An Overview of Tool Support for OpenMP Runtimes

Kevin Huck, Allen Malony
Performance Research Lab

UNIVERSITY OF OREGON

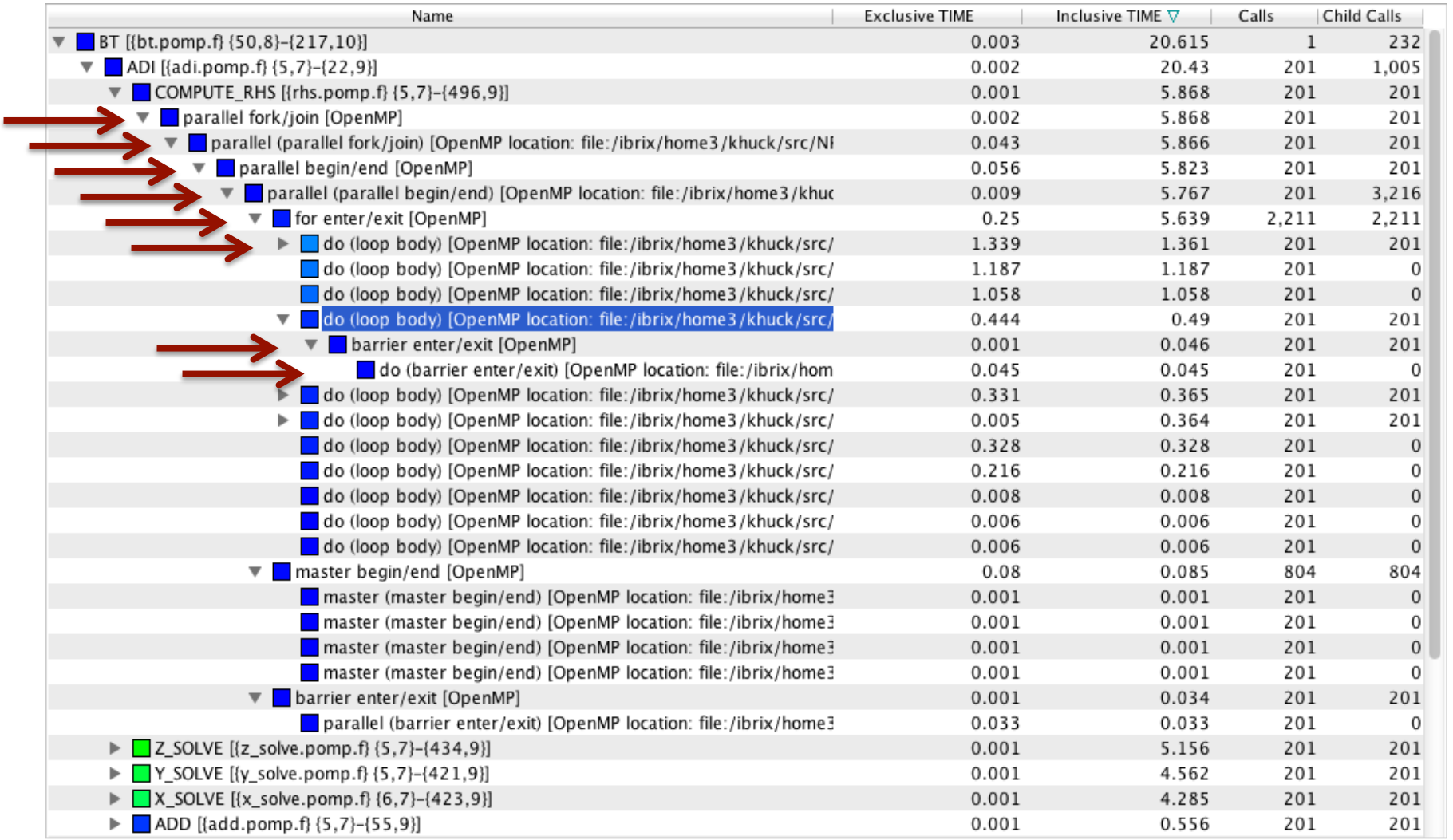
OpenMP introspection: Motivation

- Multi-process concurrency is reaching scaling limits
- Hybrid concurrency is the new normal
  - Clusters of multicore multiprocessors
  - Threads, GPUs, accelerators, MIC
- MPI+OpenMP is a popular approach
- Wanted: **portable**, **robust, efficient, always-on/available** approach to observe OpenMP concurrency, including **internals**

POMP & OPARI

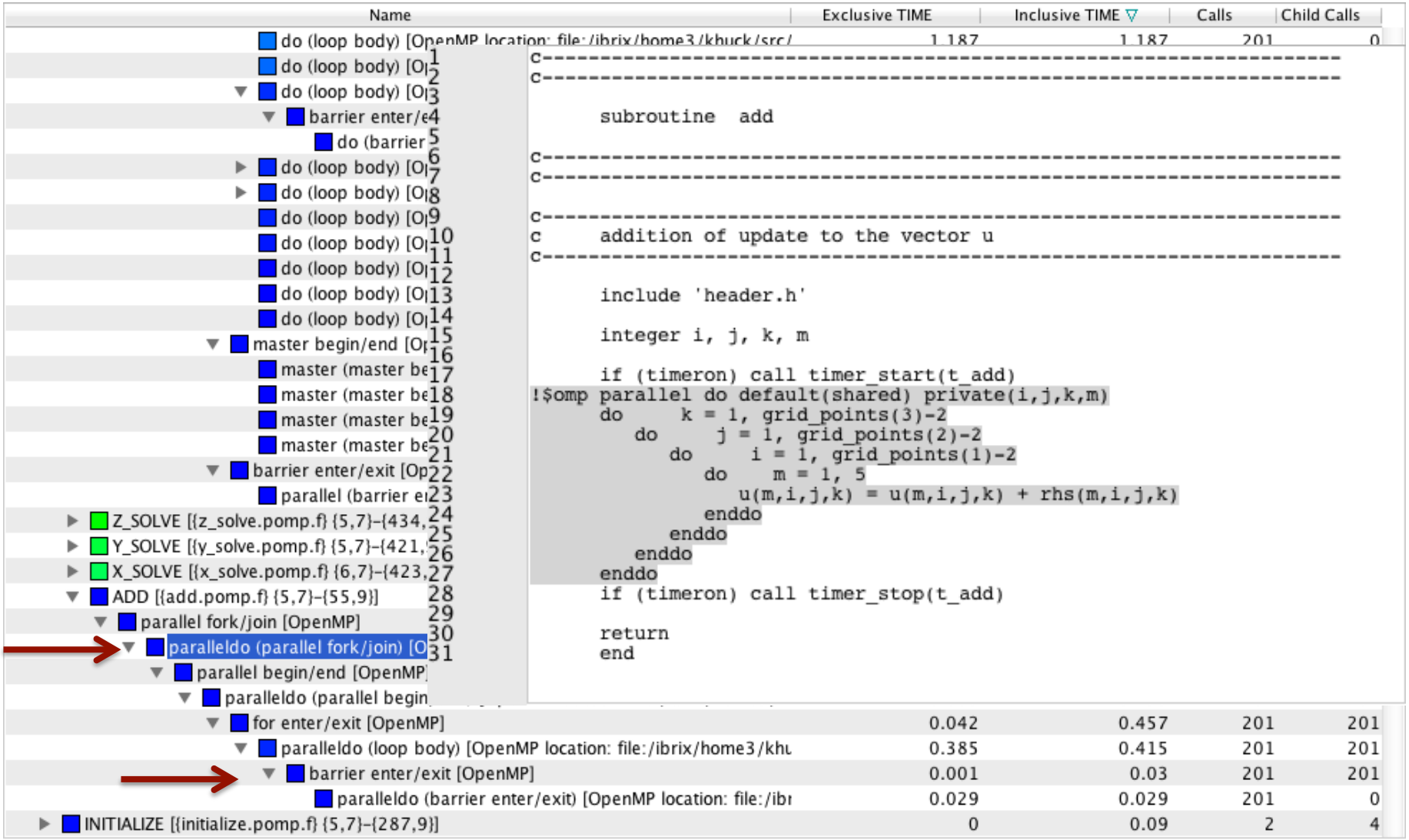
- POMP: Profiling OpenMP
- OPARI: OpenMP Pragma And Region Instrumentor
- Explicit instrumentation of OpenMP directives: Mohr et al., SC 2001
- Pros
  - Highly portable
  - Full source context
- Cons
  - Requires instrumentation, recompile
  - All barriers are explicit, extra barrier for all regions
  - Lacks explicit sampling tool support
  - Ignorance of OpenMP behavior in uninstrumented libraries
  - Ignorance of runtime library internals
- Runtimes: **all Fortran, C/C++ compilers/runtimes**
- Tool examples: Vampir, Scalasca/KOJAK, TAU, ompP

Profile Example NPB3.2 BT.B (32)



Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
BT [{bt.pomp.f} {50,8}-{217,10}]	0.003	20.615	1	232
ADI [{adi.pomp.f} {5,7}-{22,9}]	0.002	20.43	201	1,005
COMPUTE_RHS [{rhs.pomp.f} {5,7}-{496,9}]	0.001	5.868	201	201
parallel fork/join [OpenMP]	0.002	5.868	201	201
parallel (parallel fork/join) [OpenMP location: file:/ibrix/home3/khuck/src/NI	0.043	5.866	201	201
parallel begin/end [OpenMP]	0.056	5.823	201	201
parallel (parallel begin/end) [OpenMP location: file:/ibrix/home3/khuc	0.009	5.767	201	3,216
for enter/exit [OpenMP]	0.25	5.639	2,211	2,211
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	1.339	1.361	201	201
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	1.187	1.187	201	0
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	1.058	1.058	201	0
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	0.444	0.49	201	201
barrier enter/exit [OpenMP]	0.001	0.046	201	201
do (barrier enter/exit) [OpenMP location: file:/ibrix/hom	0.045	0.045	201	0
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	0.331	0.365	201	201
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	0.005	0.364	201	201
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	0.328	0.328	201	0
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	0.216	0.216	201	0
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	0.008	0.008	201	0
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	0.006	0.006	201	0
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/	0.006	0.006	201	0
master begin/end [OpenMP]	0.08	0.085	804	804
master (master begin/end) [OpenMP location: file:/ibrix/home3	0.001	0.001	201	0
master (master begin/end) [OpenMP location: file:/ibrix/home3	0.001	0.001	201	0
master (master begin/end) [OpenMP location: file:/ibrix/home3	0.001	0.001	201	0
master (master begin/end) [OpenMP location: file:/ibrix/home3	0.001	0.001	201	0
barrier enter/exit [OpenMP]	0.001	0.034	201	201
parallel (barrier enter/exit) [OpenMP location: file:/ibrix/home3	0.033	0.033	201	0
Z_SOLVE [{z_solve.pomp.f} {5,7}-{434,9}]	0.001	5.156	201	201
Y_SOLVE [{y_solve.pomp.f} {5,7}-{421,9}]	0.001	4.562	201	201
X_SOLVE [{x_solve.pomp.f} {6,7}-{423,9}]	0.001	4.285	201	201
ADD [{add.pomp.f} {5,7}-{55,9}]	0.001	0.556	201	201

Profile Example NPB3.2 BT.B (32)



OpenMP Runtime API / Collector API

- Proposed callback interface for tool support in OpenMP runtimes
- Itzkowitz et al., 2007 (2006?) – Sun / Oracle
- 1 API function for tools to make requests (registration, query state)

```
int __omp_collector_api(void *message);
```

- Callback interface to tools from runtime
  - 22 events (profiling/tracing), 11 states (sampling)
- Pros
  - Instrumentation not required
  - Event, sampling support
  - Partial view of runtime internals (implicit barriers, locks)
  - No more library blind spots
- Cons
  - Not widely implemented
  - No static executable support (requires `dlsym();` call)
  - No explicit support for locks, stack frame locations, blame shifting
  - No support for source context (integer "region id")
- Runtimes: **Open64**, **OpenUH***

*with task extensions to be presented @IWOMP2013

ORA Events & States

- Fork/Join
- Begin/End Idle
- Begin/End Implicit Barrier
- Begin/End Explicit Barrier
- Begin/End Lock wait
- Begin/End Critical wait
- Begin/End Ordered wait
- Begin/End Master
- Begin/End Single
- Begin/End Ordered
- Begin/End Atomic Wait

- Overhead
- Working
- Implicit Barrier
- Explicit Barrier
- Idle
- Serial
- Reduction
- Lock Wait
- Critical Wait
- Ordered Wait
- Atomic Wait

OpenUH new Task Events

- Begin/End Create Task Immediate/Delayed
- Begin/End Schedule Task
- Begin/End Suspend Task
- Begin/End Steal Task
- Fetched Task
- Begin Execute Task
- Begin/End Finish Task

Profile Example NPB3.2 BT.B (32)

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
BT [{bt.f} {49,8}-{215,10}]	0.003	20.306	1	232
ADI [{adi.f} {4,7}-{20,9}]	0.003	20.148	201	1,005
COMPUTE_RHS [{rhs.f} {4,7}-{426,9}]	0.006	5.522	201	201
OpenMP_PARALLEL_REGION: COMPUTE_RHS [{rhs.f} {4,7}-{426,9}]	4.987	5.516	201	1,809
OpenMP_IMPLICIT_BARRIER: COMPUTE_RHS [{rhs.f} {4,7}-{426,9}]	0.517	0.517	1,005	0
OpenMP_MASTER_REGION: COMPUTE_RHS [{rhs.f} {4,7}-{426,9}]	0.012	0.012	804	0
Z_SOLVE [{z_solve.f} {4,7}-{410,9}]	0.006	5.382	201	201
OpenMP_PARALLEL_REGION: Z_SOLVE [{z_solve.f} {4,7}-{410,9}]	5.325	5.376	201	201
OpenMP_IMPLICIT_BARRIER: Z_SOLVE [{z_solve.f} {4,7}-{410,9}]	0.051	0.051	201	0
Y_SOLVE [{y_solve.f} {4,7}-{397,9}]	0.006	4.421	201	201
OpenMP_PARALLEL_REGION: Y_SOLVE [{y_solve.f} {4,7}-{397,9}]	4.334	4.416	201	201
OpenMP_IMPLICIT_BARRIER: Y_SOLVE [{y_solve.f} {4,7}-{397,9}]	0.082	0.082	201	0
X_SOLVE [{x_solve.f} {5,7}-{399,9}]	0.006	4.23	201	201
OpenMP_PARALLEL_REGION: X_SOLVE [{x_solve.f} {5,7}-{399,9}]	4.106	4.224	201	201
OpenMP_IMPLICIT_BARRIER: X_SOLVE [{x_solve.f} {5,7}-{399,9}]	0.118	0.118	201	0
ADD [{add.f} {4,7}-{31,9}]	0.005	0.59	201	201
OpenMP_PARALLEL_REGION: ADD [{add.f} {4,7}-{31,9}]	0.385	0.585	201	201
OpenMP_IMPLICIT_BARRIER: ADD [{add.f} {4,7}-{31,9}]	0.2	0.2	201	0
INITIALIZE [{initialize.f} {4,7}-{222,9}]	0	0.088	2	4
EXACT_RHS [{exact_rhs.f} {5,7}-{351,9}]	0	0.035	1	1
VERIFY [{verify.f} {5,9}-{326,11}]	0	0.031	1	3
PRINT_RESULTS [{print_results.f} {2,7}-{136,12}]	0	0.001	1	1
TIMER_START [{timers.f} {23,7}-{38,9}]	0	0	1	1
TIMER_STOP [{timers.f} {44,7}-{61,9}]	0	0	1	1
SET_CONSTANTS [{set_constants.f} {4,7}-{200,9}]	0	0	1	0
TIMER_CLEAR [{timers.f} {4,7}-{17,9}]	0	0	22	0
TIMER_READ [{timers.f} {67,7}-{79,9}]	0	0	1	0

Task Profile Example: BOTS FFT (32)

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
int main(int, char **) C [{bots_main.c} {476,1}-{530,1}]	0.257	22.875	1	6
void fft(int, COMPLEX *, COMPLEX *) C [{tied-fft.c} {4779,1}-{4814,1}]	0	22.617	1	9
OpenMP_PARALLEL_REGION: void fft(int, COMPLEX *, COMPLEX *) C [{tied-fft.c} {477	0	22.617	2	6
OpenMP_IMPLICIT_BARRIER: void fft(int, COMPLEX *, COMPLEX *) C [{tied-fft.c} {4	0.001	22.6	4	2
OpenMP_EXECUTE_TASK	0	22.599	2	2
void fft_aux(int, COMPLEX *, COMPLEX *, int *, COMPLEX *, int) C [{tied-fft	0.001	19.588	1	21
OpenMP_SUSPEND_TASK	1.742	19.586	3	1
OpenMP_EXECUTE_TASK	0	17.844	1	1
void fft_aux(int, COMPLEX *, COMPLEX *, int *, COMPLEX *, int)	0.002	17.844	1	21
OpenMP_SUSPEND_TASK	0.756	17.841	3	10

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
int main(int, char **) C [{bots_main.c} {476,1}-{530,1}]	0.254	5.473	1	6
void fft(int, COMPLEX *, COMPLEX *) C [{tied-fft.c} {4779,1}-{4814,1}]	0	5.218	1	9
void fft_aux(int, COMPLEX *, COMPLEX *, int *, COMPLEX *, int) C [{tied-fft.c} {4621,	0.527	4.715	1	3
void fft_aux(int, COMPLEX *, COMPLEX *, int *, COMPLEX *, int) C [{tied-fft.c} {46	0.074	3.886	1	9
void fft_aux(int, COMPLEX *, COMPLEX *, int *, COMPLEX *, int) C [{tied-fft.c}	0.001	3.389	7	125
void fft_twiddle_16(int, int, COMPLEX *, COMPLEX *, COMPLEX *, int, int, int) (	0.121	0.288	1	1
void fft_unshuffle_16(int, int, COMPLEX *, COMPLEX *, int) C [{tied-fft.c} {203	0.039	0.136	1	1
void fft_twiddle_16(int, int, COMPLEX *, COMPLEX *, COMPLEX *, int, int, int) C [{t	0.158	0.158	1	0
void fft_unshuffle_16(int, int, COMPLEX *, COMPLEX *, int) C [{tied-fft.c} {2037,1}	0.144	0.144	1	0
void compute_w_coefficients(int, int, int, COMPLEX *) C [{tied-fft.c} {41,1}-{64,1}]	0.003	0.502	1	1
int factor(int) C [{tied-fft.c} {90,1}-{106,1}]			7	0
void bots_get_params(int, char **) C [{bots_main.c} {435,1}-{440,1}]			1	1
void bots_print_results() C [{bots_common.c} {115,1}-{336,1}]			1	2
void bots_set_info() C [{bots_main.c} {446,1}-{471,1}]			1	0
long bots_usecs(void) C [{bots_common.c} {78,1}-{83,1}]			2	0
void fft(int, COMPLEX *, COMPLEX *) C [{tied-fft.c} {4779,1}-{4814,1}]			1	9
void fft_aux(int, COMPLEX *, COMPLEX *, int *, COMPLEX *, int) C [{tied-fft.c} {4621,1}-{47	2.204	4.715	30,312	62,519
void fft_twiddle_16(int, int, COMPLEX *, COMPLEX *, COMPLEX *, int, int, int) C [{tied-fft.c} {	0.823	0.823	7,020	5,124
void fft_unshuffle_16(int, int, COMPLEX *, COMPLEX *, int) C [{tied-fft.c} {2037,1}-{2086,1}	0.597	0.597	7,020	5,124

Obvious takeaway: unless needed, Task Events will add overhead for lightweight tasks

GOMP ORA Support

- GOMP Implementation of OpenMP Collector API (ORA)
- GOMP_* library functions wrapped dynamically, statically
- Pros
  - No instrumentation required
  - Event, sampling support
  - Dynamic, static executable support (static requires relinking)
  - Lock support (`omp_set_lock()`, `omp_unset_lock()` wrapped)
- Cons
  - Only available for GNU
  - No introspection of GOMP internals
  - Only partial support for implicit barriers
  - GNU runtime is functional, widely available, but not optimal
  - Measures everything, all the time
  - No explicit support for blame shifting, source context
- Runtimes: **GNU**

Profile Example: forward solver

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
.TAU application	1.282	23.717	1	101
OpenMP_PARALLEL_REGION: [OPENMP] PoissonADI::SolveOmp(float, int) [{/home3/khuc	22.041	22.435	101	404
[INTERMEDIATE] OpenMP_PARALLEL_REGION: [OPENMP] PoissonADI::SolveOmp(float	0	21.9	219	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/src/H	7.9	7.9	79	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/sr	6.2	6.2	62	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/sr	0.4	0.4	4	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/sr	0.3	0.3	3	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/sr	0.3	0.3	3	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/sr	0.2	0.2	2	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/sr	0.2	0.2	2	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/sr	0.1	0.1	1	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/sr	0.1	0.1	1	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/sr	0.1	0.1	1	0
[SAMPLE] PoissonADI::SolveY(float, float, buffer*) [{/home3/khuck/src/HeadMode	7.1	7.1	71	0
[SAMPLE] PoissonADI::SolveX(float, float, buffer*) [{/home3/khuck/src/HeadMode	6.801	6.801	68	0
[SAMPLE] Poisson::Index(int, int, int) [{/home3/khuck/src/HeadModeling/trunk/sr	0.1	0.1	1	0
OpenMP_IMPLICIT_BARRIER: [OPENMP] PoissonADI::SolveOmp(float, int) [{/home3/kh	0.394	0.394	404	0
[INTERMEDIATE] .TAU application	0	1.2	12	0

Uninstrumented binary, but getting events from runtime callbacks and samples from profiling tool

Profile Example: forward solver (states)

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
.TAU application	1.214	23.402	1	101
OpenMP_PARALLEL_REGION: [OPENMP] PoissonADI::SolveOmp(float, int) [{/home3/khucl	21.239	22.188	101	404
OpenMP_IMPLICIT_BARRIER: [OPENMP] PoissonADI::SolveOmp(float, int) [{/home3/kh	0.949	0.949	404	0
OMP_IMPLICIT_BARRIER				
[INTERMEDIATE] OMP_IMPLICIT_BARRIER	0	0.5	5	0
[SAMPLE] UNRESOLVED /usr/lib64/libgomp.so.1.0.0	0.5	0.5	5	0
OMP_SERIAL				
[INTERMEDIATE] OMP_SERIAL	0	1.2	12	0
[SAMPLE] UNRESOLVED /lib64/libc-2.12.so	0.3	0.3	3	0
[SAMPLE] Poisson::init() [{/home3/khuck/src/HeadModeling/trunk/src//Poisson.cpp}	0.3	0.3	3	0
[SAMPLE] UNRESOLVED /lib64/libpthread-2.12.so	0.108	0.108	1	0
[SAMPLE] HeadModel::AdjustPaddingAir(int) [{/home3/khuck/src/HeadModeling/trur	0.1	0.1	1	0
[SAMPLE] std::_Rb_tree<int, std::pair<int const, int>, std::_Select1st<std::pair<int c	0.1	0.1	1	0
[SAMPLE] std::_Rb_tree<int, int, std::_Identity<int>, std::less<int>, std::allocator<in	0.1	0.1	1	0
[SAMPLE] Poisson::SetHeadModel(HeadModel const&) [{/home3/khuck/src/HeadMod	0.1	0.1	1	0
[SAMPLE] void std::_Rb_tree<int, int, std::_Identity<int>, std::less<int>, std::allocat	0.092	0.092	1	0
OMP_WORKING				
[INTERMEDIATE] OMP_WORKING	0	20.9	209	0
[SAMPLE] PoissonADI::SolveZ(float, float, buffer*, double*) [{/home3/khuck/src/Heac	7.499	7.499	75	0
[SAMPLE] PoissonADI::SolveY(float, float, buffer*) [{/home3/khuck/src/HeadModeling	7.199	7.199	72	0
[SAMPLE] PoissonADI::SolveX(float, float, buffer*) [{/home3/khuck/src/HeadModeling	5.902	5.902	59	0
[SAMPLE] Poisson::Index(int, int, int) [{/home3/khuck/src/HeadModeling/trunk/src//	0.3	0.3	3	0

Enabling states to separate samples

OpenMP Runtime Instrumentation

- Runtime libraries with known APIs can be wrapped or instrumented (DynInstAPI)
- Example: Extrae measurement for Paraver (BSC)
- Pros:
  - Same as collector API
- Cons:
  - Lacks implicit barrier support
  - Less portable than OPARI
  - Specific to one tool (the implementation, not the approach)
  - No explicit support for blame shifting, source context
- Runtimes: **Intel**, **IBM**, **GNU**

OMPT: OpenMP Tools Interface

- John will have covered this right before me
- Pros:
  - No instrumentation required
  - Event, sampling support
  - Dynamic, static executable support
  - Internals knowledge, surgical measurement, always on
  - Blame-shifting, debugger support
- Cons
  - Still in development
  - Not an accepted standard (yet)
  - Potential for high overhead with some events
- Available: **IBM** (limited/experimental on BGQ), **GNU** (prototyped by Rice), **Intel** (in development), **OpenUH** (planned)

Profile Example: NPB 3.2 BP (32)

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
BT [{bt.f} {49,8}-{215,10}]	0.005	18.247	1	232
ADI [{adi.f} {4,7}-{20,9}]	0.003	18.107	201	1,005
COMPUTE_RHS [{rhs.f} {4,7}-{426,9}]	0.004	5.37	201	201
OpenMP_PARALLEL_REGION: COMPUTE_RHS [{rhs.f} {4,7}-{426,9}]	0.045	5.367	201	4,020
OpenMP_LOOP: COMPUTE_RHS [{rhs.f} {4,7}-{426,9}]	4.344	4.344	2,211	0
OpenMP_BARRIER: COMPUTE_RHS [{rhs.f} {4,7}-{426,9}]	0.021	0.957	1,005	2,640
IDLE	0.936	0.936	2,640	0
OpenMP_MASTER_REGION: COMPUTE_RHS [{rhs.f} {4,7}-{426,9}]	0.021	0.021	804	0
Z_SOLVE [{z_solve.f} {4,7}-{410,9}]	0.005	4.57	201	201
OpenMP_PARALLEL_REGION: Z_SOLVE [{z_solve.f} {4,7}-{410,9}]	0.003	4.565	201	402
OpenMP_LOOP: Z_SOLVE [{z_solve.f} {4,7}-{410,9}]	4.194	4.194	201	0
OpenMP_BARRIER: Z_SOLVE [{z_solve.f} {4,7}-{410,9}]	0.002	0.367	201	202
IDLE	0.366	0.366	202	0
Y_SOLVE [{y_solve.f} {4,7}-{397,9}]	0.004	3.981	201	201
OpenMP_PARALLEL_REGION: Y_SOLVE [{y_solve.f} {4,7}-{397,9}]	0.003	3.977	201	402
OpenMP_LOOP: Y_SOLVE [{y_solve.f} {4,7}-{397,9}]	3.856	3.856	201	0
OpenMP_BARRIER: Y_SOLVE [{y_solve.f} {4,7}-{397,9}]	0.002	0.118	201	203
IDLE	0.116	0.116	203	0
X_SOLVE [{x_solve.f} {5,7}-{399,9}]	0.004	3.709	201	201
OpenMP_PARALLEL_REGION: X_SOLVE [{x_solve.f} {5,7}-{399,9}]	0.003	3.704	201	402
OpenMP_LOOP: X_SOLVE [{x_solve.f} {5,7}-{399,9}]	3.683	3.683	201	0
OpenMP_BARRIER: X_SOLVE [{x_solve.f} {5,7}-{399,9}]	0.002	0.018	201	161
IDLE	0.016	0.016	161	0
ADD [{add.f} {4,7}-{31,9}]	0.004	0.475	201	201
OpenMP_PARALLEL_REGION: ADD [{add.f} {4,7}-{31,9}]	0.003	0.47	201	402
OpenMP_LOOP: ADD [{add.f} {4,7}-{31,9}]	0.345	0.345	201	0
OpenMP_BARRIER: ADD [{add.f} {4,7}-{31,9}]	0.001	0.123	201	228
IDLE	0.121	0.121	228	0
INITIALIZE [{initialize.f} {4,7}-{222,9}]	0.014	0.07	2	4
EXACT_RHS [{exact_rhs.f} {5,7}-{351,9}]	0	0.032	1	1

Event/State Coverage

Feature	OPARI	ORA	OpenUH-ORA	GOMP-ORA	OMPT
Parallel Region	Enter/exit, begin/end	✔	✔	✔	✔
Thread create/exit	Pthread	✖	✖	✖	✔
Work Sharing	Enter/exit, begin/end	✔	✔	✔	✔
Atomics	Enter/exit	Wait State	Wait State	Wait State	✔
Barriers	Explicit only	✔	✔	Explicit, Some Implicit	✔
Critical	Enter/exit, begin/end	Wait State	Wait State	Wait State	✔
Master	Begin/end	✔	✔	✔	✔
Ordered	Enter/exit, begin/end	✔, Wait	✔, Wait State	✔, Wait State	✔
Sections	Enter/exit, begin/end	✖	✖	✖	✔
Single	Begin/end	✔	✔	✔	✔
Locks	Lock wrapper	Wait State	Wait State	Wait State	✔
Task Creation	Begin/end	✔	✔	✔	✔
Task Schedule	✖	✖	✔	n/a	switch
Task Wait	Begin/end	✔	✔	✔	✔
Task Execution	Begin/end	✔	✔	✔	✔
Task Completion	✖	✖	✔	✖	✔
Task Yield	✖	✖	✔	n/a	switch

Overhead comparisons (OMPT later)

Benchmark	Baseline	Collector	Collector+TAU	Opari+TAU
BT	19.4 ± 0.21	19.7 ± 0.19	19.9 ± 0.21	19.8 ± 0.19
CG	8.9 ± 0.30	10.2 ± 0.18	10.3 ± 0.08	10.0 ± 0.17
EP	1.6 ± 0.01	1.7 ± 0.06	1.7 ± 0.09	1.7 ± 0.06
FT	4.2 ± 0.34	4.0 ± 0.04	4.2 ± 0.04	4.2 ± 0.04
LU	13.9 ± 0.26	15.0 ± 0.65	14.3 ± 0.78	40.3 ± 0.93
LU-HP	23.6 ± 0.41	137.1 ± 2.17	142.2 ± 1.53	127.4 ± 2.41
MG	1.5 ± 0.01	2.2 ± 0.08	2.1 ± 0.05	2.1 ± 0.04
SP	18.2 ± 1.32	18.6 ± 0.19	19.7 ± 0.26	19.8 ± 0.89

Table 2. OpenUH NPB overhead measurements for 32 threads on ACISS.

Benchmark	Baseline	Collector	Collector+TAU	Opari+TAU
BT	19.0 ± 0.75	20.5 ± 0.88	20.7 ± 0.03	18.8 ± 0.05
CG	8.5 ± 0.35	9.3 ± 0.15	9.1 ± 0.03	10.2 ± 0.05
EP	2.8 ± 0.12	2.7 ± 0.04	2.9 ± 0.03	3.0 ± 0.05
FT	3.6 ± 0.02	3.8 ± 0.23	3.8 ± 0.01	3.8 ± 0.03
LU	12.6 ± 0.52	13.0 ± 0.53	13.4 ± 0.04	98.7 ± 0.07
LU-HP	23.8 ± 1.53	63.5 ± 0.83	58.2 ± 0.03	164.2 ± 0.05
MG	1.5 ± 0.02	1.9 ± 0.06	2.0 ± 0.02	2.2 ± 0.04
SP	17.8 ± 0.33	17.6 ± 0.47	16.5 ± 0.02	17.4 ± 0.05

Table 3. GCC NPB overhead measurements for 32 threads on ACISS.

OPARI Additional Events

Name	Exclusive TIME	Inclusive TIME	Inclusive TIME / Call ▽	Calls	Child Calls
for enter/exit [OpenMP]	5.9177	54.9039	0.0002	300,736	300,736
barrier enter/exit [OpenMP]	3.3869	51.7246	0.0002	300,479	300,479
parallel begin/end [OpenMP]	11.4200	108.7703	0.0004	299,963	299,963
parallel fork/join [OpenMP]	0.2048	131.8197	0.0004	299,963	299,963
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2.1/NPB3.2-O	3.8780	3.8780	0.0001	74,798	0
do (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2.1/NPB3.2-O	3.9082	3.9082	0.0001	74,798	0
parallel (barrier enter/exit) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2.	11.2482	11.2482	0.0002	74,798	0
parallel (barrier enter/exit) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2.	11.4249	11.4249	0.0002	74,798	0
paralleldo (barrier enter/exit) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.	11.6627	11.6627	0.0002	74,798	0
paralleldo (barrier enter/exit) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.	11.8941	11.8941	0.0002	74,798	0
paralleldo (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2.1/NP	5.4058	17.9379	0.0002	74,798	74,798
paralleldo (loop body) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2.1/NP	5.4488	18.2168	0.0002	74,798	74,798
parallel (parallel begin/end) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2	4.7560	22.1298	0.0003	74,798	149,596
parallel (parallel begin/end) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2	4.6536	22.3450	0.0003	74,798	149,596
paralleldo (parallel begin/end) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3	4.0559	23.4379	0.0003	74,798	74,798
paralleldo (parallel begin/end) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3	4.0142	23.6783	0.0003	74,798	74,798
parallel (parallel fork/join) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2.1	4.3245	30.6811	0.0004	74,798	74,798
parallel (parallel fork/join) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2.1	4.4580	30.7417	0.0004	74,798	74,798
JACLD [{jacld.pomp.f} {6,7}-{387,9}]	0.0480	30.7774	0.0004	74,798	74,798
JACU [{jacu.pomp.f} {6,7}-{386,9}]	0.0474	30.8371	0.0004	74,798	74,798
paralleldo (parallel fork/join) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.1	4.2251	31.9492	0.0004	74,798	74,798
BUTS [{buts.pomp.f} {5,7}-{271,9}]	0.0463	32.0417	0.0004	74,798	74,798
paralleldo (parallel fork/join) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.1	4.3732	32.3329	0.0004	74,798	74,798
BLTS [{blts.pomp.f} {5,7}-{272,9}]	0.0469	32.4282	0.0004	74,798	74,798
master begin/end [OpenMP]	0.0253	0.0389	0.0000	1,013	1,013
master (master begin/end) [OpenMP location: file:/ibrix/home3/khuck/src/NPB3.2	0.0002	0.0002	0.0000	253	0

~2,398,166 extra timers!

Collector API Extra Events

Name	Exclusive TIME	Inclusive TIME	Inclusive TIME / Call	Calls ▽	Child Calls
OpenMP_FINISH_TASK [THROTTLED]	3.5530	3.5530	0.0000	298,959	0
JACLD [{jacld.f} {5,7}-{366,9}]	0.7801	34.1022	0.0005	74,798	74,798
JACU [{jacu.f} {5,7}-{365,9}]	0.7720	34.0634	0.0005	74,798	74,798
BUTS [{buts.f} {4,7}-{248,9}]	0.7775	33.9265	0.0005	74,798	74,296
BLTS [{blts.f} {4,7}-{249,9}]	0.7483	33.5503	0.0004	74,798	74,296
OpenMP_PARALLEL_REGION: JACU [{jacu.f} {5,7}-{365,9}]	11.3871	24.8788	0.0003	74,798	149,596
OpenMP_PARALLEL_REGION: JACLD [{jacld.f} {5,7}-{366,9}]	11.1563	24.7111	0.0003	74,798	149,596
OpenMP_IMPLICIT_BARRIER: JACLD [{jacld.f} {5,7}-{366,9}]	12.6658	12.6658	0.0002	74,798	0
OpenMP_IMPLICIT_BARRIER: JACU [{jacu.f} {5,7}-{365,9}]	12.6106	12.6106	0.0002	74,798	0
OpenMP_PARALLEL_REGION: BUTS [{buts.f} {4,7}-{248,9}]	10.9541	24.4962	0.0003	74,296	148,592
OpenMP_PARALLEL_REGION: BLTS [{blts.f} {4,7}-{249,9}]	10.8518	24.3023	0.0003	74,296	148,592
OpenMP_IMPLICIT_BARRIER: BUTS [{buts.f} {4,7}-{248,9}]	12.6434	12.6434	0.0002	74,296	0
OpenMP_IMPLICIT_BARRIER: BLTS [{blts.f} {4,7}-{249,9}]	12.5688	12.5688	0.0002	74,296	0
OpenMP_MASTER_REGION: RHS [{rhs.f} {5,7}-{448,9}]	0.0077	0.0077	0.0000	1,012	0
OpenMP_IMPLICIT_BARRIER: RHS [{rhs.f} {5,7}-{448,9}]	1.7619	1.7619	0.0023	759	0
OpenMP_PARALLEL_REGION: SSOR [{ssor.f} {4,7}-{273,9}]	0.5451	0.9610	0.0019	504	1,008
OpenMP_IMPLICIT_BARRIER: SSOR [{ssor.f} {4,7}-{273,9}]	0.4149	0.4149	0.0008	504	0
RHS [{rhs.f} {5,7}-{448,9}]	0.0060	4.9819	0.0197	253	253
OpenMP_PARALLEL_REGION: RHS [{rhs.f} {5,7}-{448,9}]	3.1801	4.9436	0.0195	253	1,043.625
TIMER_CLEAR [{timers.f} {4,7}-{17,9}]	0.0000	0.0000	0.0000	44	0

Obvious takeaway: unless needed, Events will add overhead for lightweight loops

~897,103 extra timers!

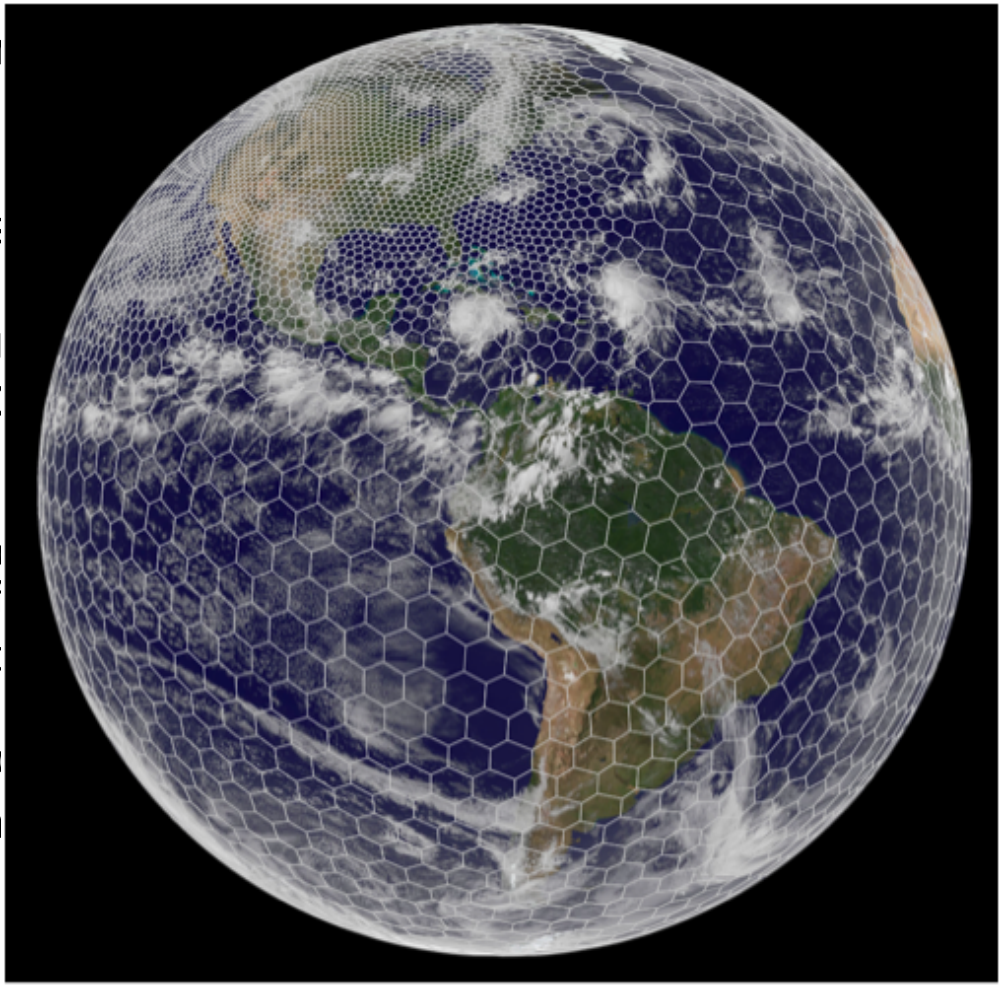
OMPT Overheads

- LU-HP
- Intel: 20.430s
- Intel+TAU: 22.911s
- Intel+TAU+OMPT (currently supported events): 137.016s
- Intel+TAU+OMPT (minimal support)+Samples: 25.569s

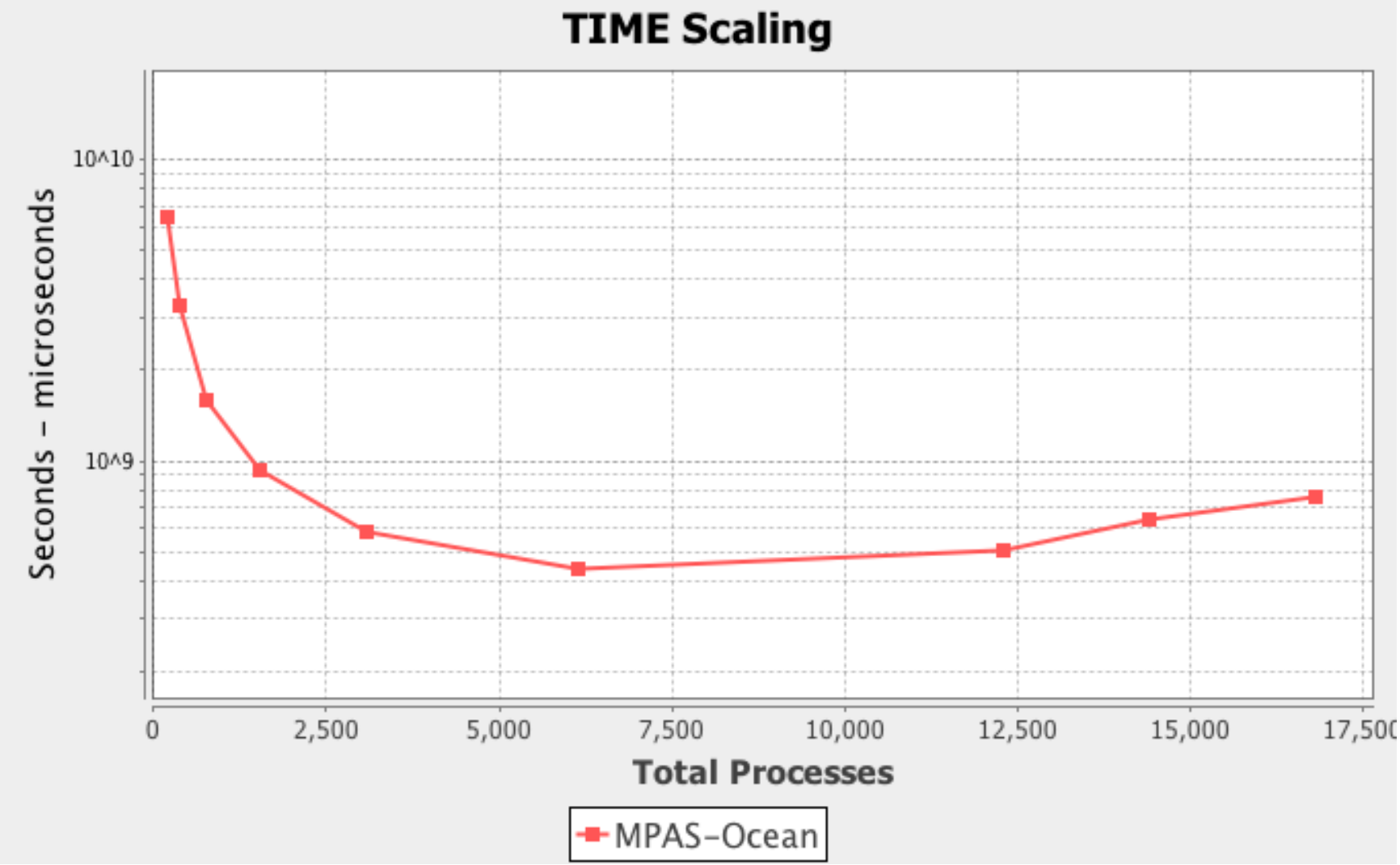
Obvious takeaway: unless needed, Task Events will add overhead for lightweight tasks

DOE SciDAC: MPAS-Ocean

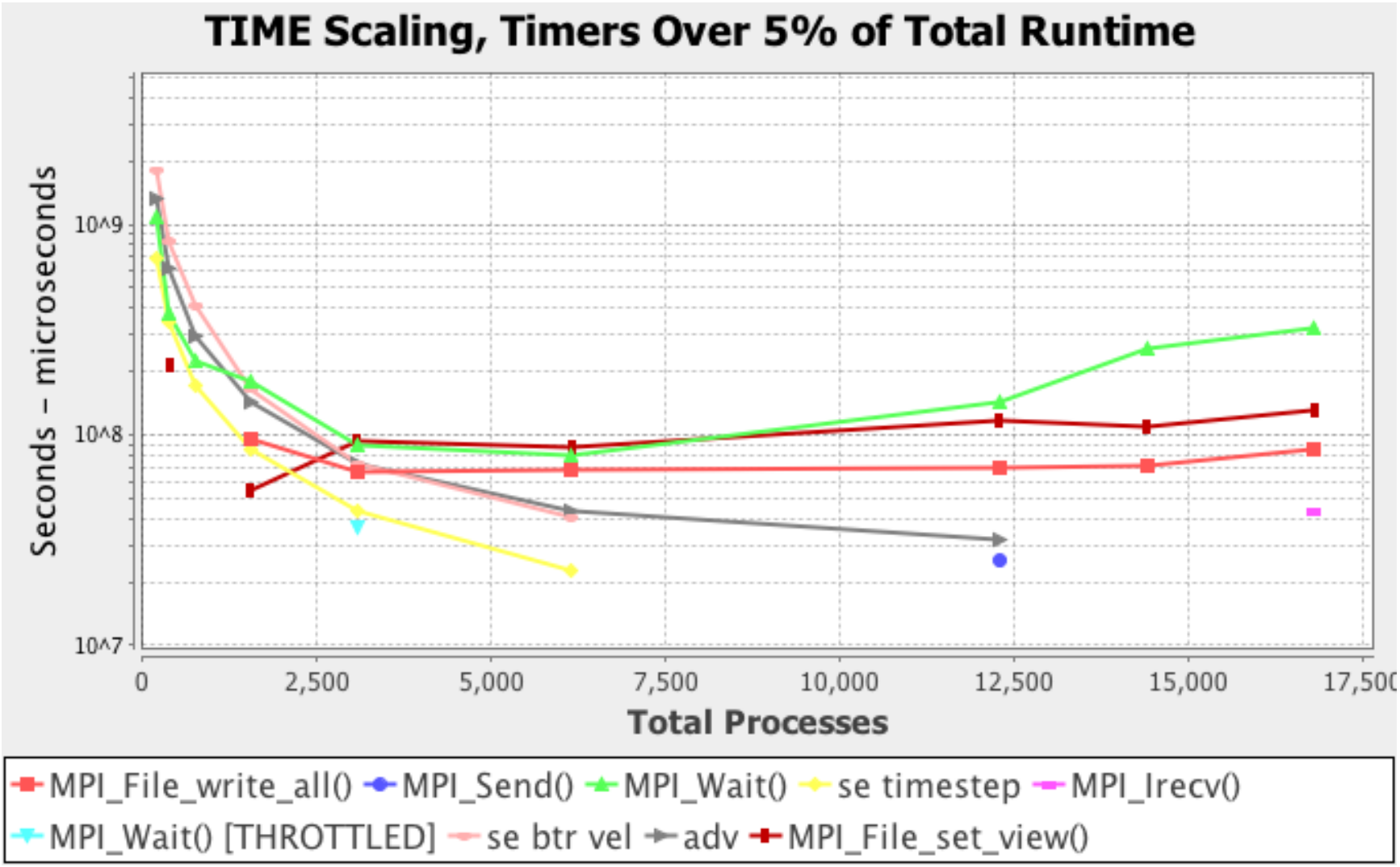
- MPAS-Ocean (SciDAC-3 MULTISCALE) has scaling limits using MPI only
- Currently adding OpenMP pragm
- TAU (SciDAC-3 SUPER) was used suggest improvements
- Tests performed by Doug Jacobs Shende (U. Oregon)
- Observations, optimizations fou
  1. MPI block decomposition + C instructions in computationa block decomposition alone
  2. Weighted block decompositi work across processes (~5% f
  3. Guided schedule balances wo
  4. Overlapping communication delays when exchanging halo
- Evaluation of OpenMP implem
- http://mpas-dev.github.io



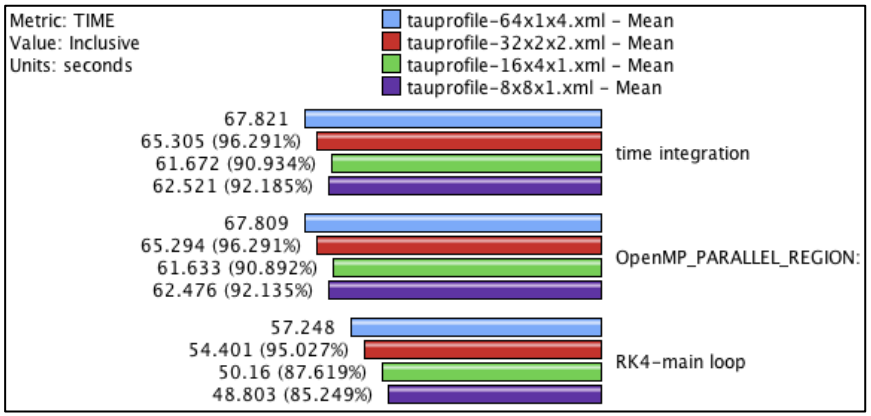
MPI Scaling to 16,800 processes



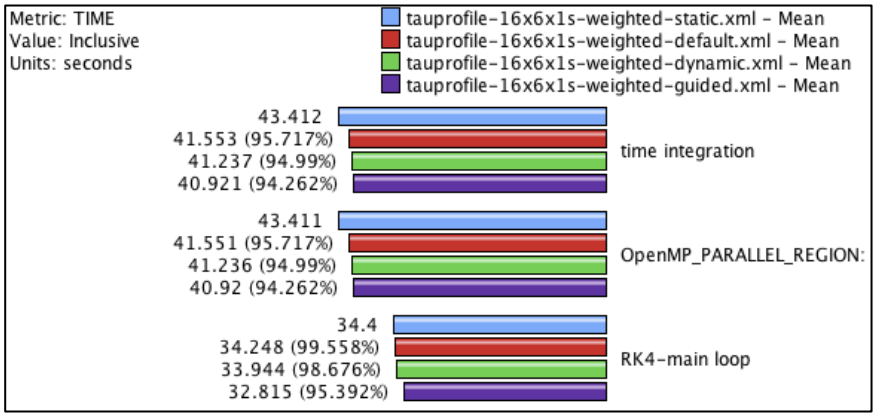
MPI Scaling to 16,800 processes



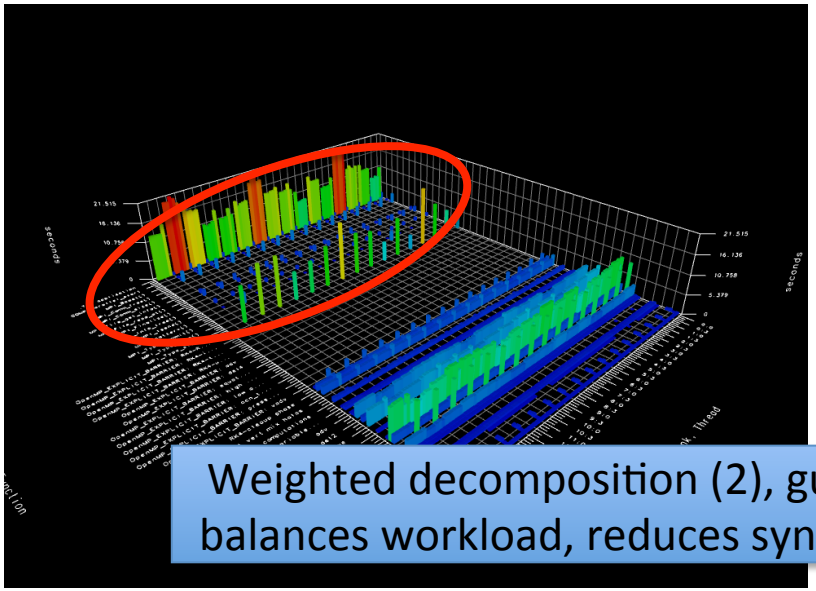
OpenMP Performance Improvements

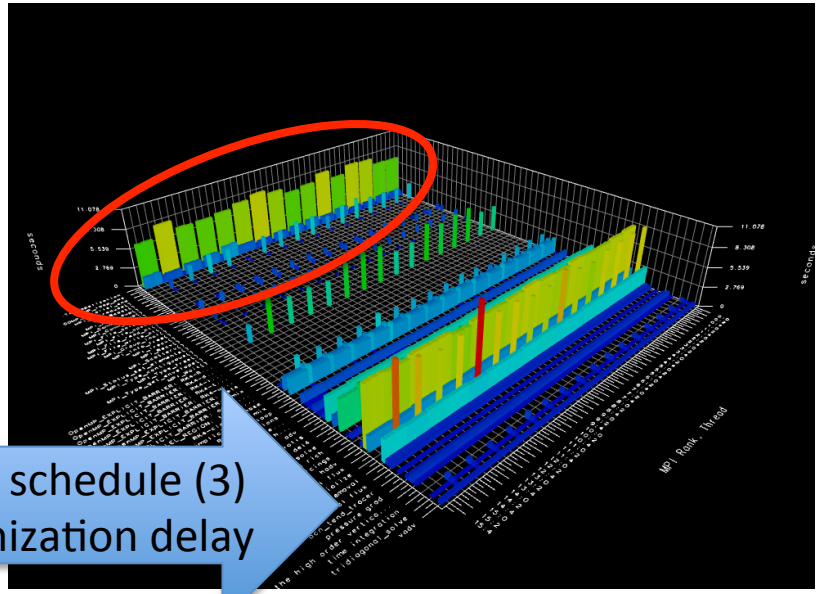


1. Increasing thread concurrency with constant resources (64 cores used in all cases).



3. Adjusting thread schedule balances workload, reduces computation time (but not all loops)





Weighted decomposition (2), guided schedule (3) balances workload, reduces synchronization delay

Summary

- Several OpenMP options to choose from
  - OPARI is the most portable alternative
- Obviously, best implementation support comes from within the runtime
- OMPT is a promising opportunity for OpenMP runtime tool support
- Still need to be careful about processing too many lightweight events (especially tasks) – even with OMPT
- Acknowledgements: The research was supported by grants from the National Science Foundation