Multi-Application Online Profiling Tool

Vi-HPS

Julien ADAM, Antoine CAPRA

ParaTools

About MALP

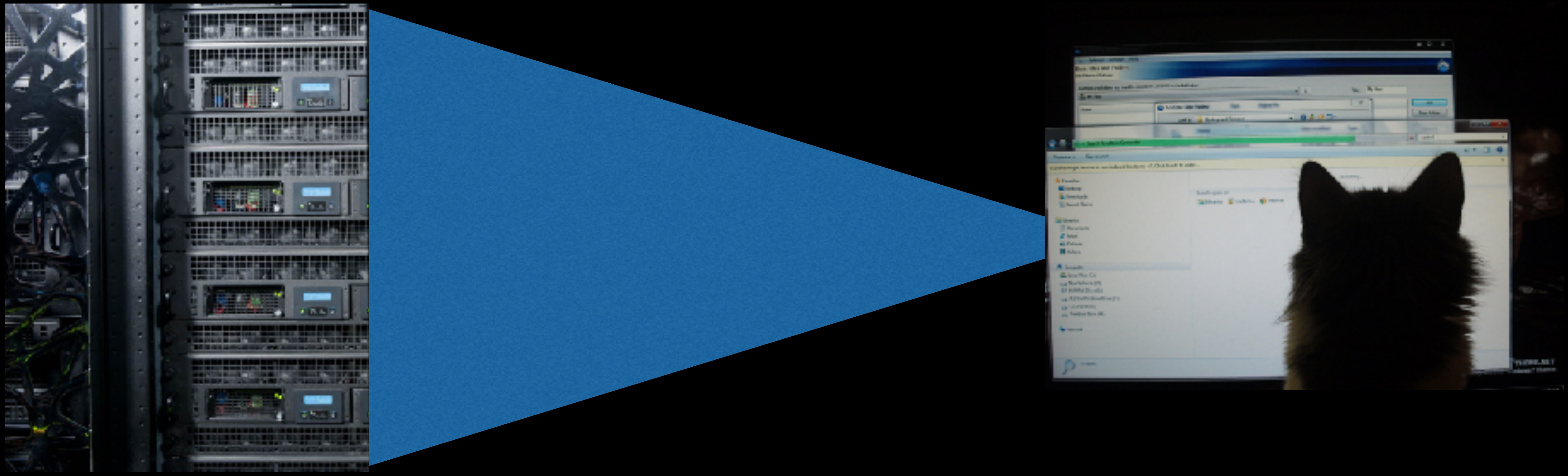
- MALP is a tool originally developed in CEA and in the University of Versailles (UVSQ)
- It generates rich HTML views describing parallel programs

- MALP relies on the on-line profiling approach (fine-grained events are reduced during the execution)
- MALP is currently MPI oriented with a focus on the topology and some temporal analysis

ParaTools

MALP's trade-offs in the Measurement Process

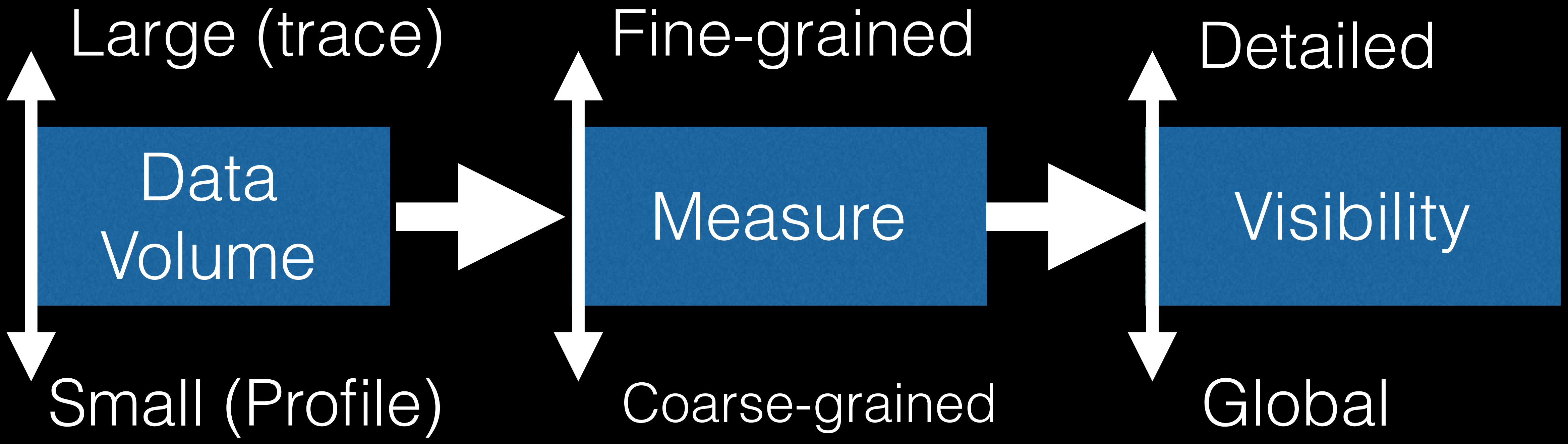
The Measurement Chain



Representing the state of millions of cores necessarily require a form of **reduction**, at least to acknowledge user's **cognitive limitations**.

ParaTools

The Measurement Chain

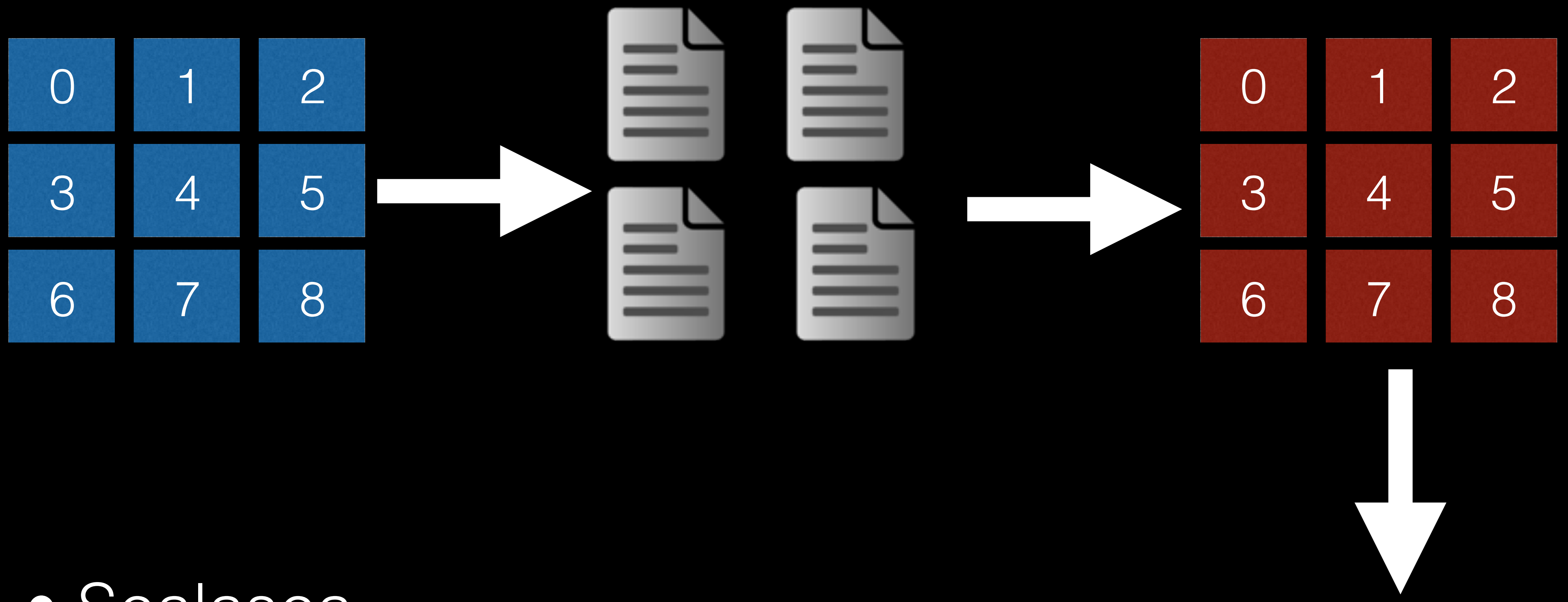


A chain **collecting** and **valorizing** data

Compromise between details and measurements' scalability

ParaTools

Trace-Based Approach

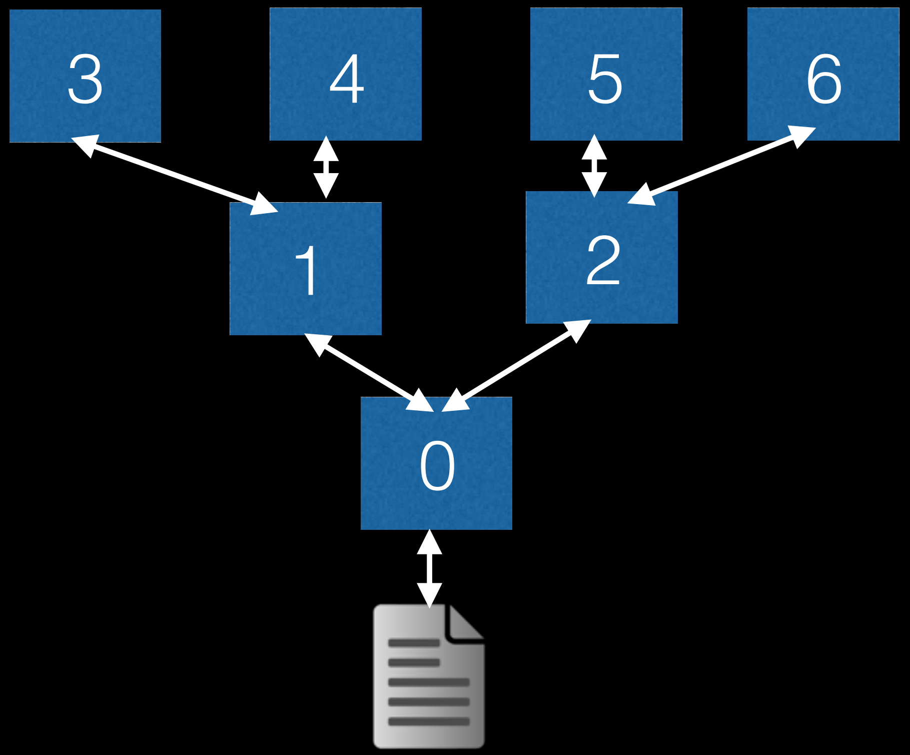


Report / Interface

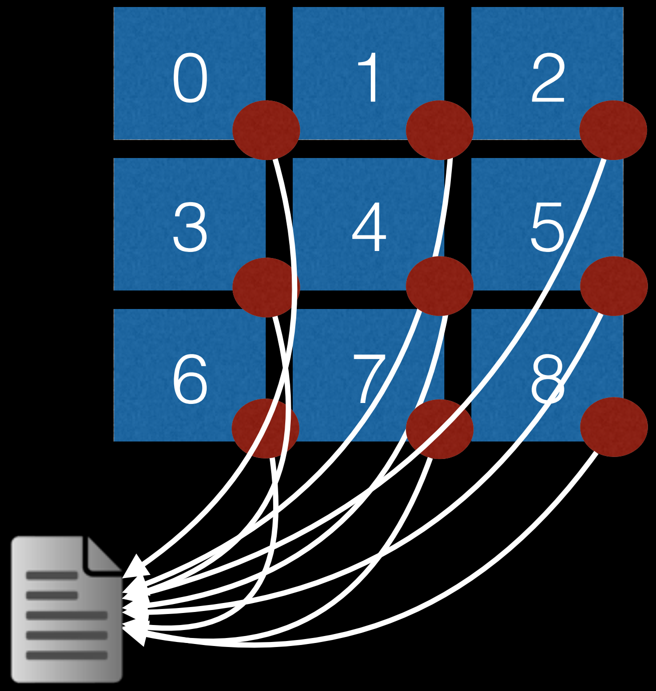
- Scalasca
- Vampir Trace
- TAU

ParaTools

On-Line



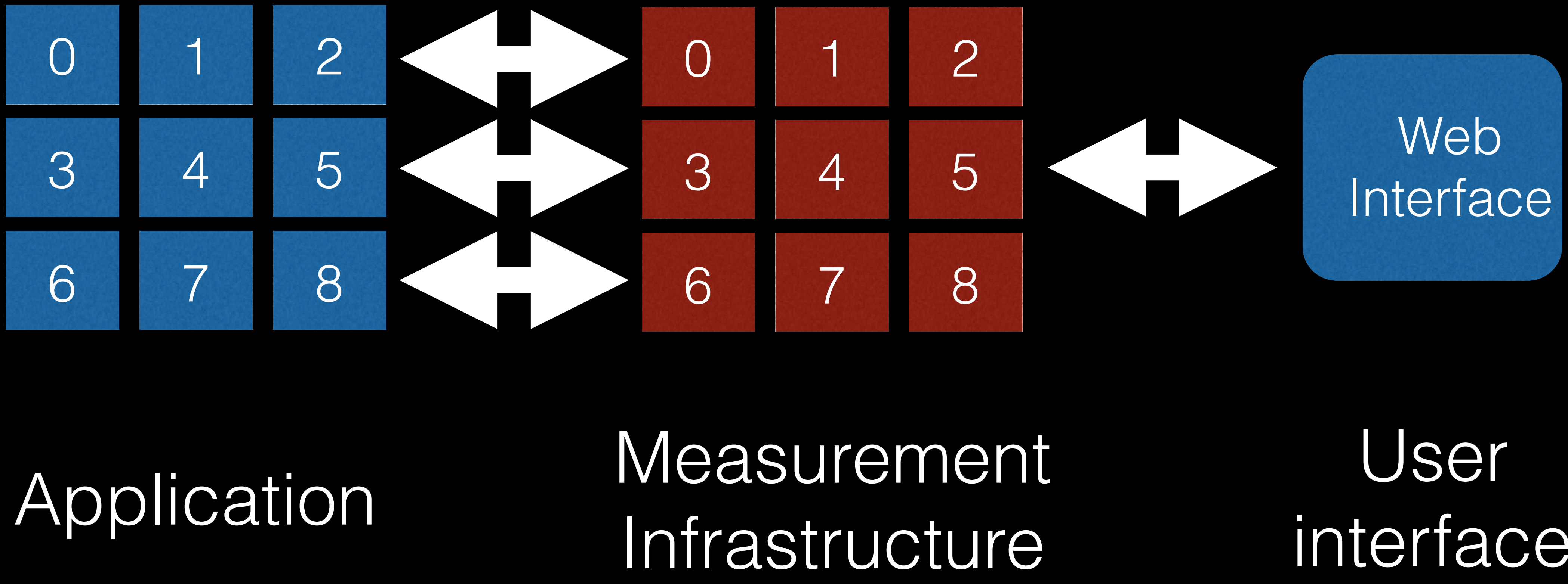
Tree-Based Overlay Network (TBON)
- Periscope
- MrNet/Stat (Debug)
- MUST (Validation)
- DDT/Totalview/MAP



Profiles:
- Mpi-P
- Scalasca (partly)

ParaTools

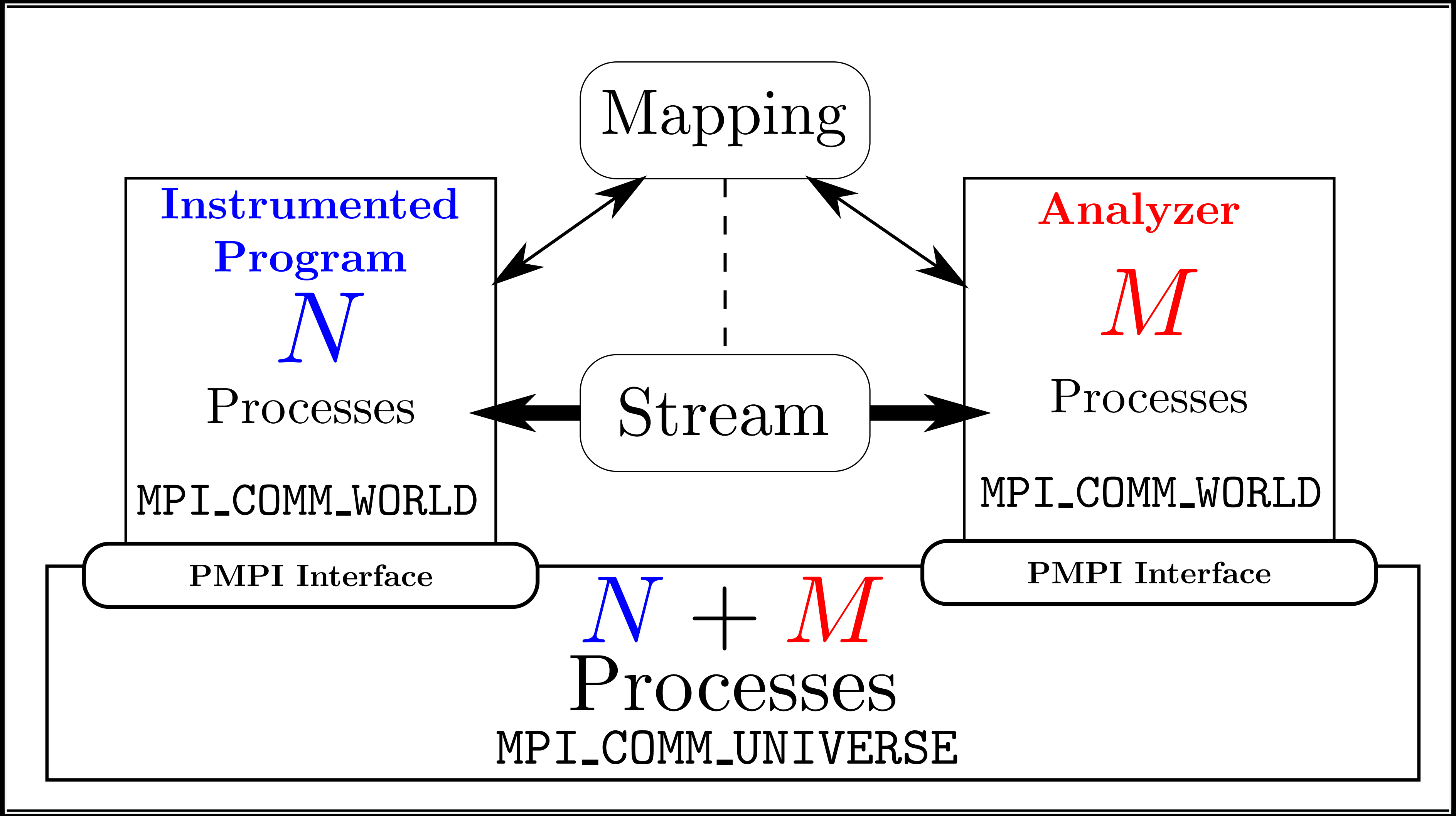
MALP



Idea: allocate cores to analyze and valorize data (reduction).

ParaTools

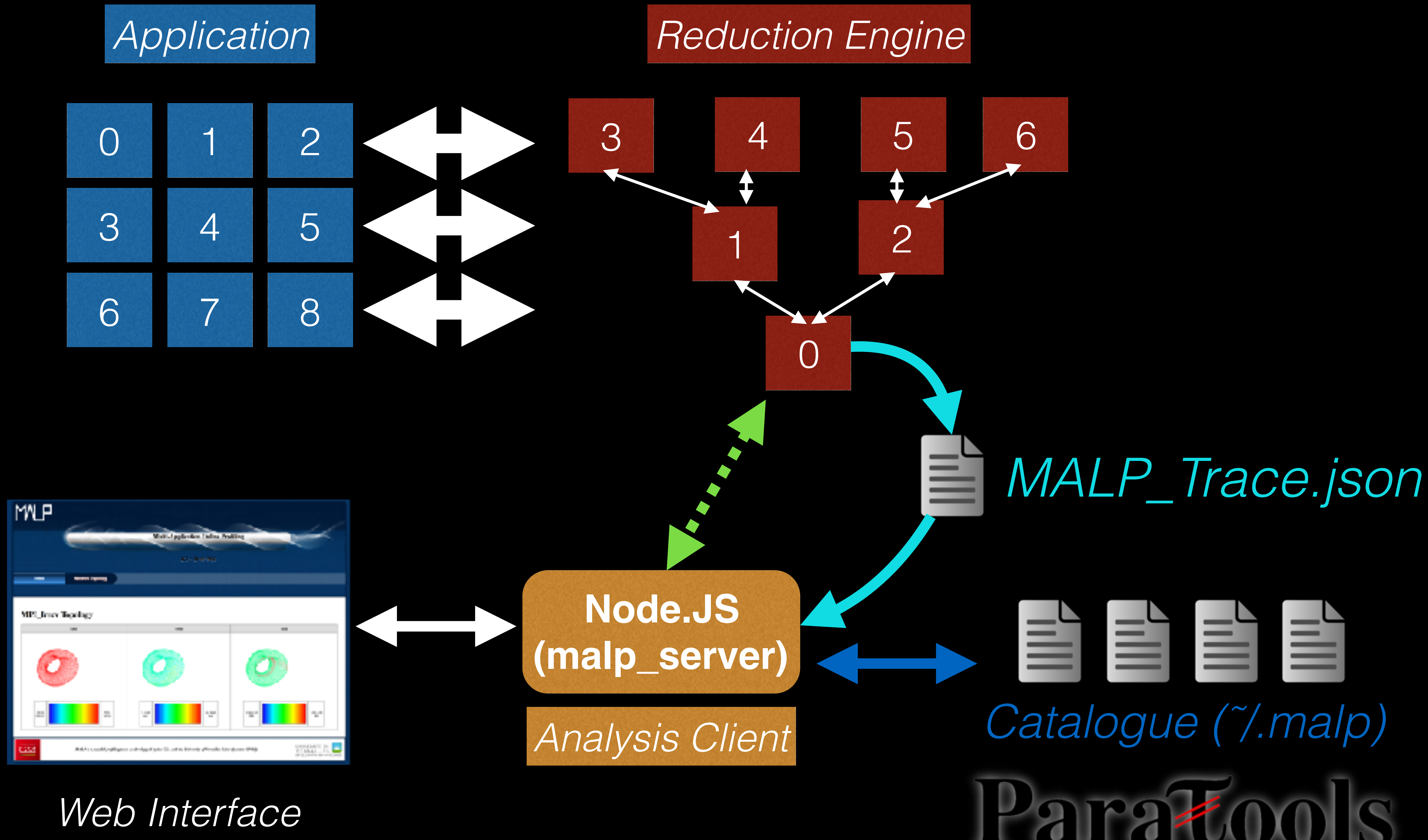
Instrumentation Analysis Coupling
Overview



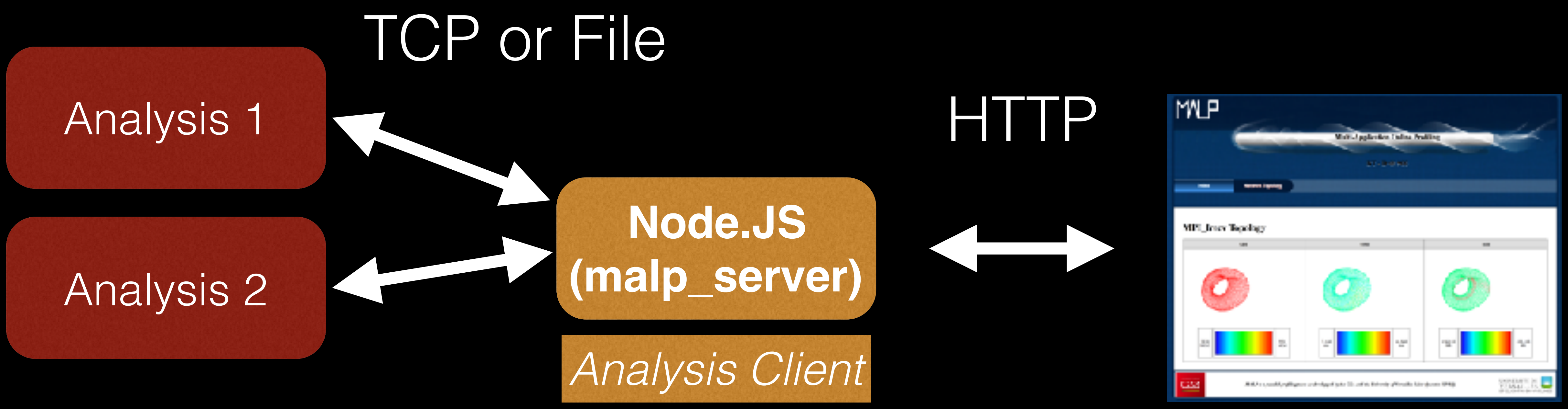
MPI Virtualization: an original idea from the P^{N}MPI paper

ParaTools

Coupling with the Interface



Closer look on the Interface Side



Dynamic Views

MALP vs Others

Pros

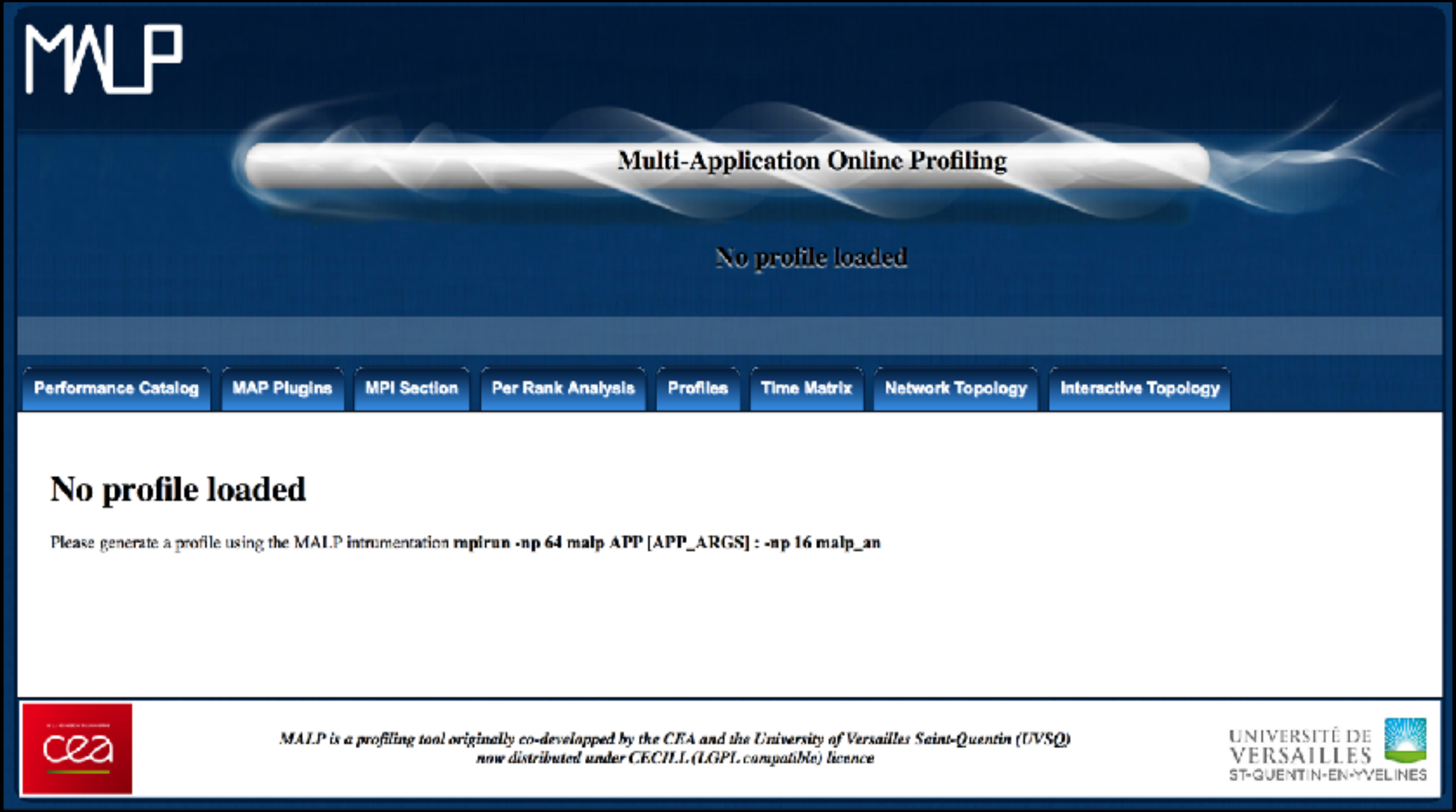
- MALP is easy to use (preload only)
- Analysis is done online (small data footprint)
- Performance data rendering is done in a browser
- Some original topological analysis
- Overhead is relatively small

Cons

- Due to its approach MALP presents reduced data
- Exhaustive program behavior is not provided by MALP for this use a trace-based approach
- Extra processing power is required during the execution

Outputs from MALP

Closer look on the Interface Side



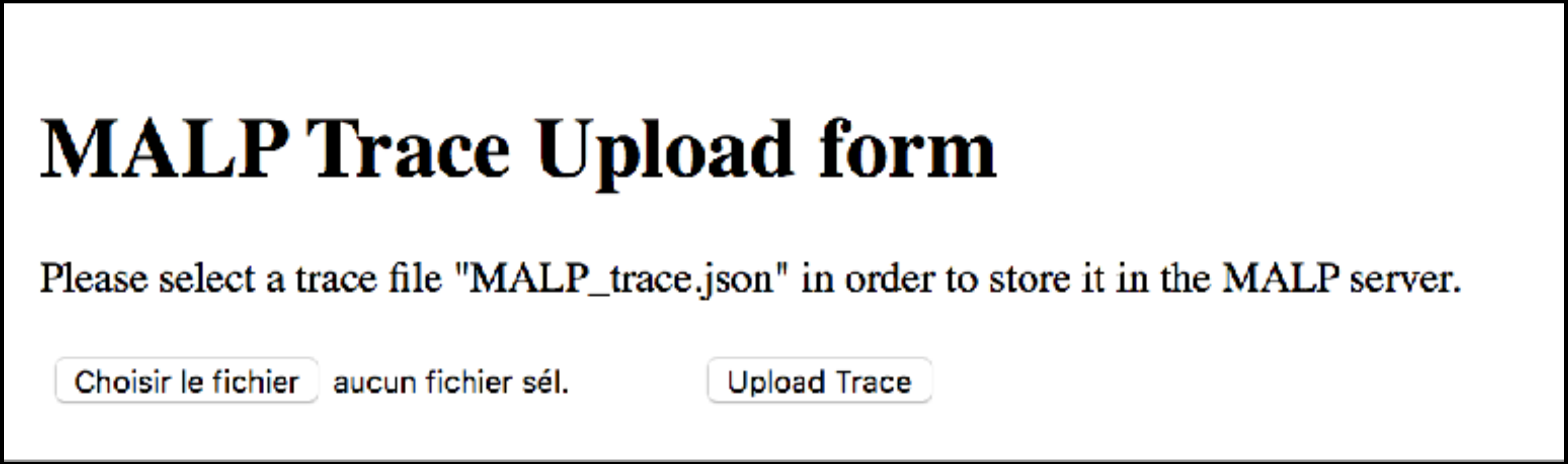
No profile loaded

Please generate a profile using the MALP intrumentation **mpirun -np 64 malp APP [APP_ARGS] : -np 16 malp_an**

MALP is a profiling tool originally co-developped by the CEA and the University of Versailles Saint-Quentin (UVSQ) now distributed under CECILL (LGPL compatible) licence

ParaTools

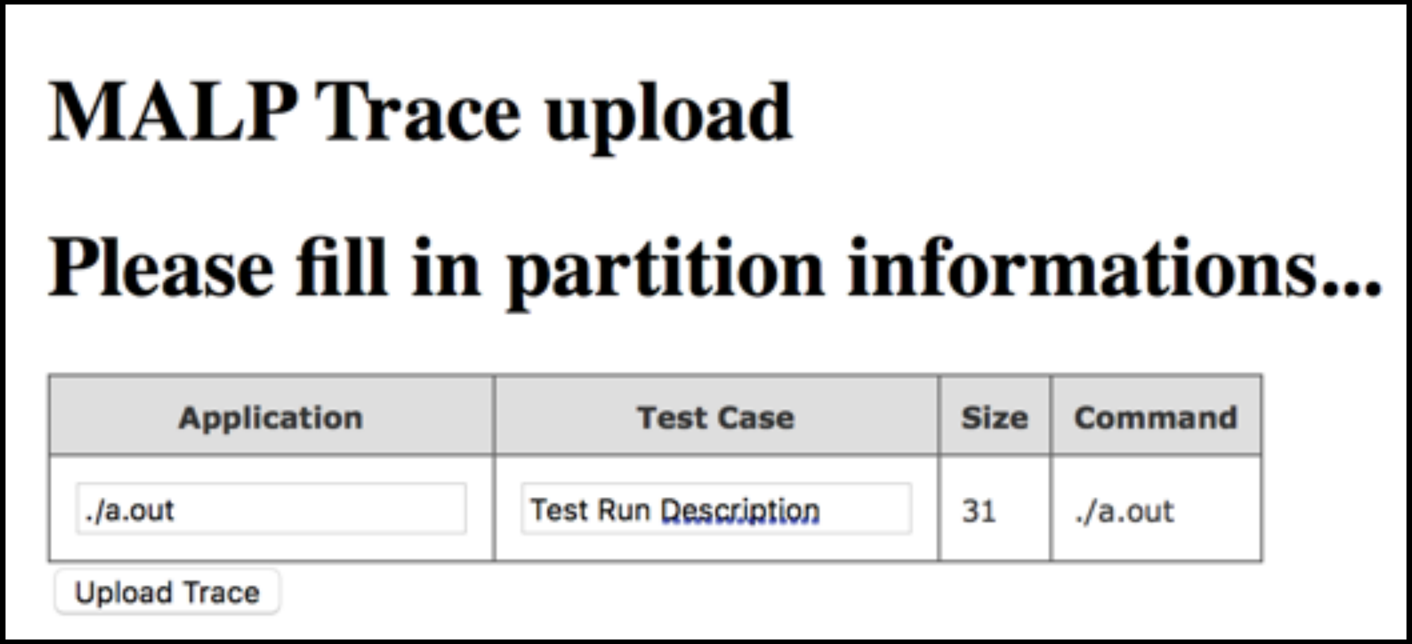
The Upload Form



MALP Trace Upload form

Please select a trace file "MALP_trace.json" in order to store it in the MALP server.

The Meta-Data Form

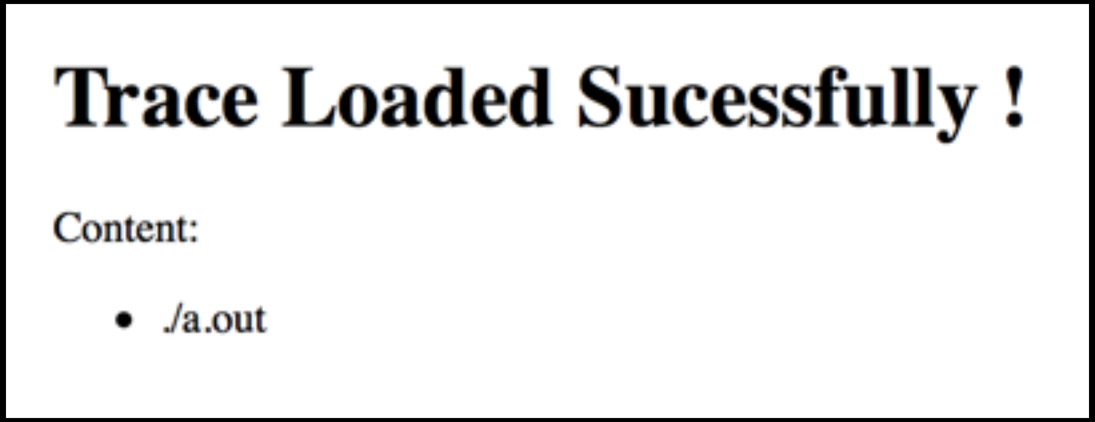


MALP Trace upload

Please fill in partition informations...

Application	Test Case	Size	Command
./a.out	Test Run Description	31	./a.out

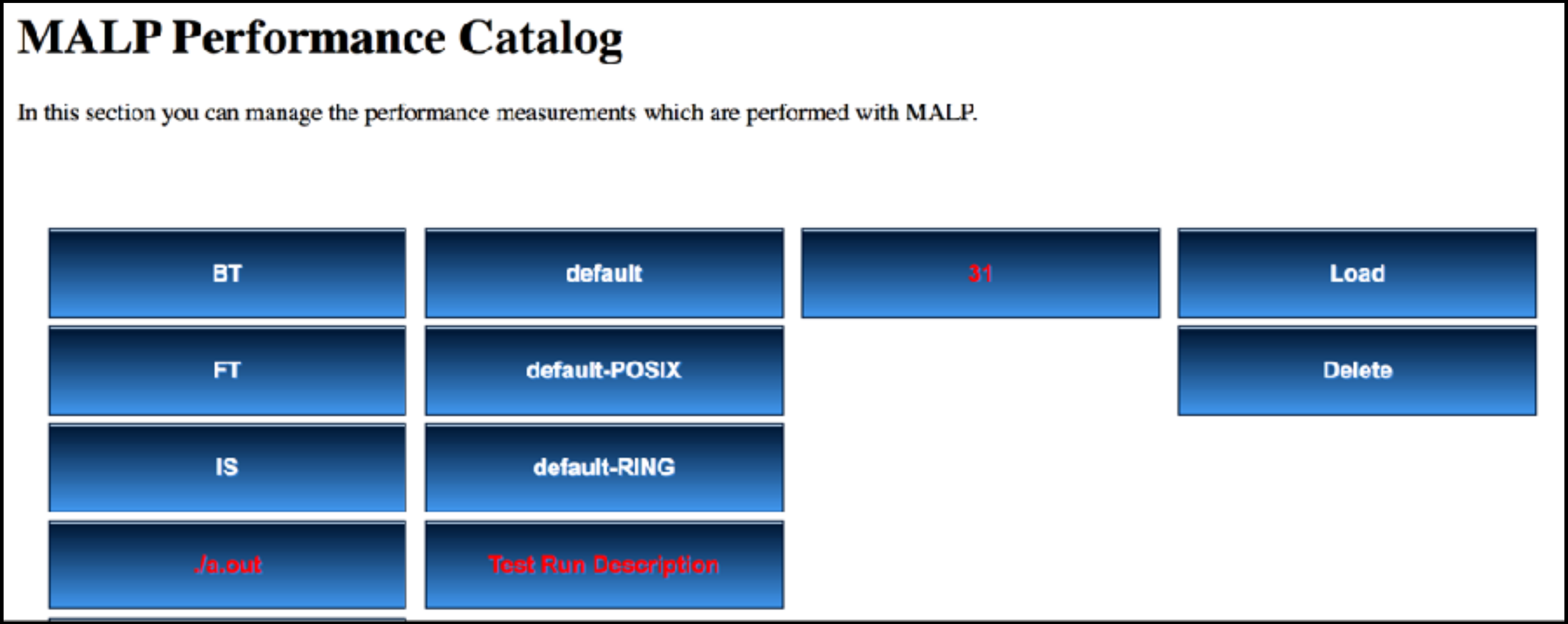
Trace Uploaded !


Trace Loaded Sucessfully !

Content:

- ./a.out

The Performance Catalog

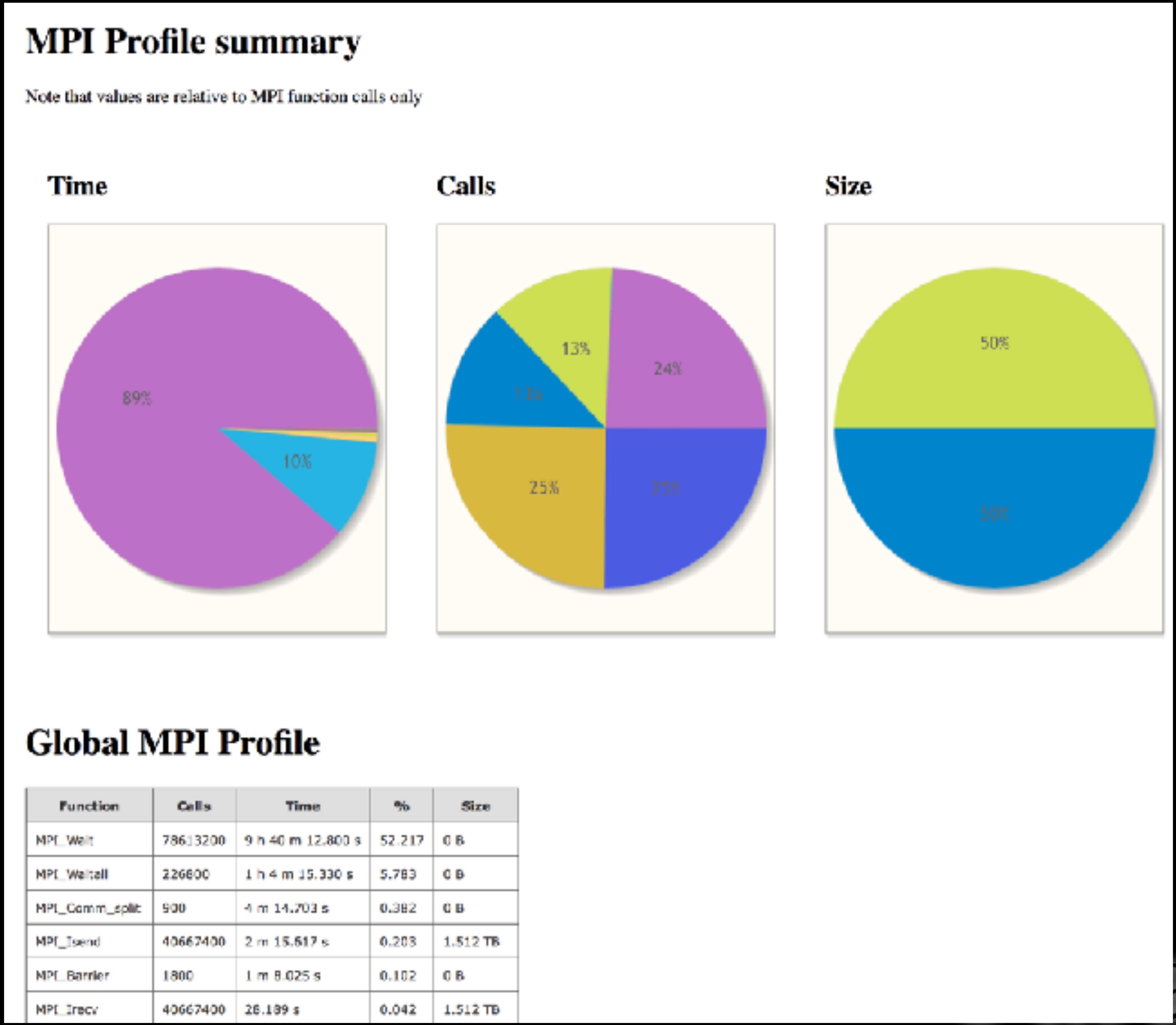


MALP Performance Catalog

In this section you can manage the performance measurements which are performed with MALP.

ParaTools

MPI Profile



MPI Profile summary

Note that values are relative to MPI function calls only

Time

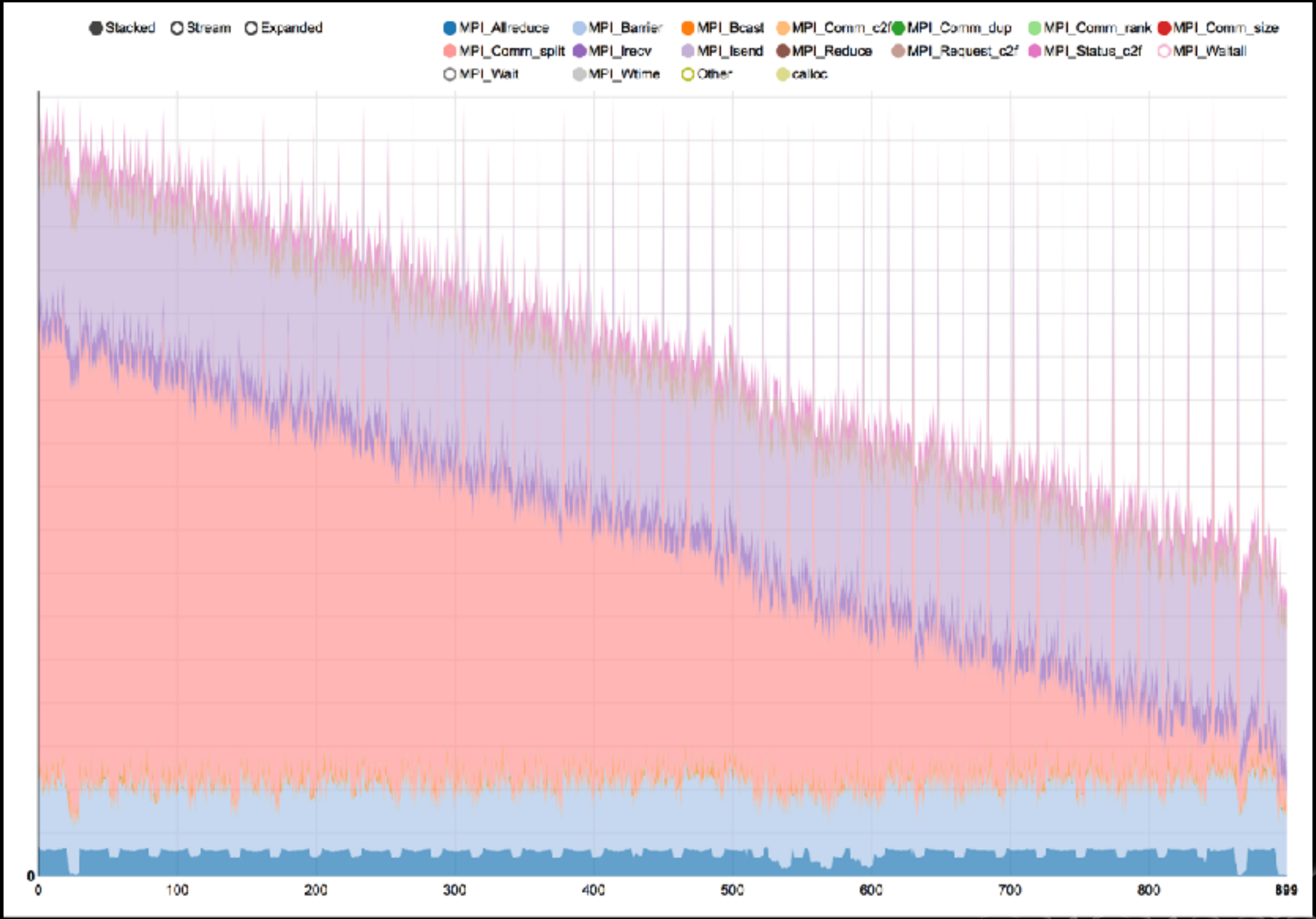
Calls

Size

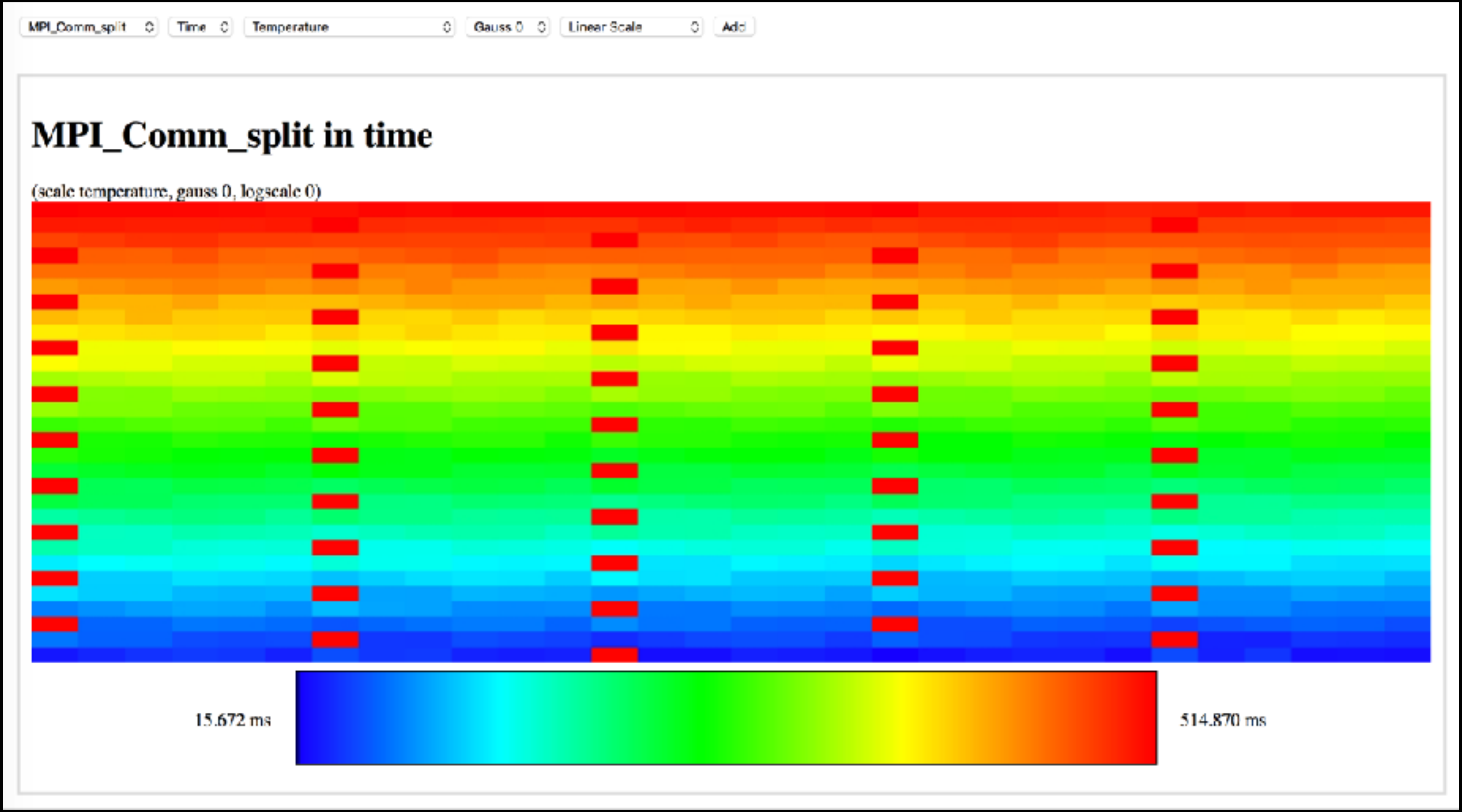
Global MPI Profile

Function	Calls	Time	%	Size
MPI_Wait	78613200	9 h 40 m 12.800 s	52.217	0 B
MPI_Waitall	226800	1 h 4 m 15.330 s	5.783	0 B
MPI_Comm_split	900	4 m 14.703 s	0.382	0 B
MPI_Isend	40667400	2 m 15.617 s	0.203	1.512 TB
MPI_Barrier	1800	1 m 8.025 s	0.102	0 B
MPI_Irecv	40667400	28.189 s	0.042	1.512 TB

« Per Rank » Analysis



« Rank Matrix » Analysis

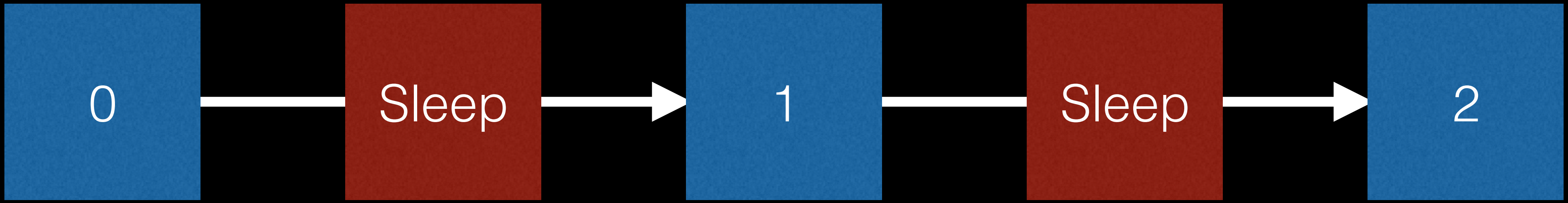


MPI_Comm_split in time

(scale temperature, gauss 0, logscale 0)

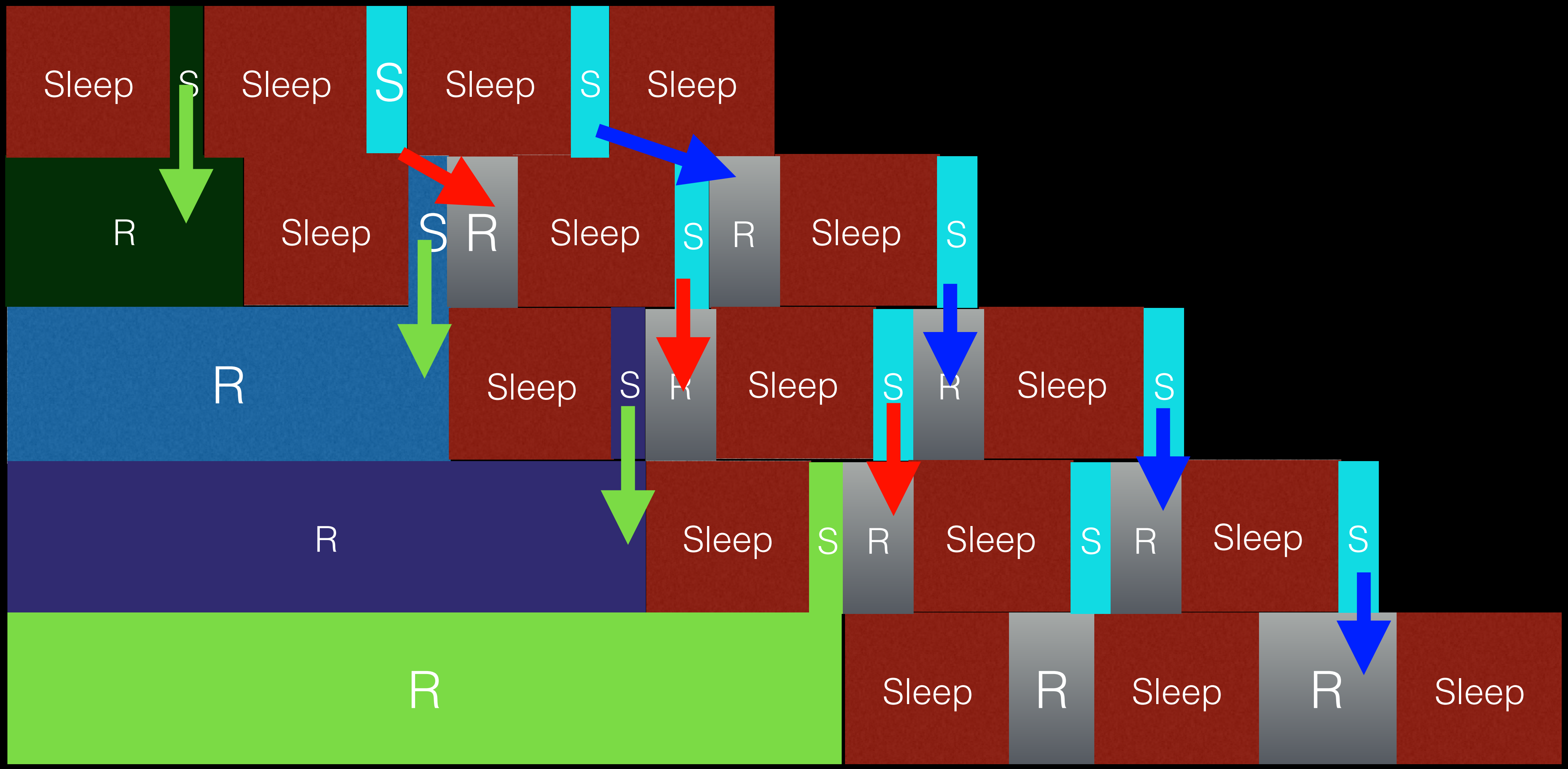
« Time Matrix » Analysis

Consider this example where you send a message to your right rank after waiting



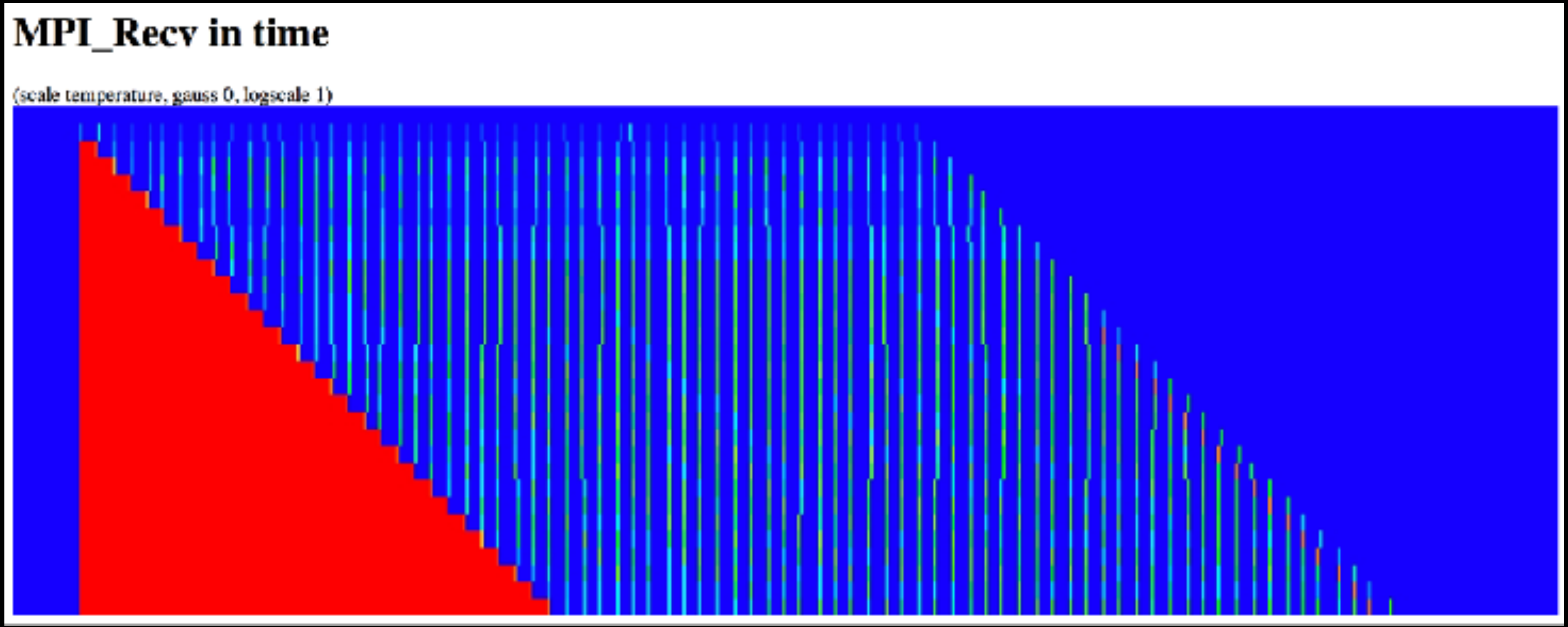
```
MPI_Recv( &value, 1, MPI_INT, rank - 1, 0, MPI_COMM_WORLD, &status );
usleep(XXX);
if (rank < size - 1)
    MPI_Send( &value, 1, MPI_INT, rank + 1, 0, MPI_COMM_WORLD );
```

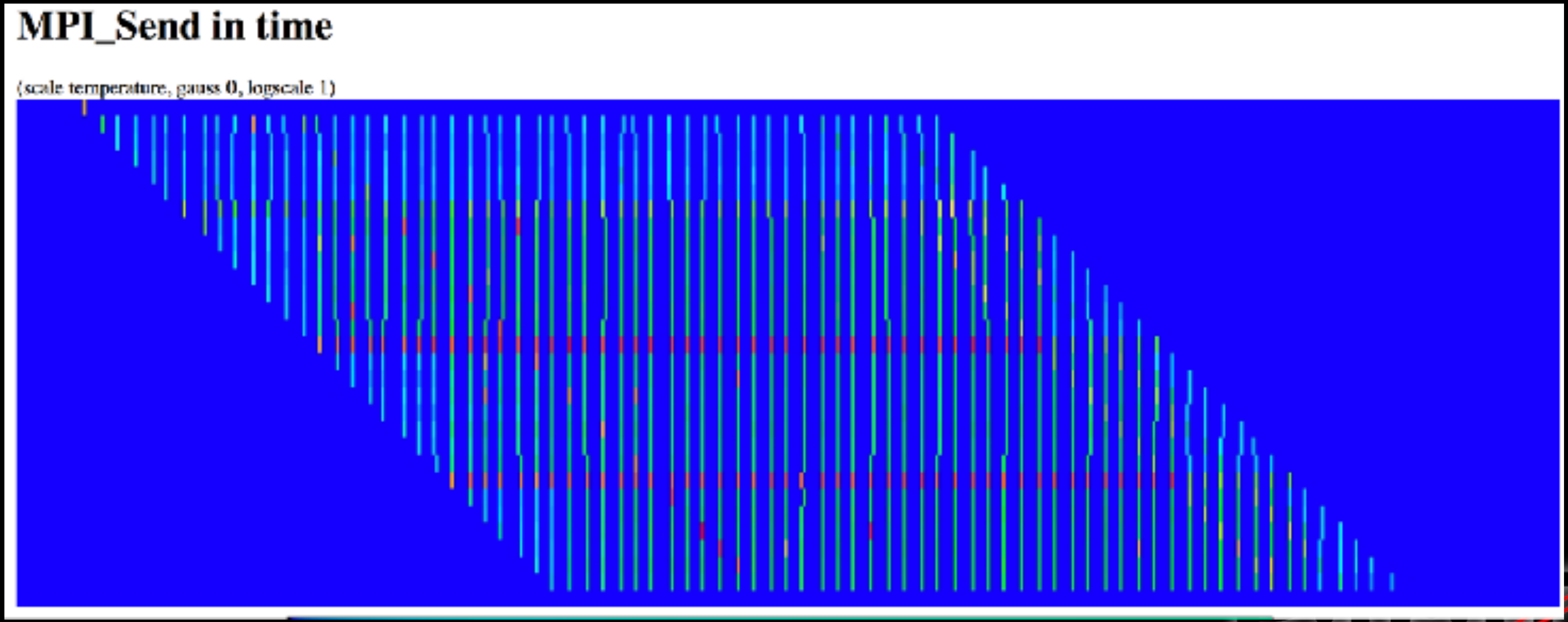
« Time Matrix » Analysis



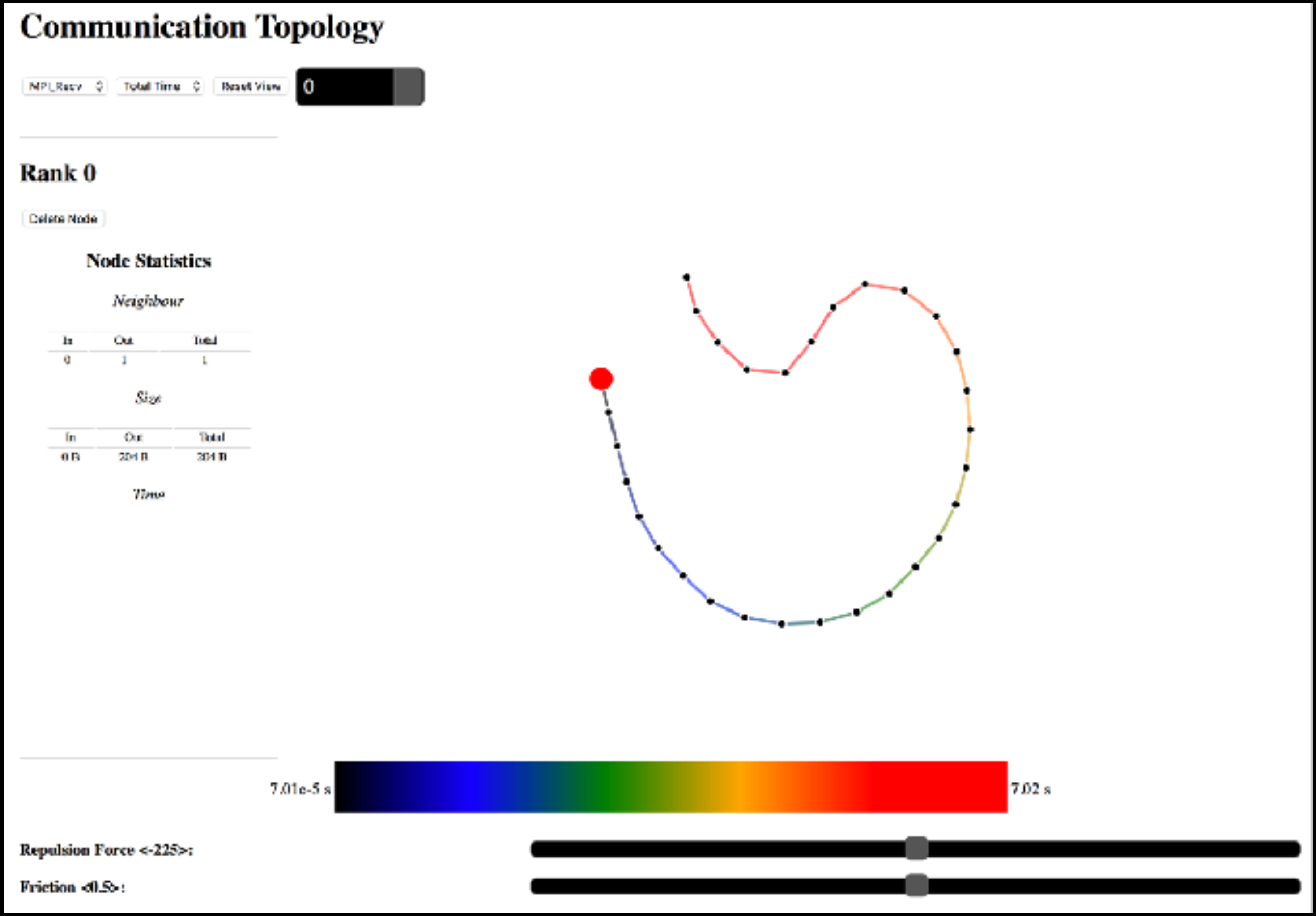
ParaTools

« Time Matrix » Analysis





« Interactive Topology » Analysis



Communication Topology

Rank 0

Node Statistics

Neighbour

In	Out	Total
0	1	1

Size

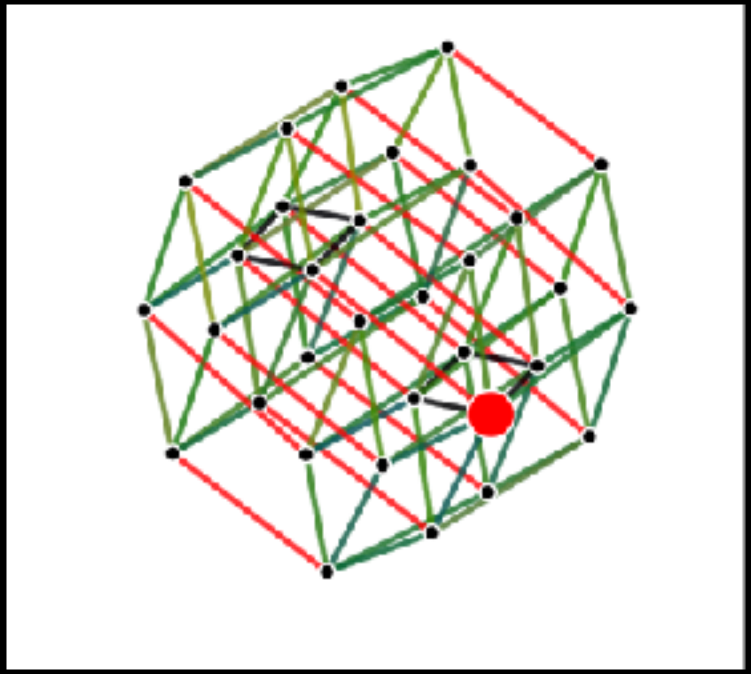
In	Out	Total
0 B	204 B	204 B

Time

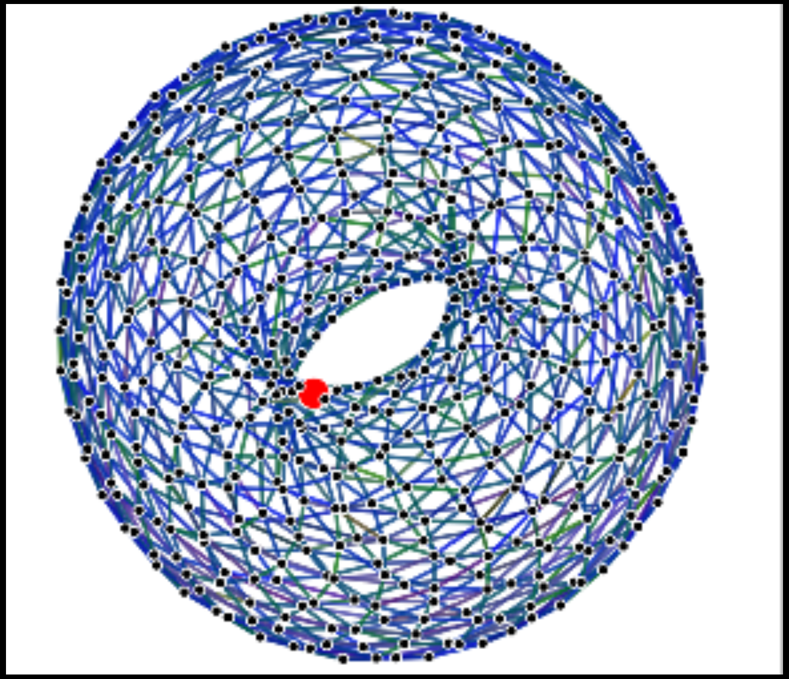
Repulsion Force <-225>:

Friction <0.5>:

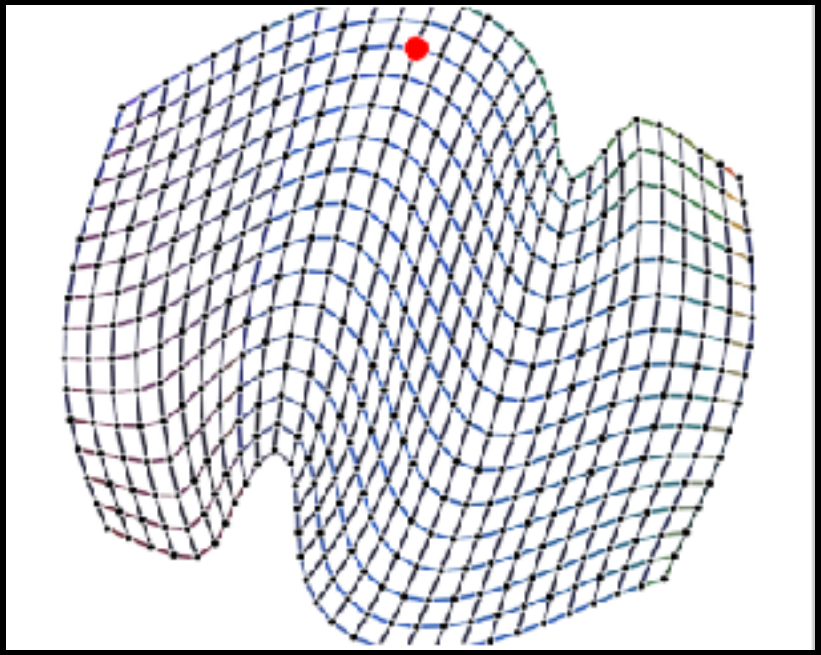
« Interactive Topology » Analysis



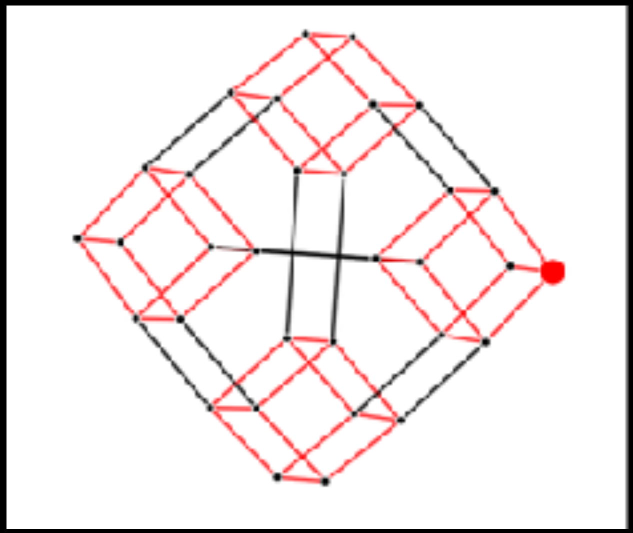
MG



BT



LU

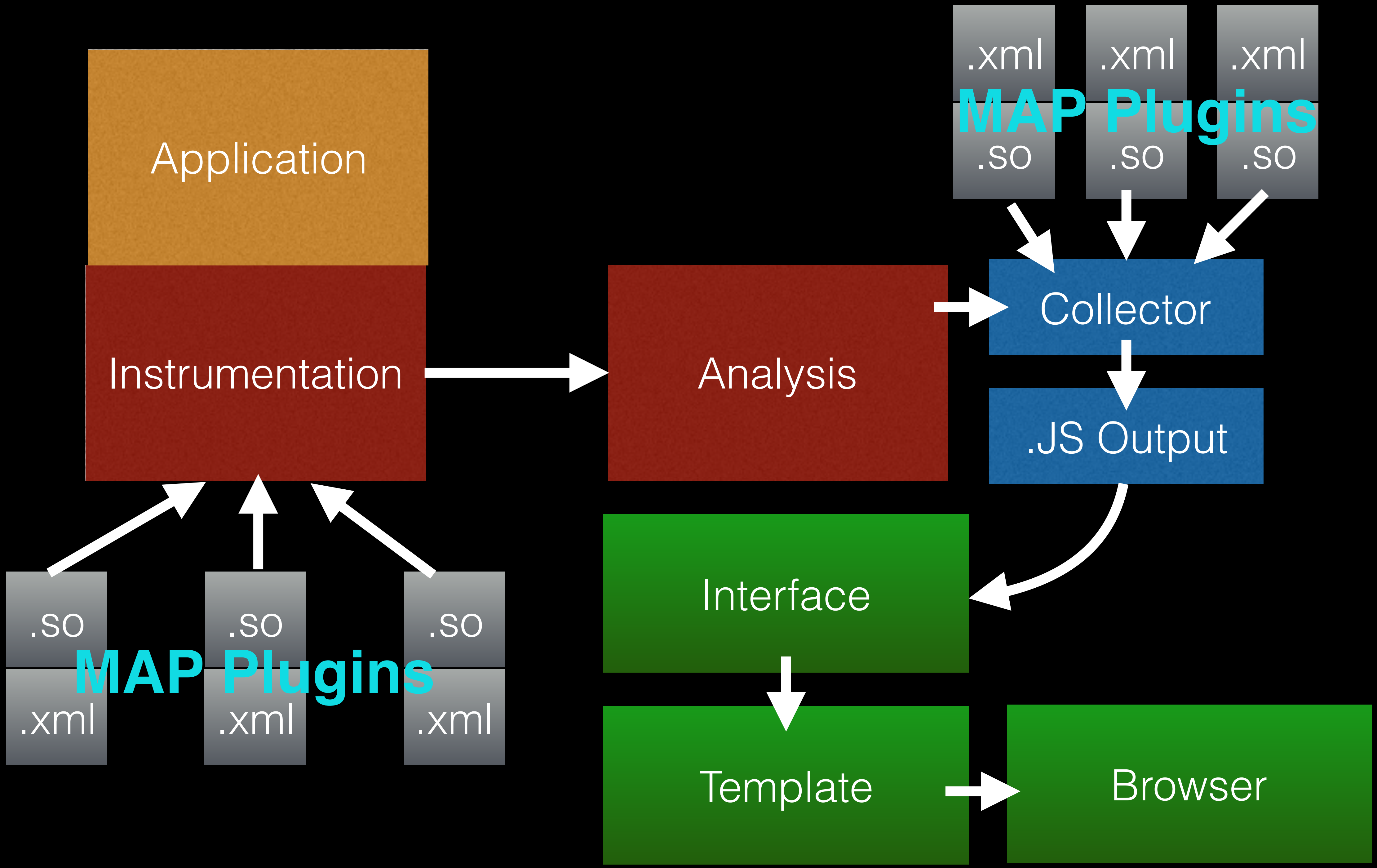


CG

ParaTools

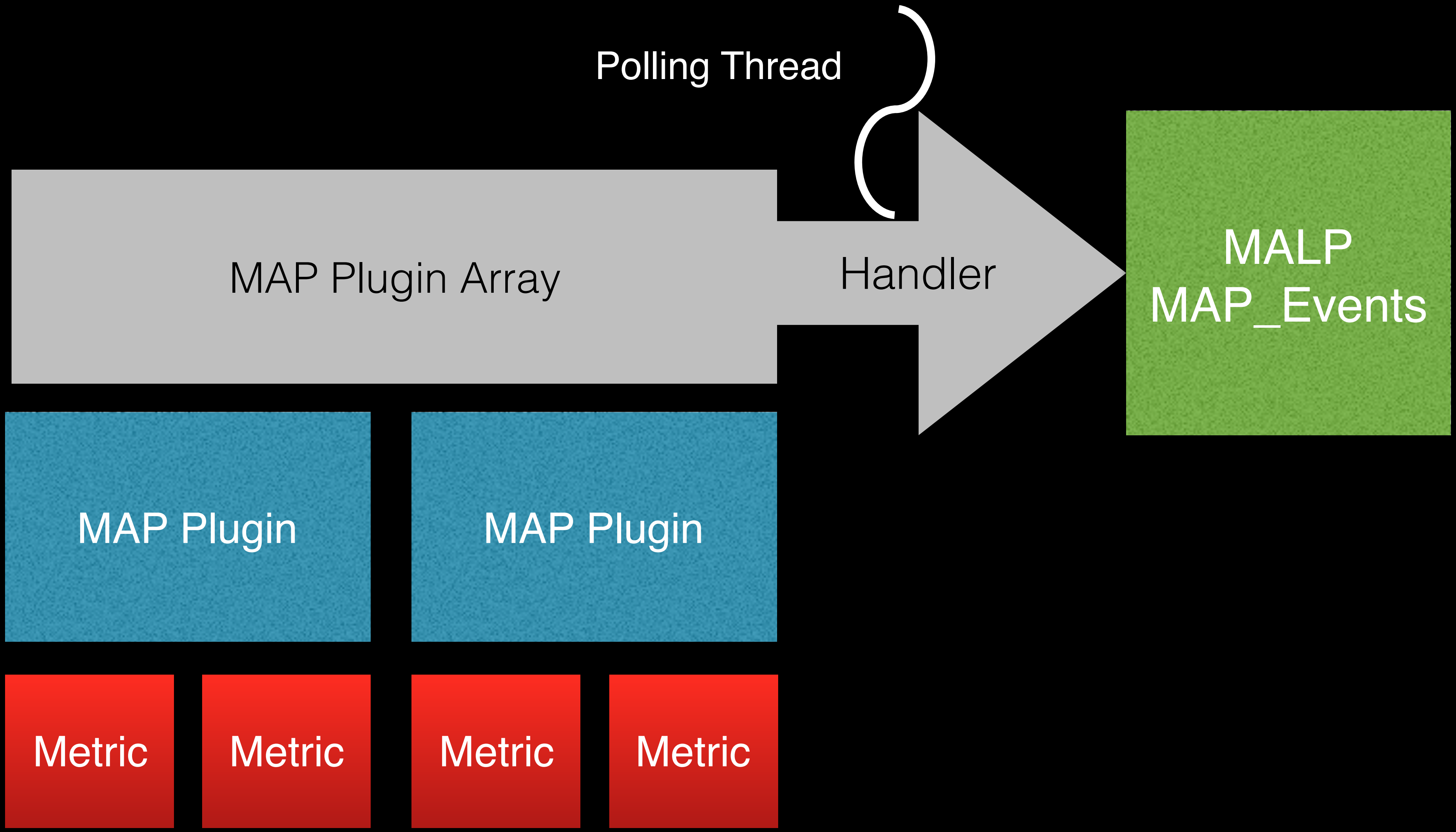
Allinea MAP Support

MALP can consume MAP events



ParaTools

MALP can consume MAP events

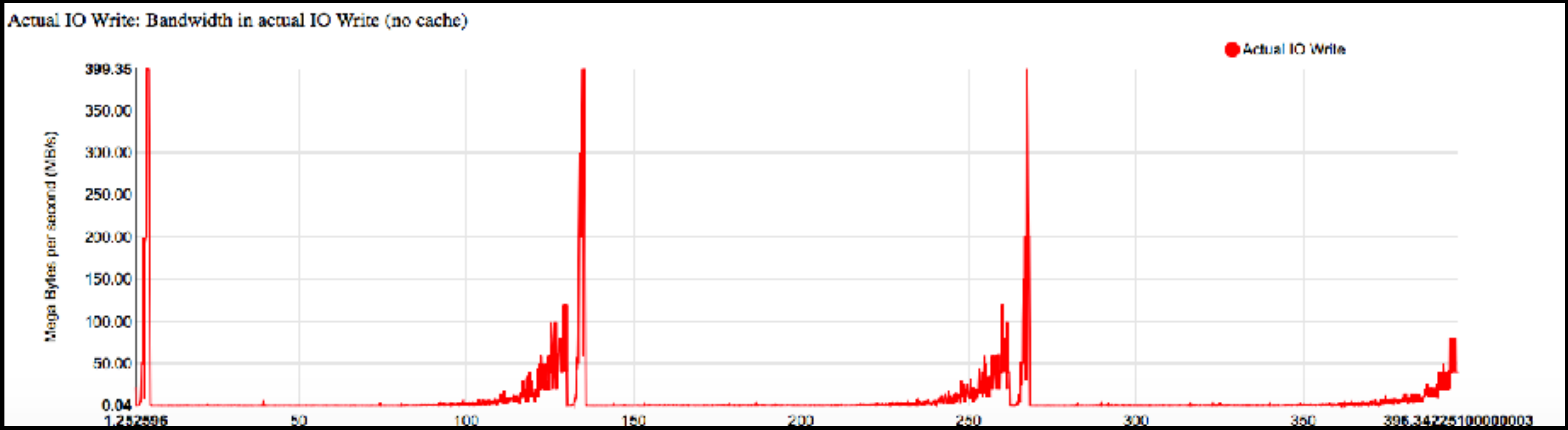


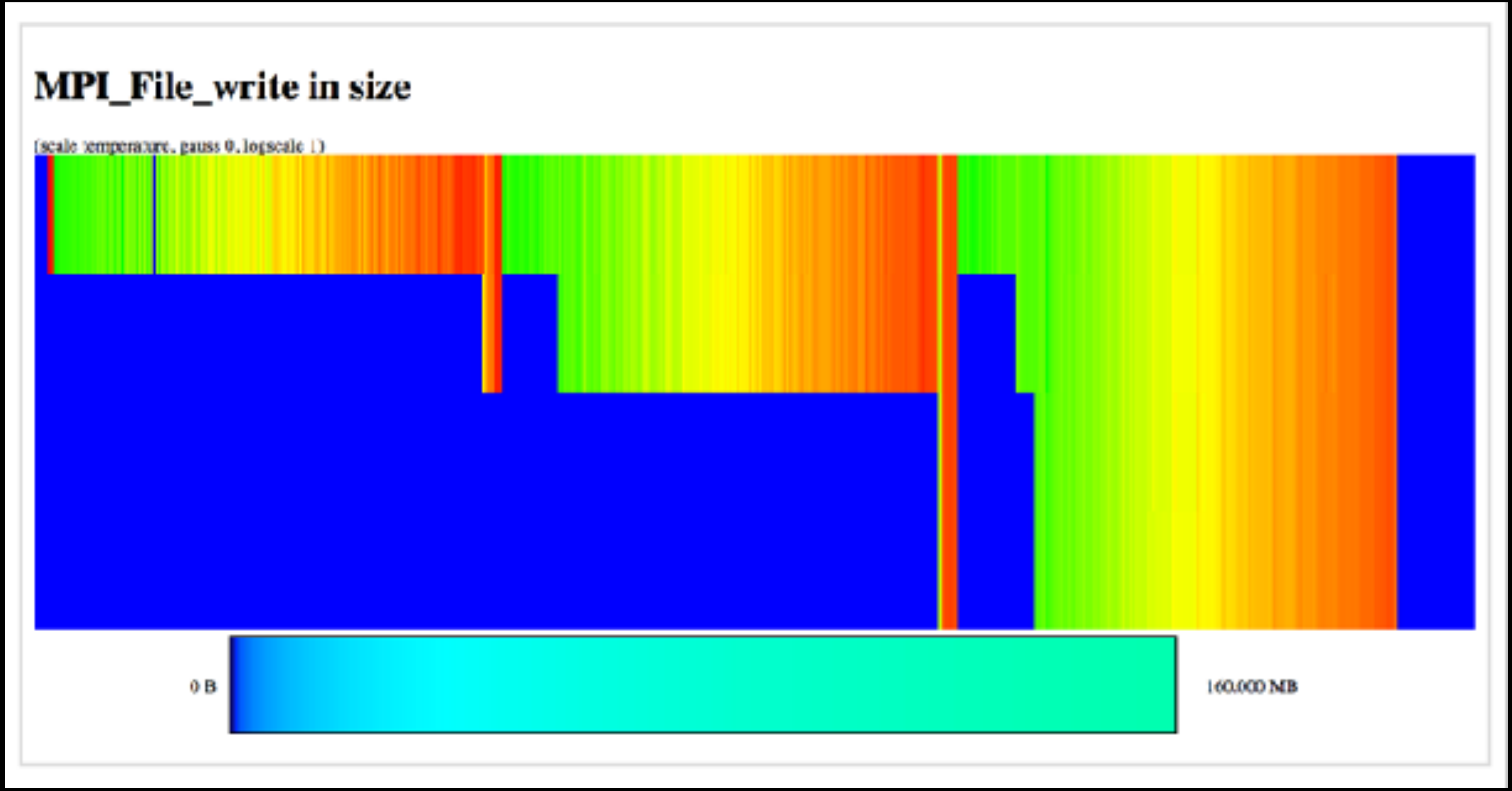
ParaTools

Sample MAP IO stat Collector

```c
int rchar_collect(metric_id_t metric_id,
                  struct timespec *in_out_sample_time,
                  double *out_value)
{
    struct io_stat st;

    if( read_io_stat( &st ) )
    {
        allinea_set_metric_error_messagef(metric_id, errno,
        "Error opening /proc/self/io\n");
        return;
    }

    *out_value = ( st.rchar - prev_values.rchar) / (1024.0 * 1024.0);
    prev_values.rchar = st.rchar;

}
```

MPI-IO Write Bandwith Example





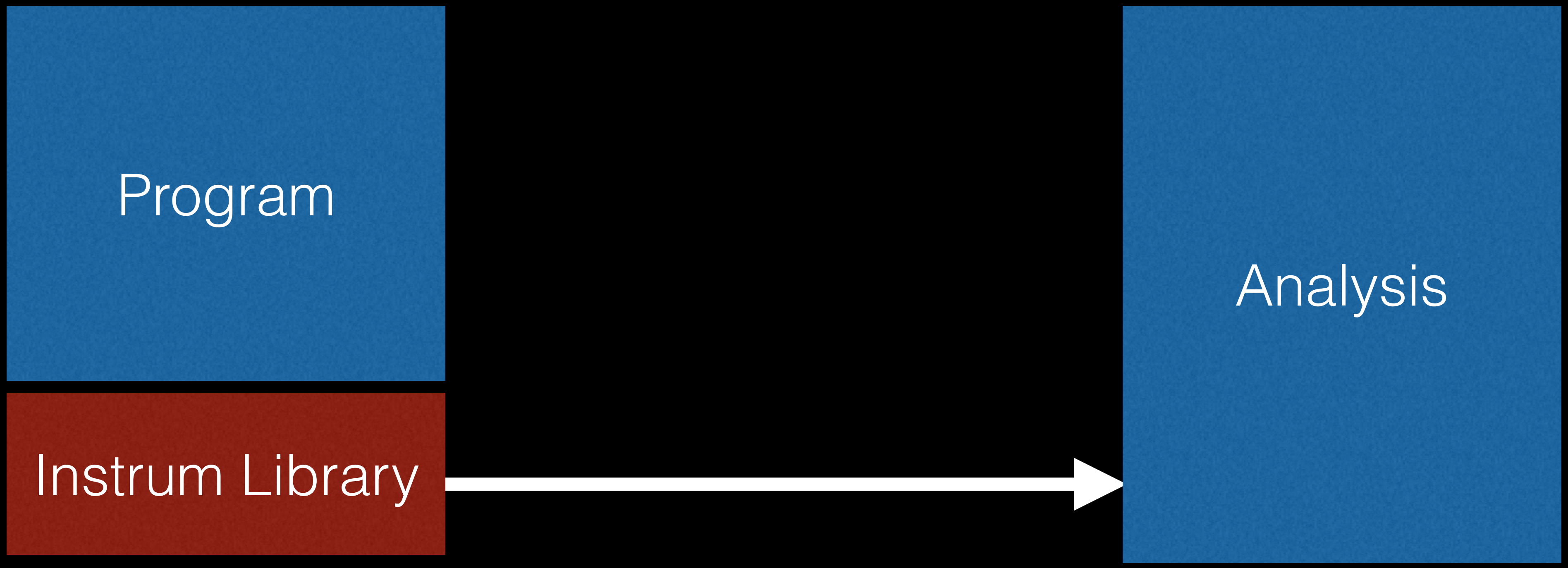
IMB-IO
MALP « time matrix »
and MAP timeline

How to use MALP ?

Measurement Process in MALP

Two applications have to be co-launched:

- The instrumented program (LD_PRELOAD)
- The analysis engine



ParaTools

Measurement Process in MALP

```
malpl 16 8 CMD [ARGS]
```

Will launch 18 processes
(16 processes plus an analysis process for 8 instrumented)

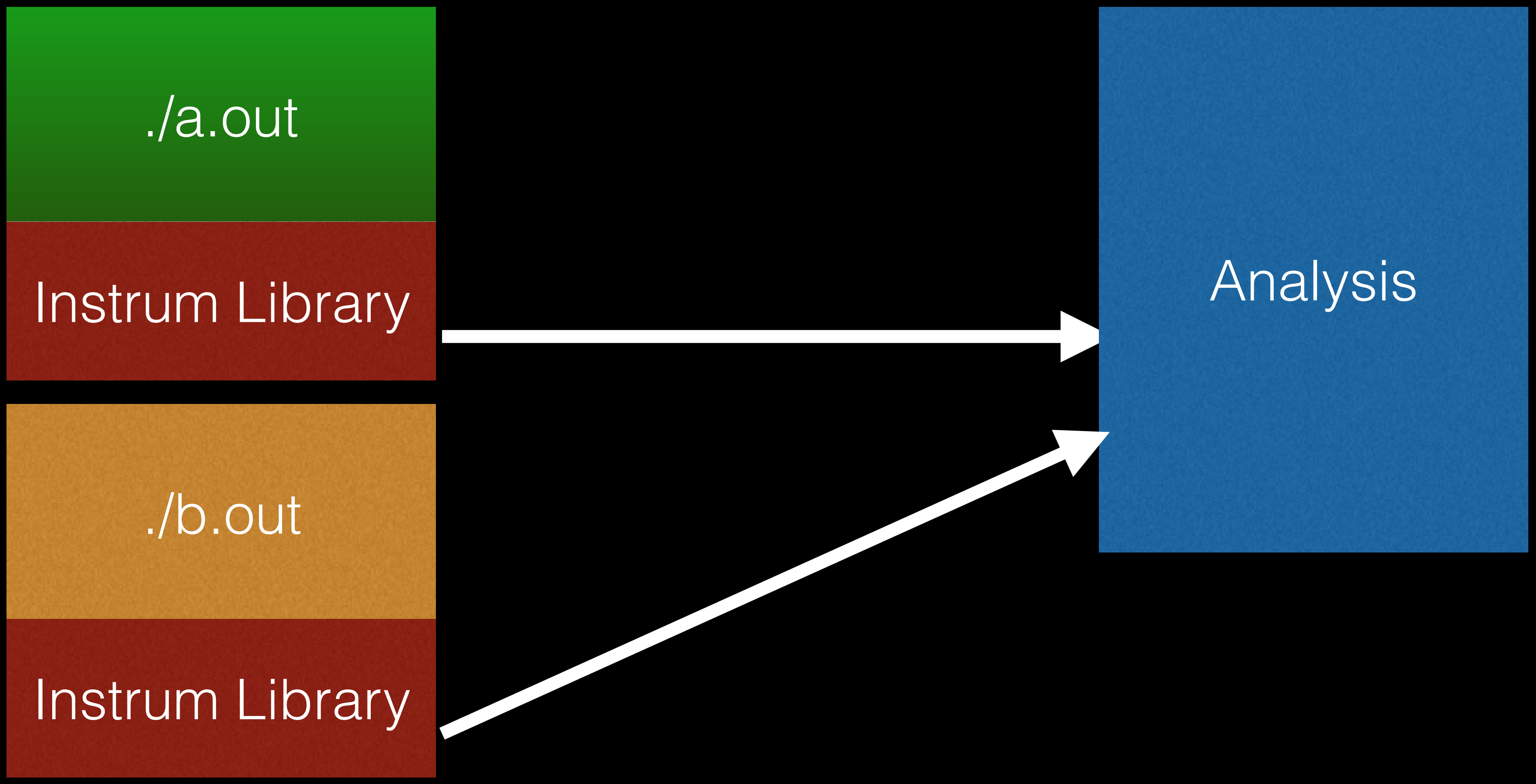
```
mpirun -np 16  malp CMD [ARGS] : -np 2 malp_an
```

Will launch 18 processes
(same as above classical MPMD syntax)

Instrumenting Multiple-Applications

```
mpirun  -np 16  malp ./a.out [ARGS] \
        : -np 16 malp ./b.out \
        : -np 4 malp_an
```

Will launch 36 processes



Enable MAP Plugin Support

Simply point MALP to your plugin directory:

```
export MAP_PLUGIN_PREFIX=$PREFIX/lib/malp_map_plugins/
```

You may change sampling frequency (milliseconds):

```
export MAP_PLUGIN_FREQ=10000
```

If correctly enabled the following should appear when running the target program:

```
## Loading MAP plugins :
Loading plugin 'interrupts.xml' ...
Loading plugin 'io.xml' ...
## Done
```

Conclusion

- You may download MALP from:

http://malp.hpcframework.com

You can now use MALP for MPI profiling on your own application or load one of the examples.

ParaTools

Future Work

Short-term

- We are in the process of adding new MPI-T and OMP-T analysis inside MALP
- An OTF-2 consumer library is also to be released
- A Spack installer is to be released

Medium term

- We are adding call-stack context support inside MALP
- We want to provide time-line views in the browser
- The notion of phase will be used for temporal view

ParaTools

Multi-Application Online Profiling Tool

Vi-HPS

Julien ADAM, Antoine CAPRA