

GPTL Profiling Library

- Open source at github, builds with autotools
- Works with C,C++, Fortran
- Works with GNU, Intel, clang, PGI
- Thread-safe: Reports timing stats per-thread
- Can auto-profile at function level with auto-instrumentation flag
- MPI-aware: Summarizes timing stats across MPI tasks
- Can auto-profile MPI functions using PMPI hooks
 - Can auto-synchronize and time the wait between collectives => avoid misrepresenting task skew as MPI time

GPTL Profiling Library cont'd)

- Can provide memory usage info (hi-water mark) periodically as program is running
- Provides summary of timed regions which have multiple parents
- Can use a register read instead of `gettimeofday()` for gathering timing info
- Provides an estimate of overhead incurred
- Provides an indented “call tree” showing who called who

Basic Usage

```
#include <omp.h>
#include <mpi.h>
#include <unistd.h>
#include <gptl.h>
int main (int argc, char **argv) {
    int ret, n, myrank, nthreads=omp_get_max_threads();
    ret = MPI_Init (&argc, &argv);
    ret = MPI_Comm_rank (MPI_COMM_WORLD, &myrank);
    ret = GPTLsetutr (GPTLgettimeofday);
    ret = GPTLinitialize ();
    ret = GPTLstart ("main");
#pragma omp parallel for private (n)
    for (n = 0; n < nthreads; ++n)
        sub (myrank, n);
    ret = GPTLstop ("main");
    ret = GPTLpr (myrank);
    ret = GPTLpr_summary (MPI_COMM_WORLD);
    ret = MPI_Finalize ();
    return 0;
}

void sub (int myrank, int n) {
    int ret;
    ret = GPTLstart ("sub_manual");
    sleep (myrank + n);
    ret = GPTLstop ("sub_manual");
}
```

Results Display (e.g. rank 1)

Stats for thread 0:

	Called	Recurse	Wall	max	min	selfOH	parentOH
main	1	-	2.001	2.001	2.001	0.000	0.000
sub	1	-	1.000	1.000	1.000	0.000	0.000
sub_manual	1	-	1.000	1.000	1.000	0.000	0.000

Stats for thread 1:

	Called	Recurse	Wall	max	min	selfOH	parentOH
sub	1	-	2.000	2.000	2.000	0.000	0.000
sub_manual	1	-	2.000	2.000	2.000	0.000	0.000

Same stats sorted by timer for threaded regions:

Thd	Called	Recurse	Wall	max	min	selfOH	parentOH
000 sub	1	-	1.000	1.000	1.000	0.000	0.000
001 sub	1	-	2.000	2.000	2.000	0.000	0.000
SUM sub	2	-	3.000	2.000	1.000	0.000	0.000
000 sub_manual	1	-	1.000	1.000	1.000	0.000	0.000
001 sub_manual	1	-	2.000	2.000	2.000	0.000	0.000
SUM sub_manual	2	-	3.000	2.000	1.000	0.000	0.000

- Env OMP_NUM_THREADS=2 mpirun -n 2 ./a.out
- Output from rank 1 issuing GPTLpr(myrank)
- Indentation shows nested regions
- Also per-thread timings for multi-threaded regions

Results Display (summary across MPI)

Name	ncalls	nrank	mean_time	std_dev	wallmax	(rank	thread)	wallmin	(rank	thread)
main	2	2	1.501	0.707	2.001	(1 0)	1.001	(0 0)
sub	4	2	1.500	0.707	2.000	(1 1)	0.000	(0 0)
sub_manual	4	2	1.500	0.707	2.000	(1 1)	0.000	(0 0)

- Output from calling
GPTLpr_summary(MPI_COMM_WORLD)
- Mean, std_dev are over nrank, max time per thread
- Stats list max/min times across all threads and ranks that participated in the computation

Example output from auto-profiling

GSI

name	ncalls	wallmax	(rank)	wallmin	(rank)
Main	160	170.073	(0)	169.526	(70)
call_crtm	222453	37.356	(75)	2.518	(39)
crtm_forward_module_mp_crtm_forward_	64962	0.364	(75)	0.023	(70)
crtm_k_matrix_module_mp_crtm_k_matrix_	222453	34.708	(75)	2.272	(35)
sync_Bcast	6080	2.814	(1)	0.957	(0)
MPI_Bcast	6080	0.193	(158)	0.030	(0)
MPI_Barrier	160	0.051	(58)	0.000	(0)
sync_Scatterv	480	0.242	(52)	0.000	(0)
MPI_Scatterv	480	0.115	(159)	0.006	(0)
sync_Alltoallv	16640	15.010	(72)	1.190	(19)
MPI_Alltoallv	16640	2.476	(80)	2.351	(79)
sync_Alltoall	1120	0.064	(13)	0.002	(140)
MPI_Alltoall	1120	0.011	(39)	0.011	(148)
sync_Allgather	186400	0.264	(35)	0.041	(154)
MPI_Allgather	186400	0.762	(0)	0.749	(71)
sync_Allgatherv	10560	0.454	(96)	0.021	(113)
MPI_Allgatherv	10560	0.612	(2)	0.552	(98)
sync_Allreduce	3888	131.105	(88)	43.493	(122)
MPI_Allreduce	3888	0.006	(24)	0.005	(139)
sync_Reduce	752	0.312	(70)	0.003	(38)
MPI_Reduce	752	0.052	(48)	0.000	(113)
sync_Gatherv	480	0.011	(125)	0.002	(44)
MPI_Gatherv	480	0.009	(0)	0.000	(2)

Auto-profiling memory usage in GSI

```
End control_vectors_mp_assign_scalar2cv_ rss grew to 1158.207031 MB
End read_obsmod_mp_read_obs_check_ rss grew to 1196.949219 MB
End ncepnmns_io_mp_tran_gfssfc_ rss grew to 1210.652344 MB
End ncepnmns_io_mp_tran_gfssfc_ rss grew to 1224.003906 MB
End ncepnmns_io_mp_tran_gfssfc_ rss grew to 1236.593750 MB
Begin grdcrd_ rss grew to 1250.464844 MB
Begin grdcrd1_ rss grew to 1264.351562 MB
Begin grdcrd1_ rss grew to 1278.351562 MB
Begin grdcrd1_ rss grew to 1292.464844 MB
Begin grdcrd1_ rss grew to 1306.464844 MB
End combine_radobs_ rss grew to 1348.347656 MB
End disobs_ rss grew to 1420.898438 MB
Begin polcas1_ rss grew to 1450.726562 MB
Begin polcas1_ rss grew to 1470.906250 MB
Begin polcas1_ rss grew to 1494.925781 MB
Begin berror_mp_init_rftable_ rss grew to 1516.566406 MB
End berror_mp_initable_ rss grew to 1539.441406 MB
End berror_mp_initable_ rss grew to 1556.437500 MB
End berror_mp_initable_ rss grew to 1572.937500 MB
End berror_mp_initable_ rss grew to 1588.937500 MB
End berror_mp_initable_ rss grew to 1604.937500 MB
End berror_mp_initable_ rss grew to 1626.371094 MB
Begin compact_diffs_mp_compact_dlat_ rss grew to 1646.148438 MB
Begin compact_diffs_mp_compact_dlat_ rss grew to 1666.312500 MB
End setupps_ rss grew to 1759.511719 MB
End setup_ rss grew to 1822.902344 MB
End setupw_ rss grew to 1859.257812 MB
```

Multiple parent info example from GSI

```
72128 read_airs_  
106676 satthin_mp_map2tgrid_  
72128 deter_sfc_mod_mp_deter_sfc_fov_  
18422 deter_nst__  
34548 read_goesndr_  
34548 deter_sfc_mod_mp_deter_sfc_  
    4 setupps_  
    64 setupt_  
    30 setupq_  
    4733 setupw_  
    25753 setupbend_  
369034  grdcrd1_
```

- Shows parents of routine “grdcrd1”, and number of times each parent invoked it