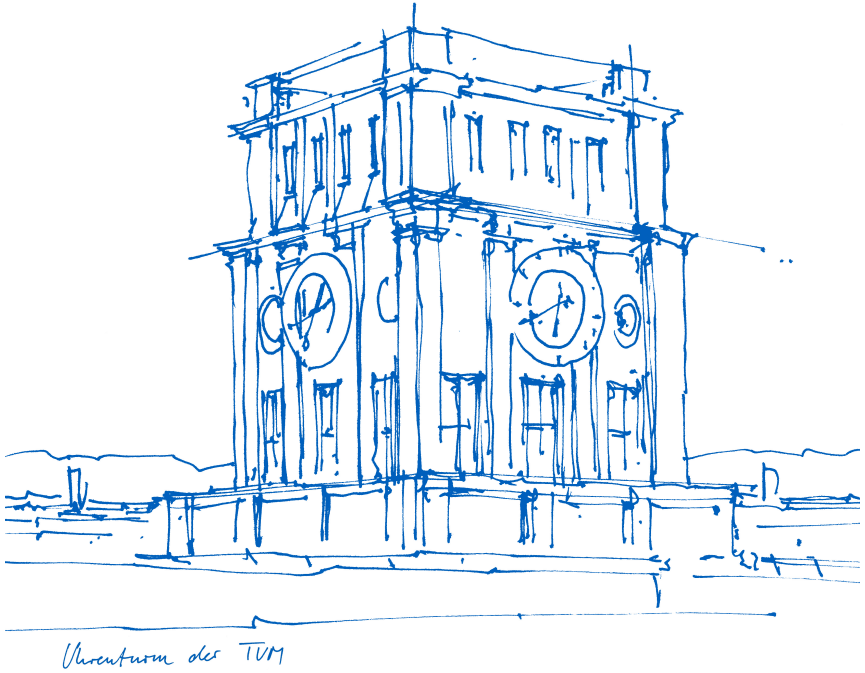
150 Jahre culture of excellence

Communication Profiling with mpiP

VI-HPS TW27, LRZ, Munich, April 23rd, 2018

Martin Schulz
Technische Universität München
Fakultät für Informatik

schulzm@in.tum.de



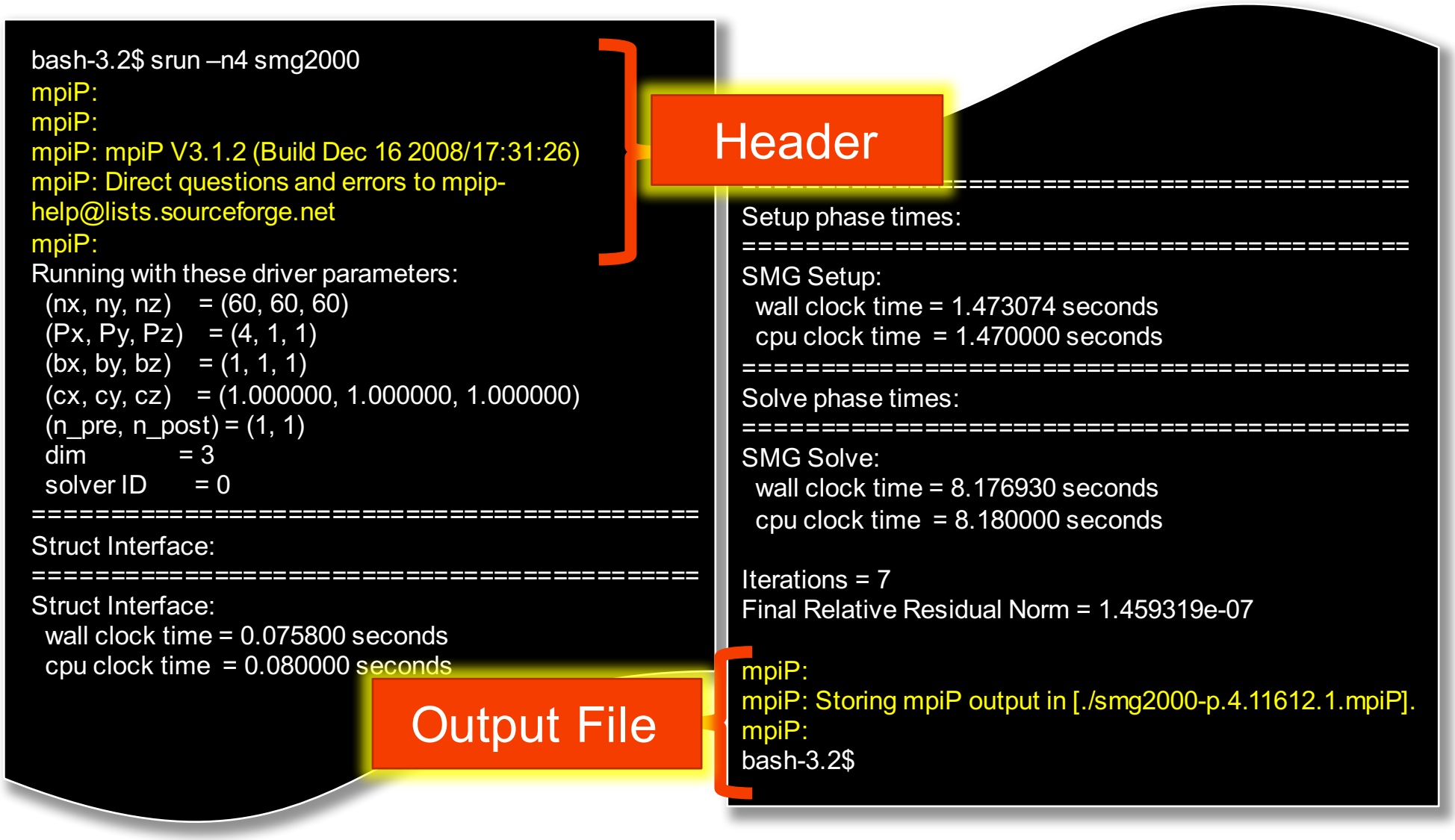
mpiP in One Slide

Open source MPI profiling library
- Categories: linker instrumentation, profiling
- Available from github (just migrated from sourceforge)
- Portable across MPI libraries and system architectures
- Available on most platforms,
  incl. IBM BG L/P/Q, Cray XT/XK/XE, Clusters

Easy-to-use and portable design
- Relies on PMPI instrumentation alone
- Single text file as output

Usage:
- Compile application with –g for better accuracy mapping to source
- Option 1: add libmpip.a/.so to the link line
- Option 2: set preload variable (e.g., LD_PRELOAD) to mpiP
- Can be hidden in job launcher

Running with mpiP 101 / Running



mpiP 101 / Output – Metadata

```
@ mpiP
@ Command : ./smg2000-p -n 60 60 60
@ Version                  : 3.1.2
@ MPIP Build date          : Dec 16 2008, 17:31:26
@ Start time               : 2009 09 19 20:38:50
@ Stop time                : 2009 09 19 20:39:00
@ Timer Used               : gettimeofday
@ MPIP env var             : [null]
@ Collector Rank           : 0
@ Collector PID            : 11612
@ Final Output Dir         : .
@ Report generation        : Collective
@ MPI Task Assignment      : 0 hera27
@ MPI Task Assignment      : 1 hera27
@ MPI Task Assignment      : 2 hera31
@ MPI Task Assignment      : 3 hera31
```

mpiP 101 / Output – Overview

```
---------------------------------------------------------------
@--- MPI Time (seconds) ----------------------------------------
---------------------------------------------------------------
Task    AppTime    MPITime     MPI%
   0       9.78       1.97    20.12
   1        9.8       1.95    19.93
   2        9.8       1.87    19.12
   3       9.77       2.15    21.99
   *       39.1       7.94    20.29
---------------------------------------------------------------
```

mpiP 101 / Output – Callsites

```
---------------------------------------------------------------------
@--- Callsites: 23 ----------------------------------------------------
---------------------------------------------------------------------
 ID Lev File/Address        Line Parent_Funct                  MPI_Call
  1   0 communication.c     1405 hypre_CommPkgUnCommit         Type_free
  2   0 timing.c             419 hypre_PrintTiming             Allreduce
  3   0 communication.c      492 hypre_InitializeCommunication Isend
  4   0 struct_innerprod.c   107 hypre_StructInnerProd         Allreduce
  5   0 timing.c             421 hypre_PrintTiming             Allreduce
  6   0 coarsen.c            542 hypre_StructCoarsen           Waitall
  7   0 coarsen.c            534 hypre_StructCoarsen           Isend
  8   0 communication.c     1552 hypre_CommTypeEntryBuildMPI   Type_free
  9   0 communication.c     1491 hypre_CommTypeBuildMPI        Type_free
 10   0 communication.c      667 hypre_FinalizeCommunication   Waitall
 11   0 smg2000.c            231 main                          Barrier
 12   0 coarsen.c            491 hypre_StructCoarsen           Waitall
 13   0 coarsen.c            551 hypre_StructCoarsen           Waitall
 14   0 coarsen.c            509 hypre_StructCoarsen           Irecv
 15   0 communication.c     1561 hypre_CommTypeEntryBuildMPI   Type_free
 16   0 struct_grid.c        366 hypre_GatherAllBoxes          Allgather
 17   0 communication.c     1487 hypre_CommTypeBuildMPI        Type_commit
 18   0 coarsen.c            497 hypre_StructCoarsen           Waitall
 19   0 coarsen.c            469 hypre_StructCoarsen           Irecv
 20   0 communication.c     1413 hypre_CommPkgUnCommit         Type_free
 21   0 coarsen.c            483 hypre_StructCoarsen           Isend
 22   0 struct_grid.c        395 hypre_GatherAllBoxes          Allgatherv
 23   0 communication.c      485 hypre_InitializeCommunication Irecv
---------------------------------------------------------------------
```

mpiP 101 / Output – per Function Timing

```
---------------------------------------------------------------
@--- Aggregate Time (top twenty, descending, milliseconds) ---
---------------------------------------------------------------
Call                 Site       Time    App%    MPI%     COV
Waitall                10    4.4e+03   11.24   55.40    0.32
Isend                   3   1.69e+03    4.31   21.24    0.34
Irecv                  23        980    2.50   12.34    0.36
Waitall                12        137    0.35    1.72    0.71
Type_commit            17        103    0.26    1.29    0.36
Type_free               9       99.4    0.25    1.25    0.36
Waitall                 6       81.7    0.21    1.03    0.70
Type_free              15       79.3    0.20    1.00    0.36
Type_free               1       67.9    0.17    0.85    0.35
Type_free              20       63.8    0.16    0.80    0.35
Isend                  21         57    0.15    0.72    0.20
Isend                   7       48.6    0.12    0.61    0.37
Type_free               8       29.3    0.07    0.37    0.37
Irecv                  19       27.8    0.07    0.35    0.32
Irecv                  14       25.8    0.07    0.32    0.34
...
```

mpiP 101 / Output – per Fct Message Size

```
---------------------------------------------------------------
@--- Aggregate Sent Message Size (top twenty, descending, bytes) ------
---------------------------------------------------------------
Call                 Site      Count      Total       Avrg  Sent%
Isend                   3     260044    2.3e+08        885  99.63
Isend                   7       9120   8.22e+05       90.1   0.36
Isend                  21       9120   3.65e+04          4   0.02
Allreduce               4         36        288          8   0.00
Allgatherv             22          4        112         28   0.00
Allreduce               2         12         96          8   0.00
Allreduce               5         12         96          8   0.00
Allgather              16          4         16          4   0.00
```

mpiP 101 / Output – per Callsite Timing

```
---------------------------------------------------------------------
@--- Callsite Time statistics (all, milliseconds): 92 ---------------
---------------------------------------------------------------------
Name                 Site Rank  Count      Max     Mean      Min   App%   MPI%
Allgather              16    0      1    0.034    0.034    0.034   0.00   0.00
Allgather              16    1      1    0.049    0.049    0.049   0.00   0.00
Allgather              16    2      1     2.92     2.92     2.92   0.03   0.16
Allgather              16    3      1        3        3        3   0.03   0.14
Allgather              16    *      4        3      1.5    0.034   0.02   0.08

Allgatherv             22    0      1     0.03     0.03     0.03   0.00   0.00
Allgatherv             22    1      1    0.036    0.036    0.036   0.00   0.00
Allgatherv             22    2      1    0.022    0.022    0.022   0.00   0.00
Allgatherv             22    3      1    0.022    0.022    0.022   0.00   0.00
Allgatherv             22    *      4    0.036   0.0275    0.022   0.00   0.00

Allreduce               2    0      3    0.382    0.239    0.011   0.01   0.04
Allreduce               2    1      3     0.31    0.148    0.046   0.00   0.02
Allreduce               2    2      3    0.411    0.178    0.062   0.01   0.03
Allreduce               2    3      3     1.33    0.622    0.062   0.02   0.09
Allreduce               2    *     12     1.33    0.297    0.011   0.01   0.04

...
```

mpiP 101 / Output – per Callsite Msg Size

```
--------------------------------------------------------------------------
@--- Callsite Message Sent statistics (all, sent bytes) ------------------
--------------------------------------------------------------------------
Name                 Site Rank    Count        Max       Mean        Min        Sum
Allgather              16    0        1          4          4          4          4
Allgather              16    1        1          4          4          4          4
Allgather              16    2        1          4          4          4          4
Allgather              16    3        1          4          4          4          4
Allgather              16    *        4          4          4          4         16

Allgatherv             22    0        1         28         28         28         28
Allgatherv             22    1        1         28         28         28         28
Allgatherv             22    2        1         28         28         28         28
Allgatherv             22    3        1         28         28         28         28
Allgatherv             22    *        4         28         28         28        112

Allreduce               2    0        3          8          8          8         24
Allreduce               2    1        3          8          8          8         24
Allreduce               2    2        3          8          8          8         24
Allreduce               2    3        3          8          8          8         24
Allreduce               2    *       12          8          8          8         96

...
```

Fine Tuning a Profile Run

Tweak parameters to gain better result and/or reduce data
- Default settings provide reasonable first overview
- Change parameters to pin point more details

Callsites are determined using stack traces
- Longer stack traces enabled better tracking
- Shorter stack traces reduce overhead
- User can set stack trace lengths for each run

Other options
- Path name handling
- Output options

In mpiP controlled by MPIP environment variable
- Set by user before profile run / command line style argument list
- Example: MPIP = “-c –o –k 4” (stack trace 4, include callsites)

mpiP Parameters

Param.	Description	Default
-c	Concise Output / No callsite data	
-f dir	Set output directory	
-k n	Set callsite stack traceback size to n	1
-l	Use less memory for data collection	
-n	Do not truncate pathnames	
-o	Disable profiling at startup	
-s n	Set hash table size	256
-t x	Print threshold	0.0
-v	Print concise & verbose output	

Controlling the Stack Trace

Callsites are determined using stack traces
- Starting from current call stack going backwards
- Useful to avoid MPI wrappers
- Helps to distinguishes library invocations

Tradeoff: stack trace depth
- Too short: can't distinguish invocations
- Too long: extra overhead / too many call sites

User can set stack trace depth
- -k <n> parameter

Limiting Scope

By default, mpiP measures entire execution
- Any event between MPI_Init and MPI_Finalize

Optional: controlling mpiP from within the application
- Disable data collection at startup (-o)
- Enable using MPI_Pcontrol(x)

Pcontrol options:
- x=0: Disable profiling
- x=1: Enable profiling
- x=2: Reset call site data
- x=3: Generate full report
- x=4: Generate concise report

Limiting Scope / Example

```
for(i=1; i < 10; i++)
{ switch(i)
  {
    case 5:
      MPI_Pcontrol(2);
      MPI_Pcontrol(1);
      break;
    case 6:
      MPI_Pcontrol(0);
      MPI_Pcontrol(4);
      break;
    default:
      break; }
  /* ... compute and communicate for one timestep ... */
}
```

Reset & Start in Iteration 5

Stop & Report in Iteration 6

Summary and Resources

mpiP = easy message passing profiling
- Enable simply by linking or preloading
- Output: simple text file
- Easily portable

Target scenario:
- Get first overview of communication behavior
  - Is time spent in MPI reasonable?
  - Which routines / call sites dominate?
  - How much data is actually exchanged?
- Prepare for deeper investigations, like MPI trace analysis

Availability:
- Source: https://github.com/LLNL/mpip
- Local location: /home/hpc/a2c06/lu23bur/local/lib/libmpiP.so

Sample Batch Script

```
#!/bin/bash
#SBATCH -J npb_btmz                 # Job name
#SBATCH -o npb_btmz.o%j             # Stdout output file(%j expands to jobId)
#SBATCH -e npb_btmz.e%j             # Stderr output file(%j expands to jobId)
#SBATCH --get-user-env              # Copy environment
#SBATCH --clusters=mpp3             # KNL cluster
#SBATCH --nodes=1                   # Total number of nodes requested
#SBATCH -n 32                       # Total number of mpi tasks requested
#SBATCH -t 00:05:00                 # Run time (hh:mm:ss) - 5 minutes
#SBATCH --constraint=cache,quad     # Request partition in cache-quadrant mode

source /etc/profile.d/modules.sh

# benchmark configuration
export OMP_NUM_THREADS=4
export NPB_MZ_BLOAD=0
PROCS=32
CLASS=C
EXE=./bt-mz_$CLASS.$PROCS

#export MPIP=
export LD_PRELOAD=/home/hpc/a2c06/local/dylib/libmpiP.so

# run the application
mpiexec $EXE
```