup

The setup window is available via Options->View.



Select Font

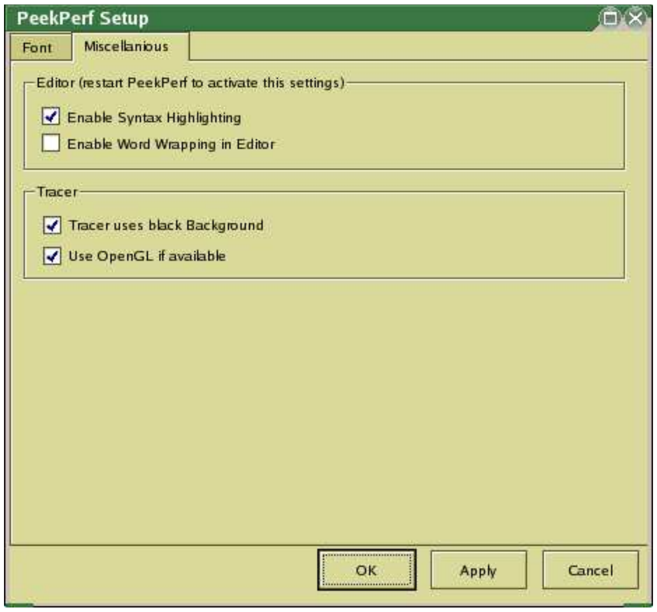
Font: Console8x8, [✓ Courier], Courier 10 Pitch, Cursor, David, Dingbats, East Syriac Adiabene, East Syriac Ctesiphon, Estrangelo Antioch, Estrangelo Edessa, Estrangelo Midyat

Bold, Bold Italic, Italic, [✓ Normal]

12, 14, 16, [✓ 18], 20, 22, 24, 26, 28, 36, 48, 72

Example

The Quick Brown Fox



Editor (restart PeekPerf to activate this settings)

- [✓] Enable Syntax Highlighting
- [ ] Enable Word Wrapping in Editor

Tracer

- [✓] Tracer uses black Background
- [✓] Use OpenGL if available

Here you can chose the font used by the whole application and some options for the source code window.

You are able to enable the black background in the tracer. One the most workstations today the OpenGL graphics interface should be available. Try to enable this options. Usually the trace viewer should be much faster when using bigger traces (more than half a million events; the number of trace events is shown on the command line after loading a trace file). If there is no OpenGL available the viewer will automatically fall back to the normal viewer. There can be one reason to disable the OpenGL support even if it is available. The default for setup for some linux distributions does not include installing an appropriate OpenGL driver according to your graphics card. Only a default software emulation is used. This software emulation seems to be a bit slower than using the normal viewer. So if you are confronted with this problem, check you X setup first.

PeekPerf Command Line Options

```
peekperf <vizfiles>
```

`<vizfiles>`	a list of .viz files to be analyzed

PeekView Command Line Options

```
peekview [Options] <tracefile(s)>
```

`<tracefile(s)>`	the tracefile(s) to be shown

PeekView (like PeekPerf) reads at its start a configuration file (see next section for details). To overide this defaults ther use the `[Options]` which include:

`-n or --noopengl`	do not try to use OpenGL
`-b or --bright`	force tracer to have a bright background
`-d or --dark`	force tracer with black background

PeekPerf Configuration Files

PeekPerf and PeekView use two configuration files located within the .peekperf (please note the dot, for Microsoft® Windows® without the dot) directory within your home directory (AIX,Linux: ~ or /home/<username>, Microsoft® Windows®: C:\Documents and Settings\<username>, but this differs for the various versions). The file 'config' contains some configuration data used by the setup window. When you start PeekPerf the first time, the configuration directory and these configuration files will be created for you.

The file 'mpi_ids' contains a mapping from an integer (first column of the file) to an color (rgb encoded, second through fourth column) and an string (fifth column). Please do not change the first and last column. This will mix up the the trace view. But change the colors as you want. Each of the columns two to four hold the brightness of the red, green and blue part of a color. The possible values are 0 to 255, where 0 means 'nothing from this part' and 255 means 'full intensity'. The triple (255,255,255) encodes white, the triple (0,0,0) black and the triple (100,0,0) a darker red.