



Profiling Tutorial #4



Batch Options and
Advanced Profiling Flags



Outline:

ProfVis batch reference:

- Runtime flags
- Batch Options
- Batch Analyses

ProfVis Batch Options



Runtime flags

Number of output files for each type of profiling database (standard, trace, comm):

Size of database before flushing (comm and trace):

Copied from amrex.
Not yet implemented:

Default values:

Found in Src/Base/AMReX_BLPProfiler.{H,cpp}

```
ParmParse pParse("blprofiler");  
pParse.query("prof_nfiles", nProfFiles);  
pParse.query("prof_csflushsize", csFlushSize);  
pParse.query("prof_traceflushsize", traceFlushSize);  
pParse.query("prof_flushinterval", flushInterval);  
pParse.query("prof_flushtimeinterval", flushTimeInterval);  
  
const int defaultFlushSize = 8192000;  
int BLPProfiler::csFlushSize = defaultFlushSize;  
int BLPProfiler::traceFlushSize = defaultFlushSize;  
int BLPProfiler::nProfFiles = 256;
```

amrvis -help

```
amrvis2dpp [<options>] [<filename(s)>]
  -help          print help and exit.
  ...
  [ standard amrvis help output]
  ...
----- amrparser functions
[-actpf f] output a plotfile for all call times for func f.
           f is a quoted string.
[-check]  data integrity check.
[-dispatch] use the dispatch interface.
```

Note: My link name is amrvis2dpp for amrvis2d, prof parser.

- List of batch options available to the profiling tools.
- This was the original toolset, but the GUI adds additional functionality to each option (filtering, visualization of data) so everything is being transitioned over.
- Currently, profiling options are independent from standard amrvis options and cannot be combined. (Expected to change in the future.)
- To combine ProfVis with Amrvis, some features may be incompatible or the number of flags may be limited.

Using profvis batch features

```
amrvis2dpp [<options>] [<filename(s)>]
```

e.g: analyzing a gridlog

In the directory with “gridlog”:

```
amrvis2dpp -gl bl_prof
```

Not, as you might expect:

```
amrvis2dpp -gl gridlog or amrvis2dpp -gl
```

- To distinguish the Profvis features from the AMRVis features, there is an initial check for whether or not it is pointed at a profiling database (bl_prof).
 - If so, it looks for profiling batch options.
 - If not, it uses standard amrvis options.
 - Currently mutually independent.
- This can lead to slightly awkward batch implementations, as the profiler must be pointed at a profiling database, even if one isn't needed.
- Can trick the profiler in those cases by using a folder starting with “bl_prof”:

e.g. `amrvis2dpp -gl bl_prof.empty`

GUI duplicates in batch

amrvis2dpp [<options>] [<filename(s)>]

----- amrproparser functions

[-rplt] make region plot file. ← ProfVis Main Window
[-sendspf] output a sends plotfile.
[-srlist] list sends and receives.
[-timelinepf] output a timeline plotfile.
[-ttrace] write text call trace. ← Complete Text Trace
[-ws] write summary. ← Function List Results
[-wts] write trace summary. ← Function List Results

- Limited parallel processing.
- Works only on the entire database. (No filtering).
- Immediately get desired plotfile.

Batch Options #1

amrvis2dpp [<options>] [<filename(s)>]

----- amrproparser functions

[-check] data integrity check.

[-dispatch] use the dispatch interface.

[-gpct] set percent threshold for xgraphs. range [0, 100]

[-msil n] sets maxSmallImageLength.

[-nocomb] do not combine adjacent call traces.

Check profiling database is consistent and correct.

Debugging GUI interface.

xgraphs like function plot.

Change the setting used in generated plotfiles.

Call Trace option for text and html traces.

Batch Options #2

amrvis2dpp [<options>] [<filename(s)>]

----- amrproparser functions

[-nts n] sets number of time slots (default: 25600).

[-of fn] sets output file name.

[-proc n] sets processor number for single processor queries (default: 0).

[-prof] profile the parser.

~~[-proxmap] remap ranks to proximity ranks.~~

Resolution of timeline plotfile.

Most written batch results.

In “summaries”, change which processor fills “nCalls”.

Profile the profiler:
“bl_profprof”

Was used on Edison.
Deprecated now.

Batch Options #3

amrvis2dpp [<options>] [<filename(s)>]

----- amrproparser functions

~~[-redist]~~ redistribute files.

~~[-rra n]~~ sets refRatioAll.

~~[-stats]~~ print database statistics.

~~[-tce]~~ process only topolcoords for edison.

~~[-v n]~~ verbose: n can be 0, 1, or 2.

~~[-mff]~~ make filter file.

~~[-pff]~~ parse filter file.

Broken. Outline to redistribute bl_prof across different number of files.

Set refRatio for created plotfiles

Simple statistics of profiling database (e.g. nProcs used, which profiling types are available).

Deprecated Edison proximity mapping.

Profiler debugging information.

Deprecated batch parsing files. Should use GUI now.

Send/Receive (Not available)

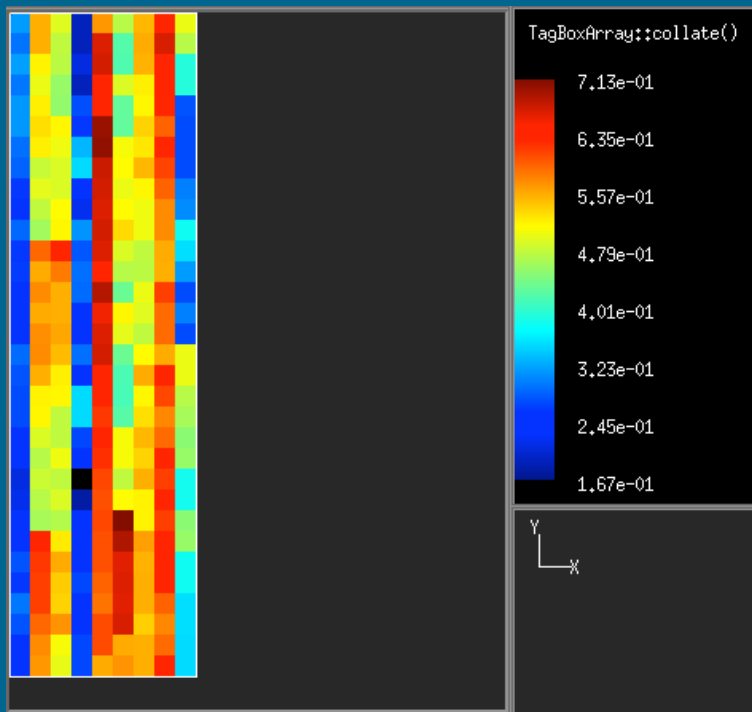
```
amrvis2dpp [<options>] [<filename(s)>]
----- amrparser functions
[-sr] process sends and receives.
-----
```

Currently commented out.
(Don't know why. Original version of "sends" that's been replaced?)

Will check it out and bring it back with MPI GUI.

All Call Times Plotfile

Processor #



Function Call #

```
amrvis2dpp [<options>] [<filename(s)>]
```

----- amrproparser functions

[-actpf f] output a plotfile for all call times for func f.
f is a quoted string.

f = function name used BL_PROFILE_VAR

Output files, named after the function:

- Plotfile of runtime for each time function is call for each proc.
- Xgraph of coefficient of variation for each call.

.xgr files are just space separated values.
Look at with xgraph *.xgr, or in your favorite plotter.

Grid Log Analysis

Analyses are sums over the entire grid log.
(All time steps)

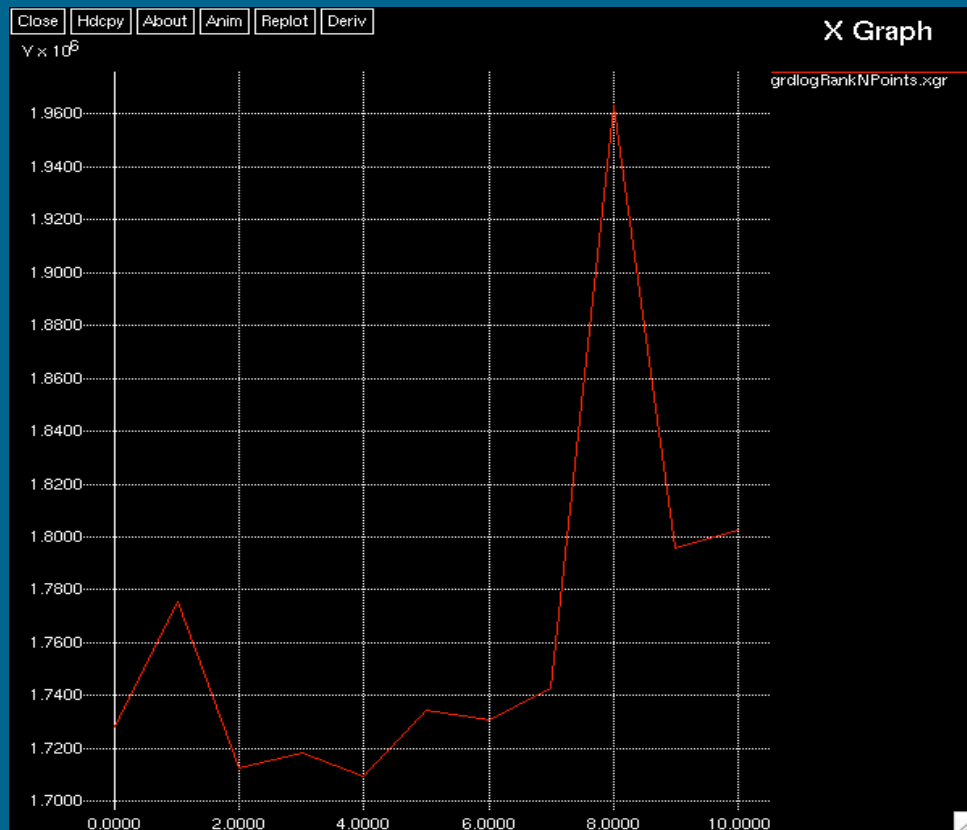
```
amrvis2dpp [<options>] [<filename(s)>]
----- amrparser functions
[-gl]     process only grdlog.
-----
```

Creates .xgr files of:

Number of points vs. MPI rank.
(grdlogRankNPoints.xgr)

Number of cells vs. number of grids of that
size.
(grdlogSizeNGrids.xgr)

Processed from the grdlog file generated by:
amr.gridlog = grdlog



Html Text Trace

```
amrvis2dpp [<options>] [<filename(s)>]
----- amrparser functions
[-html] write html.
[-htmlnc] write html showing ncalls.
-----
```

- CallTrace.html or CallTraceNC.html
- Complete, collapsible call trace in html with legend at top.
- Can be created with or without number of calls (affects time to create.)

Function calls (function number :: function name :: inclusive time :: exclusive time :: 1-e/i % :: ncalls :: callstackdepth)

- 0 :: VisMF::Initialize :: 1.62125e-05 :: 1.62125e-05 :: 0 % :: 1 :: 0
- 1 :: main() :: 23.1945 :: 0.120938 :: 99.4786 % :: 1 :: 0
 - 2 :: Amr::InitAmr() :: 0.0577853 :: 0.0577638 :: 0.0371335 % :: 1 :: 1
 - 3 :: Amr::initSubcycle() :: 2.14577e-05 :: 2.14577e-05 :: 0 % :: 1 :: 2
 - 4 :: Amr::init() :: 6.09591 :: 2.86102e-05 :: 99.9995 % :: 1 :: 1
 - 5 :: Amr::initialInit() :: 5.04044 :: 6.19888e-06 :: 99.9999 % :: 1 :: 2
 - 6 :: Amr::InitializeInit() :: 0.128591 :: 1.88351e-05 :: 99.9854 % :: 1 :: 3
 - 7 :: Amr::readProbinFile() :: 0.000195265 :: 0.000195265 :: 0 % :: 1 :: 4
 - 8 :: Amr::defBaseLevel() :: 0.128377 :: 5.126e-05 :: 99.9601 % :: 1 :: 4

1 - exclusive/inclusive

Sync Point Data

```
amrvis2dpp [<options>] [<filename(s)>]
----- amrparser functions
[-spd]    process sync point data.
-----
```

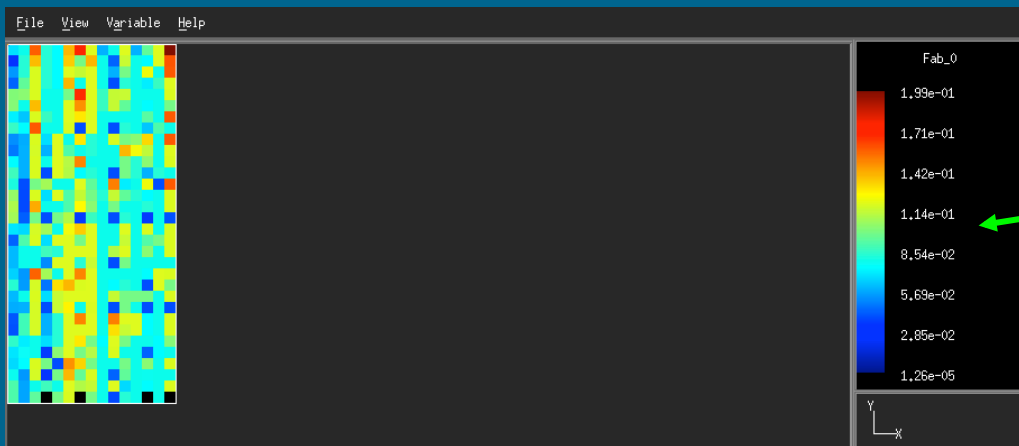
Creates fabs of:

Barrier wait times - bwaits.2d.fab
Barrier exit times - bexits.2d.fab
Reduction wait times - rwaits.2d.fab

Plot shows the **chosen time** for each barrier/reduction on each processor.

```
amrvis2dpp -spd bl_prof
```

Processor #



Barrier #

In Conclusion: **Let Us Know!**

Now that you know what tools are available, start using the profiling tools to analyze your code!

- If you need a little more information, **let us know!** We'll add new features.
- If you find a bug, **let us know!** We'll fix it ASAP.
- If you need a depreciated or broken feature, **let us know!** We'll make fix it or find an appropriate alternative.
- If the profiling tools help you, **let us know** for success stories in upcoming publications!

For Profiling Help Contact:

Kevin Gott:

kngott@lbl.gov