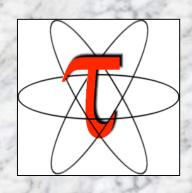
## **TAU Parallel Performance System**

#### **PDC Summer School in HPC**



Part 3: TAU Applications and Developments



#### **Tutorial Outline – Part 3**

#### **TAU Applications and Developments**

- Selected Applications
  - O PETSc, EVH1, SAMRAI, Stommel
  - O mpiJava, Blitz++, SMARTS
  - C-SAFE/Uintah
  - O HYCOM, AVUS
- □ Current developments
  - PerfDMF
  - Online performance analysis
  - O ParaVis
- □ Integrated performance evaluation environment

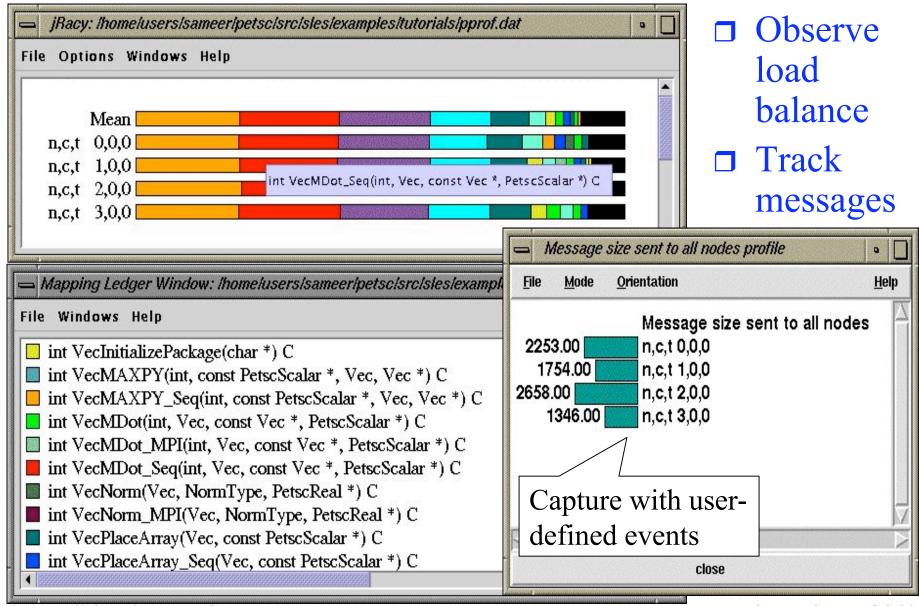
#### Case Study: PETSc v2.1.3 (ANL)

- □ Portable, Extensible Toolkit for Scientific Computation
- □ Scalable (parallel) PDE framework
  - Suite of data structures and routines (374,458 code lines)
  - Solution of scientific applications modeled by PDEs
- □ Parallel implementation
  - MPI used for inter-process communication
- **TAU** instrumentation
  - PDT for C/C++ source instrumentation (100%, no manual)
  - MPI wrapper interposition library instrumentation
- □ Example
  - Linear system of equations (Ax=b) (SLES) (ex2 test case)
  - Non-linear system of equations (SNES) (ex19 test case)

#### PETSc ex2 (Profile - wallclock time)

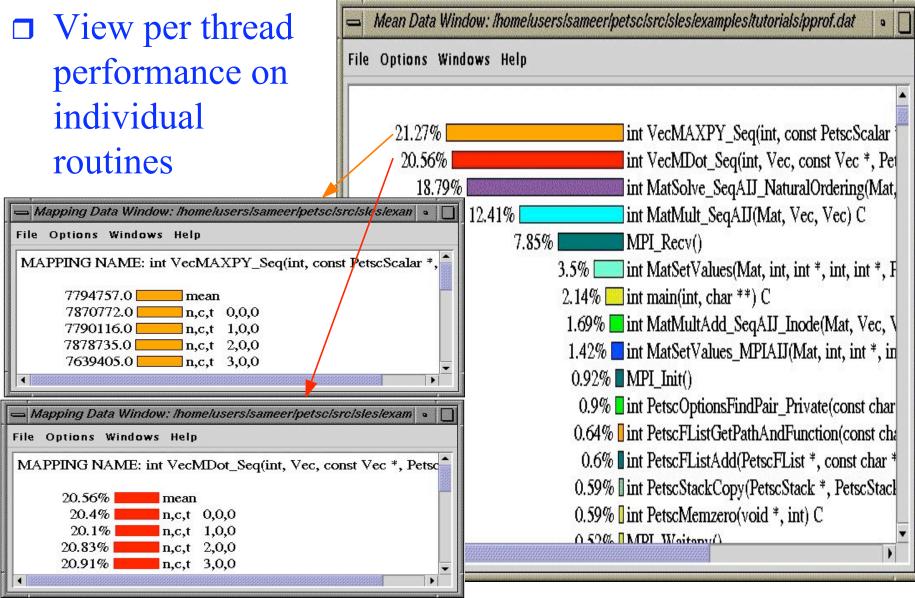
A Mean Total Stat Window: Inomelusers/sameer/petsc/src/sles/examples/tutorials/pprof.dat								
ile Opti <mark>ons</mark>	Windows							
		Son	ted with	n respe	ct to ex	xclusive time		
%time	msec	total msec	#call	#cubre	usec/call			
76 C 1 IIIC	insec.	cocar insect	#Call	#SUDIS	usec/ carr			
	7 704	7 704	407		10150			
21.3	7,794	7,794	407	0		2 int VecMAXPY_Seq(int, const PetscScalar *, Vec,		
20.6	7,534	7,534	393	0		2 int VecMDot_Seq(int, Vec, const Vec *, PetscScal		
18.8	6,886	6,908	407	1628		3 int MatSolve_SeqAIJ_NaturalOrdering(Mat, Vec, Ve		
12.5	4,548	4,599	407	1628	11302	2 int MatMult_SeqAIJ(Mat, Vec, Vec) C		
7.9	2,877	2,877	1353.75	0	2126	6 MPI_Recv()		
4.9	1,282	1,801	49800	49800	36	6 int MatSetValues(Mat, int, int *, int, int *, Pe		
100.0	785	36,651	1	49832	36651451	1 int main(int, char **) C		
1.7	618	627	407	1628	1543	3 int MatMultAdd_SegAIJ_Inode(Mat, Vec, Vec, Vec)		
1.4	519	519	49800	0	10	0 int MatSetValues_MPIAIJ(Mat, int, int *, int, in		
0.9	337	337	1	35		0 MPI_Init()		
1.1	328	394	3142	15205		6 int PetscOptionsFindPair_Private(const char *, c		
0.7	233	240	182	649		0 int PetscFListGetPathAndFunction(const char *, c		
	000000000		04/01/32/202					
1.3	219	463	153	1110		2 int PetscFListAdd(PetscFList *, const char *, co		
0.6	215	215	1526.25	0	141	1 int PetscStackCopy(PetscStack *, PetscStack *) C		

## PETSc ex2(Profile - overall and message counts)



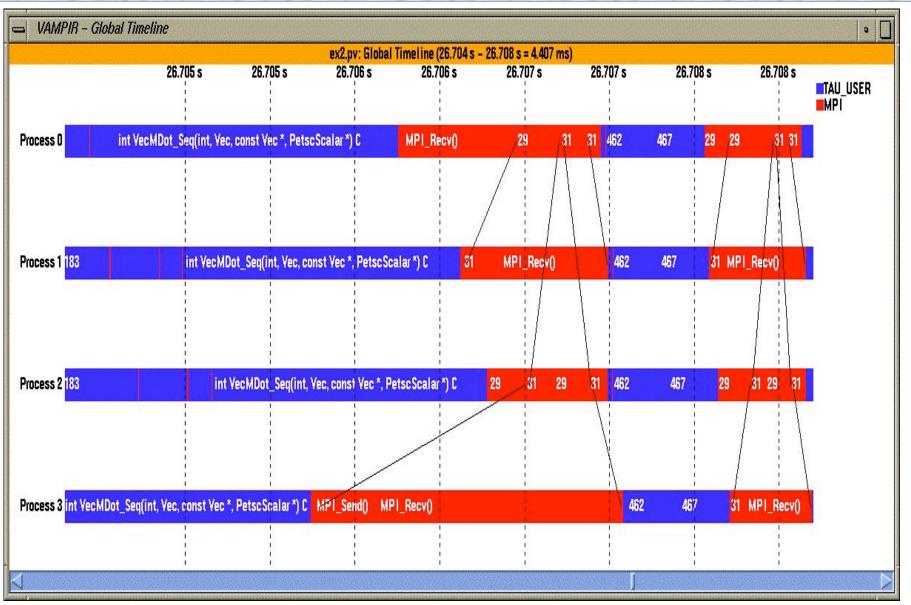
TAU Parallel Performance System

#### PETSc ex2 (Profile - percentages and time)



TAU Parallel Performance System

## PETSc ex2 (Trace)



TAU Parallel Performance System

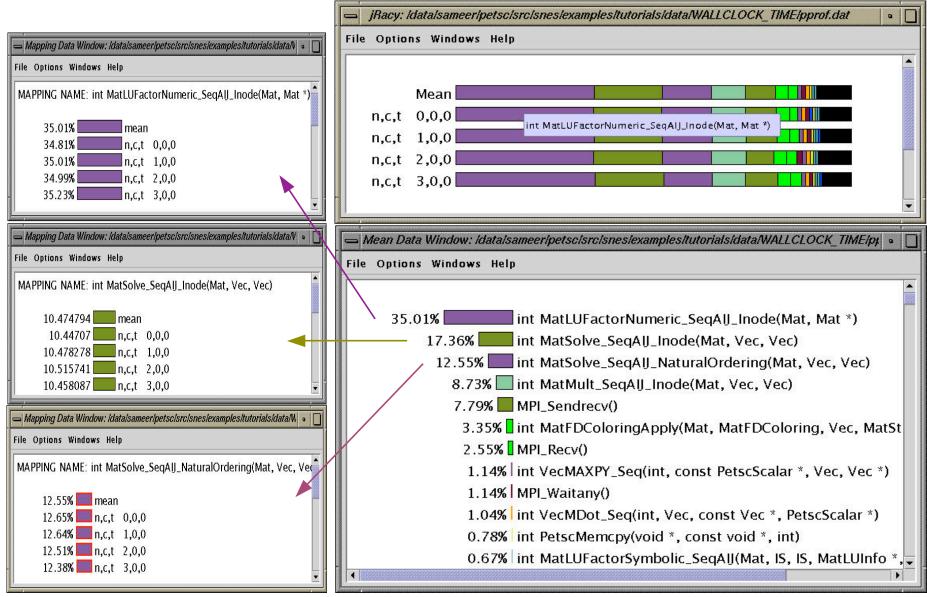
#### PETSc ex19

- □ Non-linear solver (SNES)
  - O 2-D driven cavity code
  - Uses velocity-vorticity formulation
  - Finite difference discretization on a structured grid
- Problem size and measurements
  - O 56x56 mesh size on quad Pentium III (550 Mhz, Linux)
  - Executes for approximately one minute
  - MPI wrapper interposition library
  - PDT (*tau\_instrumentor*)
  - Selective instrumentation (*tau\_reduce*)
    - > three routines identified with high instrumentation overhead

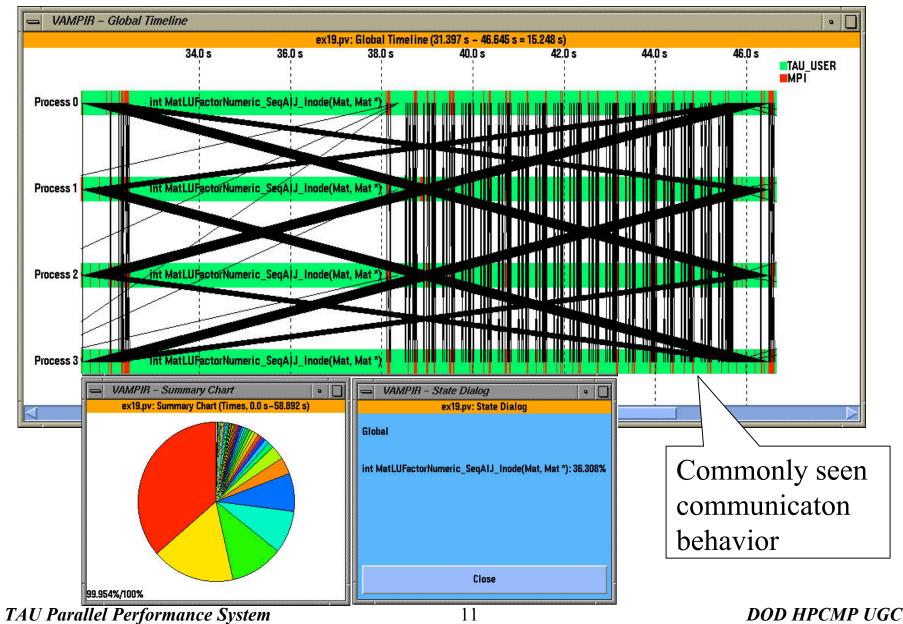
#### PETSc ex19 (Profile - wallclock time)

	' <i>Stat Window: Idata</i> Windows Help	alsameerlpetsclsrclsn	eslexamples/tutorials/c	dataWALLCLOCK_TIM	Sort	ed by	inclusive time
%time	msec	total msec	#call	#subrs usec	/call name		
100.0	19	1:00.337	1			nain(int, ch	
93.7	0.117	56,558	2			MMGSolve(DM	
93.7	0 0477	56 556	C alsameerineteolerala	reslexamples/tutorials		CK TIMEIDDO	C(DMMC * int)
93.7 93.7		Windows Help	азатестретостотото	теменатревниконак	SIGALAI WALLULU	CK_TIMEIPPIO	Sorted by exclusive time
88.0							<u> </u>
50.6							
50.6							
50.6	%time	msec	total msec	#call	#subrs	usec/call	name
46.4							
46.4	35.0	21,123	21,136	6	90	3522740	int MatLUFactorNumeric_SeqAIJ_Inode(Mat,
46.4	17.4	10,474	10,479	68	544		<pre>int MatSolve_SeqAIJ_Inode(Mat, Vec, Vec)</pre>
45.7	12.6	7,570	7,574	136	544		int MatSolve_SeqAIJ_NaturalOrdering(Mat,
38.3	8.7	5,267	5,272	208	832	25351	<pre>int MatMult_SeqAIJ_Inode(Mat, Vec, Vec)</pre>
38.2	7.8	4,702	4,704	1212	1212	3881	MPI_Sendrecv()
37.3	5.2	2,020	3,151	8	1280	393963	int MatFDColoringApply(Mat, MatFDColorin
37.3	2.5	1,536	1,536	994.75	0	1545	MPI_Recv()
	1.1	688	688	242	0	2847	int VecMAXPY_Seq(int, const PetscScalar
	1.1	686	686	978.5	0	701	MPI_Waitany()
	1.0	629	629	170	0	3703	<pre>int VecMDot_Seq(int, Vec, const Vec *, P</pre>
	0.8	470	470	1075	0	437	<pre>int PetscMemcpy(void *, const void *, in</pre>
	1.3	404	797	2	52	398722	<pre>int MatLUFactorSymbolic_SeqAIJ(Mat, IS,</pre>
	0.7	400	400	3934	0	102	<pre>int PetscMemzero(void *, int)</pre>
	0.6	356	359	208	832	1726	<pre>int MatMultAdd_SeqAIJ_Inode(Mat, Vec, Ve</pre>
	0.5	291	291	1	35		MPI_Init()
	0.7	253	414	386	4632		int VecScatterBegin_PtoP(Vec, Vec, Inser
	0.4	2 5 2	253	48	82	5284	<pre>int Mat_AIJ_CheckInode(Mat, PetscTruth) _</pre>

#### PETSc ex19 (Profile - overall and percentages)

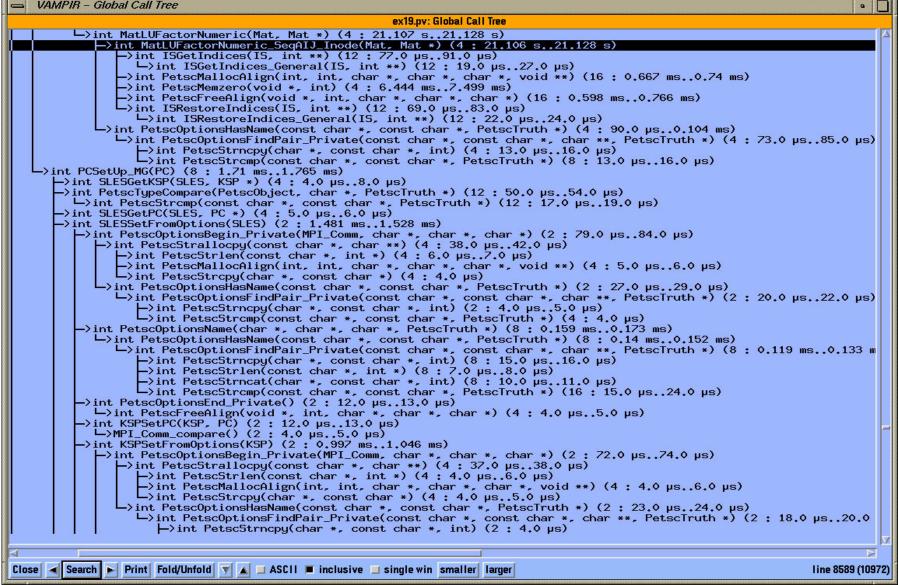


## PETSc ex19 (Tracing)



#### **PETSc** ex19 (Tracing - callgraph)

VAMPIR - Global Call Tree



TAU Parallel Performance System

# PETSc ex19 (PAPI\_FP\_INS, PAPI\_L1\_DCM)

jRacy: Idata/sameer/petsc/src/snes/examples/tutorials/data/PAPI_FP_INS/pprof.dat         File Options Windows Help       PAPI_FP_II         Mean       Image: Comparison of the second secon	Uses multiple counter vs profile measurement
n,c,t 3,0,0 Mean Data Window: IdataIsameerIpetscIsrcIsnesIexamplesItutorialsIdataIPAPI_FP_INSIpprof.dd File Options Windows Help 63.13% int MatLUFactorNumeric_SeqAIJ_Inode(Mat, N 16.71% int MatSolve_SeqAIJ_Inode(Mat, Vec, Vec) 7.97% int MatMult_SeqAIJ_Inode(Mat, Vec, Vec) 5.08% int MatSolve_SeqAIJ_NaturalOrdering(Mat, Vec 2.2% int FormFunctionLocal(DALocalInfo *, Field * 1.27% int VecMAXPY_Seq(int, const PetscScalar *, V	File Options Windows Help       PAPI_L1_DCM         Mean       Image: Contract of the second se
	18.87% int MatSolve_SeqAIJ_Inode(Mat, Vec, Vec) 11.3% int MatSolve_SeqAIJ_NaturalOrdering(Mat, Vec, Vec) 9.14% int MatMult_SeqAIJ_Inode(Mat, Vec, Vec) 2.84% int MatFDColoringApply(Mat, MatFDColoring, Vec, Mat

## **EVH1 – High Energy and Nuclear Physics**



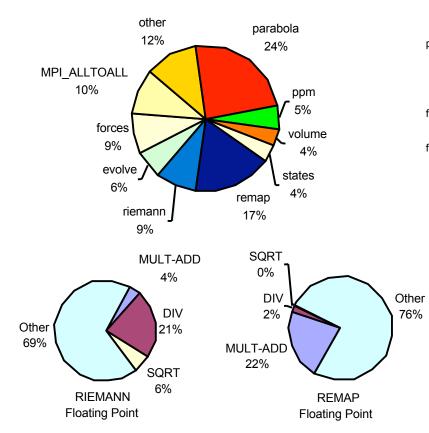
#### □ Enhanced Virginia Hydrodynamics #1 (EVH1)

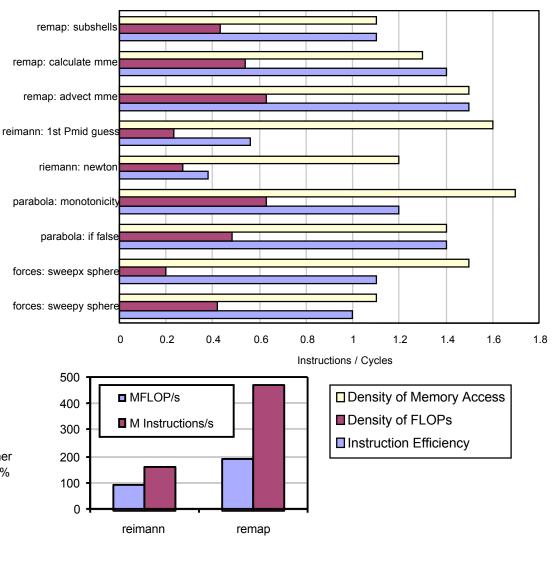
- "TeraScale Simulations of Neutrino-Driven Supernovae and Their Nucleosynthesis" SciDAC project
- Configured to run a simulation of the Sedov-Taylor blast wave solution in 2D spherical geometry
- $\square$  EVH1 communication bound for > 64 processors
  - Predominant routine (>50% of execution time) at this scale is MPI\_ALLTOALL
  - Used in matrix transpose-like operations
  - Current implementation uses 1D matrix decomposition
- □ PERC benchmark code

#### **EVH1 Aggregate Performance**

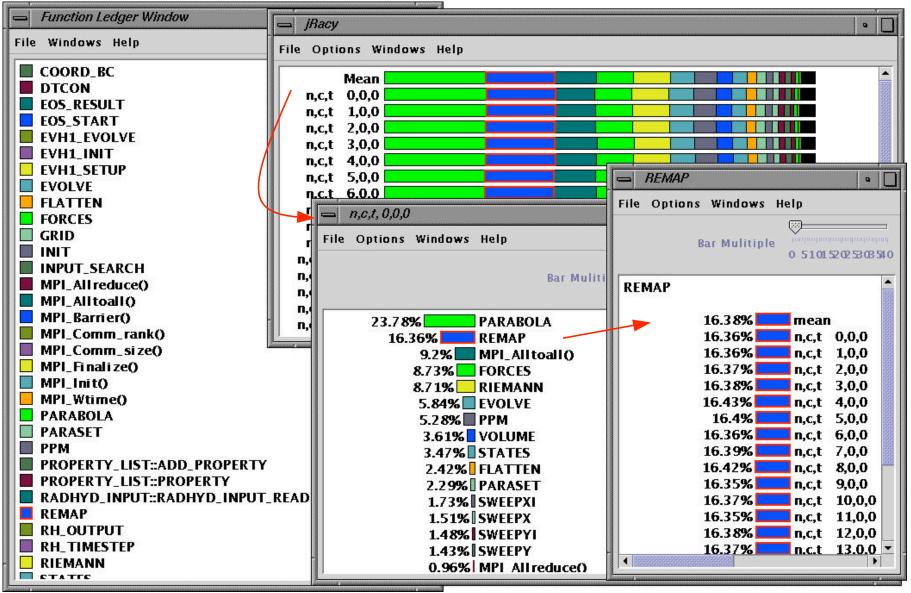
 Aggregate performance measures over all tasks for .1 second simulation

• Using PAPI on IBM SP



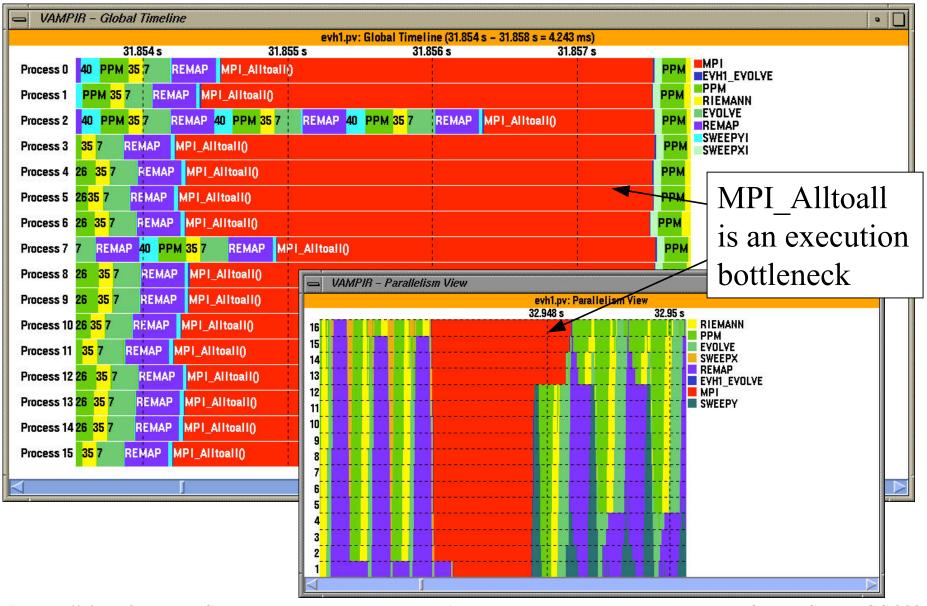


#### **EVH1 Execution Profile**



TAU Parallel Performance System

#### **EVH1 Execution Trace**

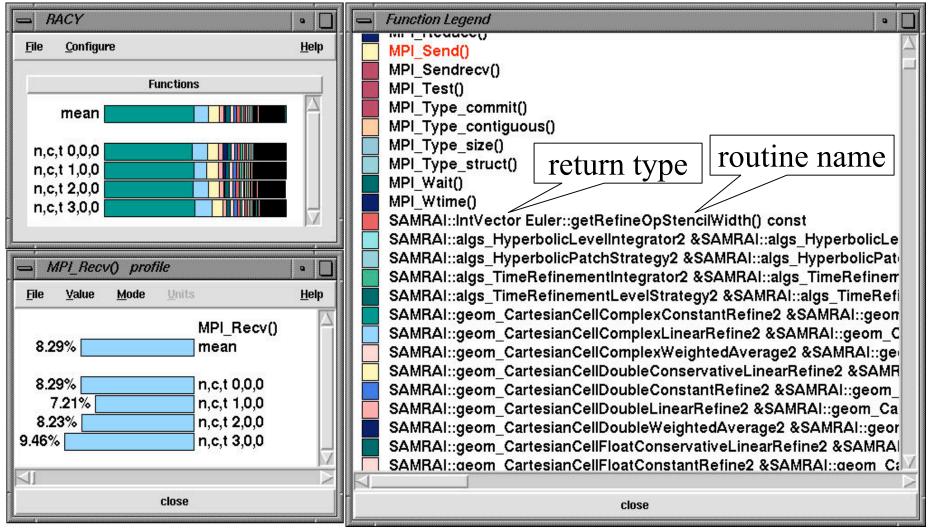


# SAMRAI

- Structured Adaptive Mesh Refinement Application Infrastructure (SAMRAI)
  - O Andy Wissink (LLNL)
- □ Programming
  - C++ and MPI
  - o SPMD
- □ Instrumentation
  - PDT for automatic instrumentation of routines
  - MPI interposition wrappers
  - SAMRAI timers for interesting code segments
    - > timers classified in groups (apps, mesh, ...)
    - timer groups are managed by TAU groups

## **SAMRAI Execution Profile**

#### $\Box$ Euler (2D)



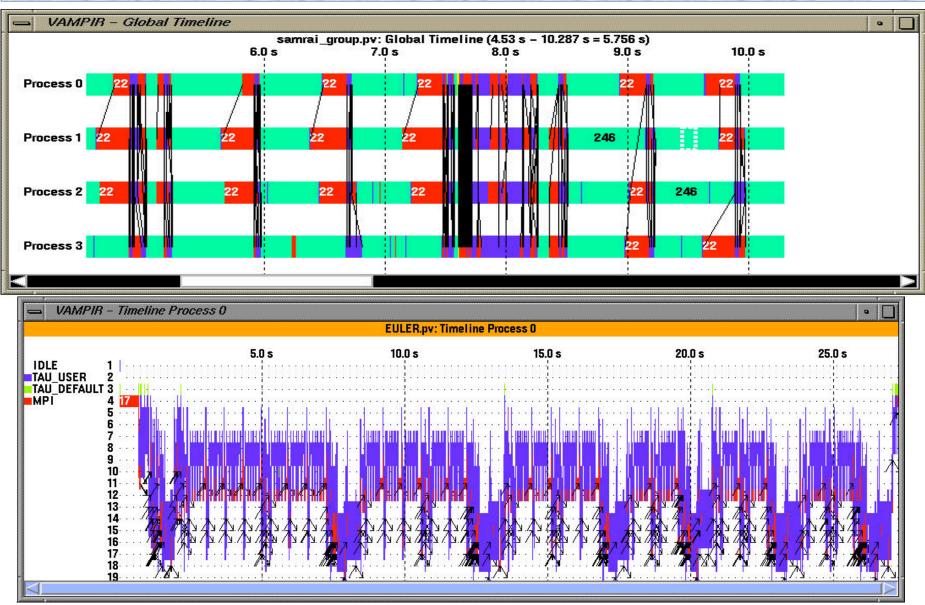
TAU Parallel Performance System



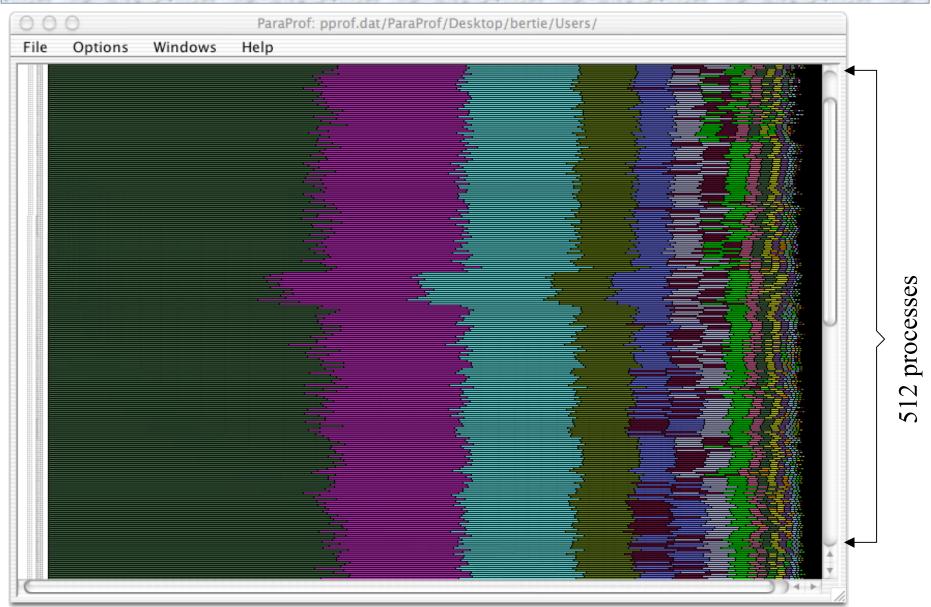
#### SAMRAI Euler Profile

- $n, c, c, c, c$	),0 profile			•
<u>F</u> ile <u>V</u> alue	e <u>O</u> rder	<u>M</u> ode	Units	<u>H</u> elp
			n,c,t 0,0,0	
48.55% 📃			void Euler::computeFluxesOnPatch(SAMRAI::Patch &, double, double)	
	8.29	1.	MPI_Recv()	
			void SAMRAI::xfer_RefineSchedule2::generateCommunicationSchedule	
			void SAMRAI::hier_Patch2::allocatePatchData(const SAMRAI::hier_Com	poner
			MPI_Init()	
			void SAMRAI::tbox_PointerBase::~tbox_PointerBase()	
			void SAMRAI::tbox_ConstPointerBase::~tbox_ConstPointerBase()	
			void Euler::conservativeDifferenceOnPatch(SAMRAI::Patch &, double, d	
			void Euler::postprocessRefine(SAMRAI::Patch &, const SAMRAI::Patch	
		o	SAMRAI::tbox_Pointer <samrai::hier_boxoverlap2> SAMRAI::pdat_Cell</samrai::hier_boxoverlap2>	
		· · · · · · · · · · · · · · · · · · ·	void Euler::tagGradientDetectorCells(SAMRAI::Patch &, int, double, bool	l, int, i
		.10%	MPI_Test()	
41				,
			close	
10				
⇒ void Eu	ler::comput	'eFluxes	sOnPatch(SAMRAI::Patch &, double, double) profile	•
<i>void Eu.</i>  	•	<i>teFluxes</i> <u>U</u> nits		
	•		sOnPatch(SAMRAI::Patch &, double, double) profile	
<u>File V</u> alue	•			Helf
	•		sOnPatch(SAMRAI::Patch &, double, double) profile void Euler::computeFluxesOnPatch(SAMRAI::Patch &, double, double)	
<u>File V</u> alue	•		sOnPatch(SAMRAI::Patch &, double, double) profile void Euler::computeFluxesOnPatch(SAMRAI::Patch &, double, double) mean	
<u>File ⊻</u> alue 49.26% 48.55%	•		sOnPatch(SAMRAI::Patch &, double, double) profile void Euler::computeFluxesOnPatch(SAMRAI::Patch &, double, double)	
<u>File V</u> alue	•		sOnPatch(SAMBAI::Patch &, double, double) profile void Euler::computeFluxesOnPatch(SAMRAI::Patch &, double, double) mean n,c,t 0,0,0	

## SAMRAI Euler Trace



### Full Profile Window (512 Processors)

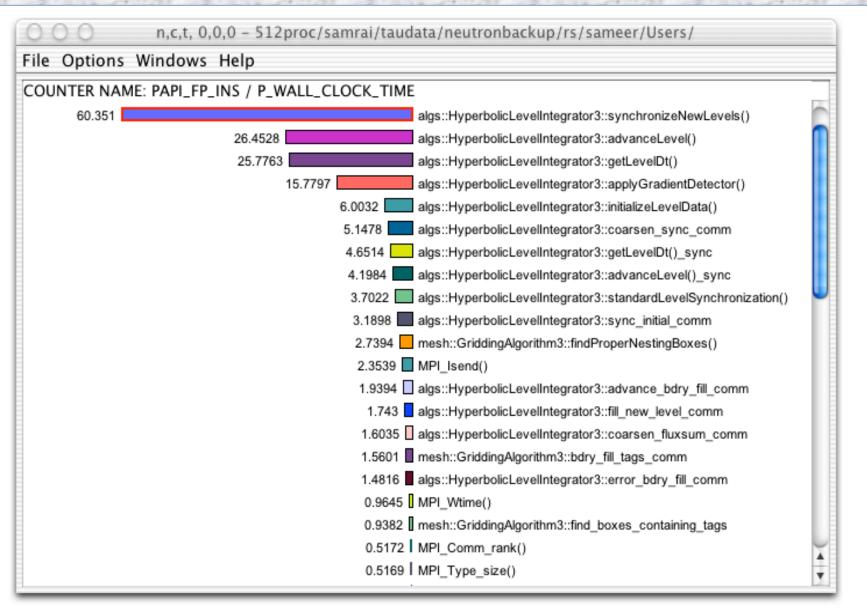




#### Node / Context / Thread Profile Window

000	n,c,t, 0,0,0 – 512proc/samrai/taudata/neutronbackup/rs/sameer/Users/
ile Options Wind	ows Help
COUNTER NAME: F	P_WALL_CLOCK_TIME (seconds)
345.5474	MPI_Allreduce()
	116.4951 algs::HyperbolicLevelIntegrator3::advance_bdry_fill_create
	103.2566 algs::HyperbolicLevelIntegrator3::advanceLevel()
	59.0096 algs::HyperbolicLevelIntegrator3::fill_new_level_create
	37.4482 mesh::GriddingAlgorithm3::load_balance_boxes
	32.8548 algs::HyperbolicLevelIntegrator3::advance_bdry_fill_comm
	21.4095 🔜 mesh::GriddingAlgorithm3::findRefinementBoxes()
	13.4925 📕 algs::HyperbolicLevelIntegrator3::coarsen_fluxsum_create
	12.6572 🗖 algs::HyperbolicLevelIntegrator3::coarsen_sync_create
	10.4408 🗌 mesh::GriddingAlgorithm3::find_boxes_containing_tags
	8.9215 MPI_Init()
	8.6893 emesh::GriddingAlgorithm3::bdry_fill_tags_create
	7.2717 MPI_Bcast()
	7.1321 MPI_Wait()
	4.0833 algs::HyperbolicLevelIntegrator3::error_bdry_fill_comm
	3.6778 MPI_Finalize()
	3.1405 MPI_lsend()
	3.0156 MPI_Waitall()
	2.3457 mesh::GriddingAlgorithm3::remove_intersections_regrid_all
	1.7275 MPI_Test()
	1.6515 algs::HyperbolicLevelIntegrator3::fill_new_level_comm
	1.3919 MPI_Comm_rank()

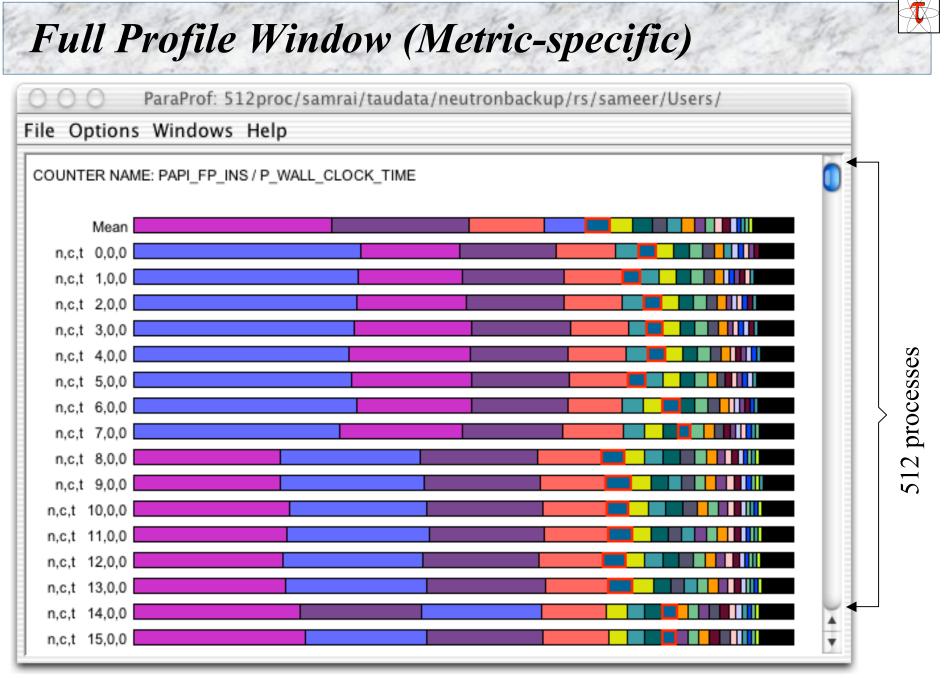
#### **Derived Metrics**



24

#### Paraprof Profile Browser Routine Window

	v: 512proc/samrai/taudata/neutronbackup/rs/sameer/Users/
ile Options Windows Help	
OUNTER NAME: P_WALL_CLOCK_TIME (microse	econds) FUNCTION NAME: algs::HyperbolicLevelIntegrator3::advance_bdry_fill_create
1.174639847793E8	mean
1.16495088E8	n,c,t 0,0,0
1.16779742E8	n,c,t 1,0,0
1.16661784E8	n,c,t 2,0,0
1.17729717E8	n,c,t 3,0,0
1.16718371E8	n,c,t 4,0,0
1.16821729E8	n,c,t 5,0,0
1.17232578E8	n,c,t 6,0,0
1.1684568E8	n,c,t 7,0,0
1.18414461E8	n,c,t 8,0,0
1.16503345E8	n,c,t 9,0,0
1.16891702E8	n,c,t 10,0,0
1.16879062E8	n,c,t 11,0,0
1.17066456E8	n,c,t 12,0,0
1.16470525E8	n,c,t 13,0,0
1.16652028E8	n,c,t 14,0,0
1.16801268E8	n,c,t 15,0,0
1.16830014E8	n,c,t 16,0,0
1.16811997E8	n,c,t 17,0,0
1.16986149E8	n,c,t 18,0,0
1.17786889E8	n,c,t 19,0,0
1.16936729E8	n,c,t 20,0,0
1.17203411E8	n,c,t 21,0,0
1.17213872E8	n,c,t 22,0,0
1.17491906E8	n,c,t 23,0,0
1.16756319E8	n,c,t 24,0,0



TAU Parallel Performance System

## Mixed-mode Parallel Programs (OpenMPI + MPF)

- □ Portable mixed-mode parallel programming
  - Multi-threaded shared memory programming
  - Inter-node message passing
- Performance measurement
  - Access to RTS and communication events
  - Associate communication and application events
- □ 2D Stommel model of ocean circulation
  - OpenMP for shared memory parallel programming
  - MPI for cross-box message-based parallelism
  - Jacobi iteration, 5-point stencil
  - Timothy Kaiser (San Diego Supercomputing Center)

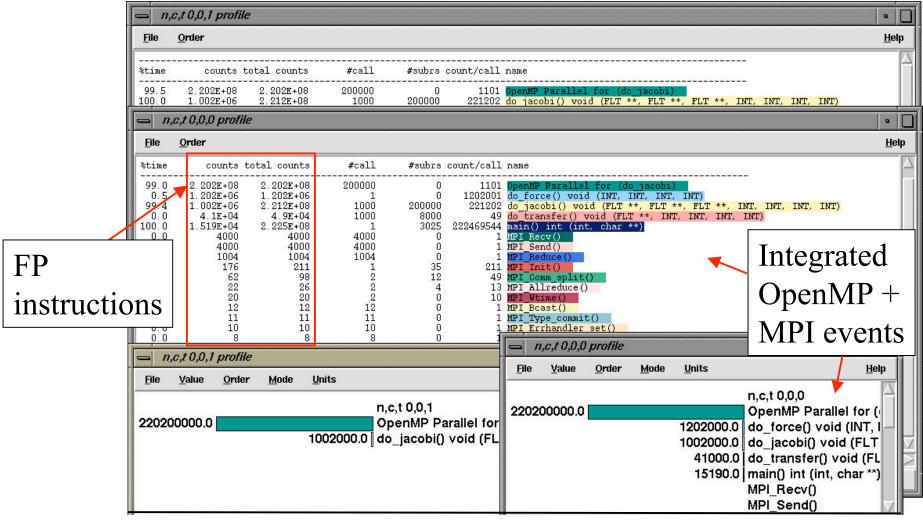
#### **Stommel Instrumentation**

#### OpenMP directive instrumentation

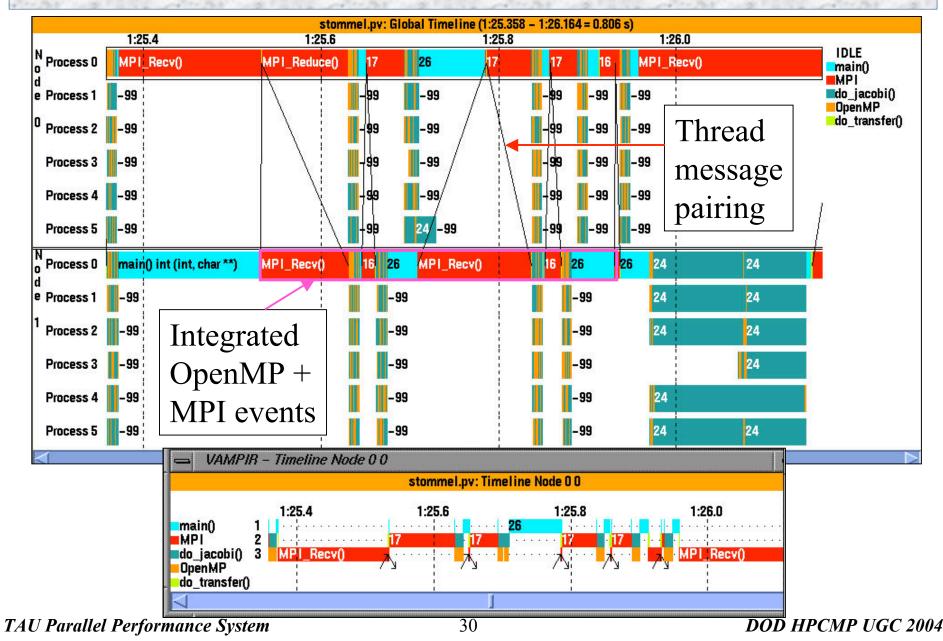
```
pomp for enter(&omp rd 2);
#line 252 "stommel.c"
#pragma omp for schedule(static) reduction(+: diff) private(j)
  firstprivate (a1,a2,a3,a4,a5) nowait
for( i=i1;i<=i2;i++) {</pre>
  for(j=j1;j<=j2;j++) {</pre>
   new_psi[i][j]=a1*psi[i+1][j] + a2*psi[i-1][j] + a3*psi[i][j+1]
      + a4*psi[i][j-1] - a5*the for[i][j];
   diff=diff+fabs(new psi[i][j]-psi[i][j]);
  }
pomp barrier enter(&omp rd 2);
#pragma omp barrier
pomp barrier exit(&omp rd 2);
pomp for exit(&omp rd 2);
#line 261 "stommel.c"
```

## **OpenMP + MPI Ocean Modeling (HW Profile)**

% configure -papi=../packages/papi -openmp -c++=pgCC -cc=pgcc -mpiinc=../packages/mpich/include -mpilib=../packages/mpich/libo



## **OpenMP + MPI Ocean Modeling (Trace)**



# HYbrid Coordinate Ocean Model (HYCOM)



- Primitive equation ocean circulation model (MICOM)
- Improved vertical coordinate scheme that remains isopycnic in the open, stratified ocean
- Transitions smoothly
  - To *z-level* coordinates in the weakly-stratified upperocean mixed layer, to sigma coordinates in shallow water conditions and back to *z-level* coordinates in very shallow water conditions
- □ User has control over the model domain
  - Generating the forcing field
- Dr. Avi Purkayastha





## Getting started with HYCOM

- □ For generation of the serial or parallel executables
  - Make.com => ../config/\$(ARCH)\_\$(TYPE)
  - Contains all Makefile macro definitions for the preprocessor, fortran, C, parser, and instrumentor compile and link options and associated libraries
- Serial runs
  - 2.00 degree Atlantic Ocean regional grid was used for input, without any change to domain or resolution
- Parallel runs
  - O Global ocean (GLBA0.24) was the domain
  - Performance analysis was carried out only for the MPI version of HYCOM

## **Gprof Profile Data on HYCOM**

#### **Gprof profile data conclusions**

- Prime candidates for optimization
  - > momtum
  - > tsadvc
  - *≻ mx\**
  - ➤ cnuity
- O Mathematical functions also contribute significantly
   *> atan* and *sqrt*

## TAU Profile Analysis of HYCOM

- Exclusive time shows the relative largest time consuming functions
  - O Including, surprisingly, MPI\_Waitall
- Exclusive time spent can be used as an indicator for measuring efficiency of these functions
  - For example, obtaining MFLOP rates

%Time	Exclusive msec	Inclusive total msec	#call	#subrtns	Inclusive usec/call	Name
100.0	8:25.60	49:01.37	1	54686.1	2941369933	НҮСОМ
34.3	16:47.70	16:47.70	54191.4	0	18595	MPI_Waitall
32.1	10:41.98:6	15:43.07	192	113178	4911808	TSADVC
19.7	:11.67	9:40.60	192	31488	3023979	MOMTUM
9.3	2:12.77	4:33.82	192	20601.6	1426132	CNUITY

Tau profile data excerpt with the highest time-consuming functions

#### TAU Profile Analysis of HYCOM

- Low and high end of the time variance spent by the MPI\_Waitall call in some of the processors
- Additional investigation is then required from the tracefiles for better understanding overall communication model

Proc #	%Time	Exclusive msec	Inclusive total msec	#call	Inclusive usec/call
13	4.7	2:17.33	2:17.33	54182	2535
14	5.6	2:43.63	2:43.63	54182	3020
18	80.6	39:31.07	39:31.07	54229	43723
23	75.3	36:54.28	36:54.28	54182	40867
24	81.1	39:45.41	39:45.41	54229	43988

Tau profile data excerpt highlighting load imbalance on MPI\_Waitall

#### **PAPI profile analysis of HYCOM**

 PAPI profiling (obtained with Tau) exclusive operation count can show performance of individual functions
 • TSADVC

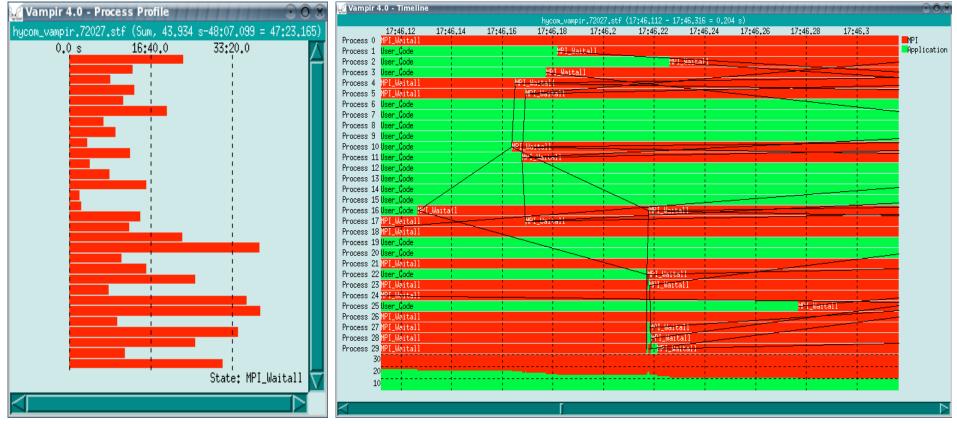
> 2.07e+11/10:41.98(=642s) = 323 Mflops (~6% of peak)

%Time	Exclusive counts	Inclusive total counts	#call	#subrtns	Inclusive usec/call	Name
100.0	1.56e+11	5.25e+11	1	54686.1	52543238273	НҮСОМ
39.3	2.07e+11	2.08e+11	192	113178	1085499128	TSADVC
22.2	1.15e+11	1.16e+11	192	31488	606724005	MOMTUM
8.3	4.32e+10	4.37e+10	192	20601.6	227820049	CNUITY
0.5	2.57e+09	2.57e+09	54191.4	0	47497	MPI_Waitall

Papi profile data excerpt with the highest time-consuming functions

### Vampir Trace analysis of HYCOM

- □ *MPI\_Waitall* time per process shows the wide disparity
  - Time spent for blocking call to return => load imbalance
- Small snapshot of global timeline indicates *MPI\_Waitall* waiting on non-blocking receive operations to complete

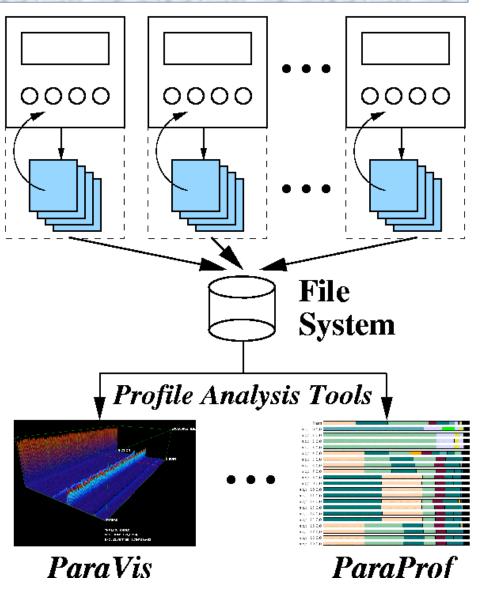


#### Vampir Trace analysis of HYCOM

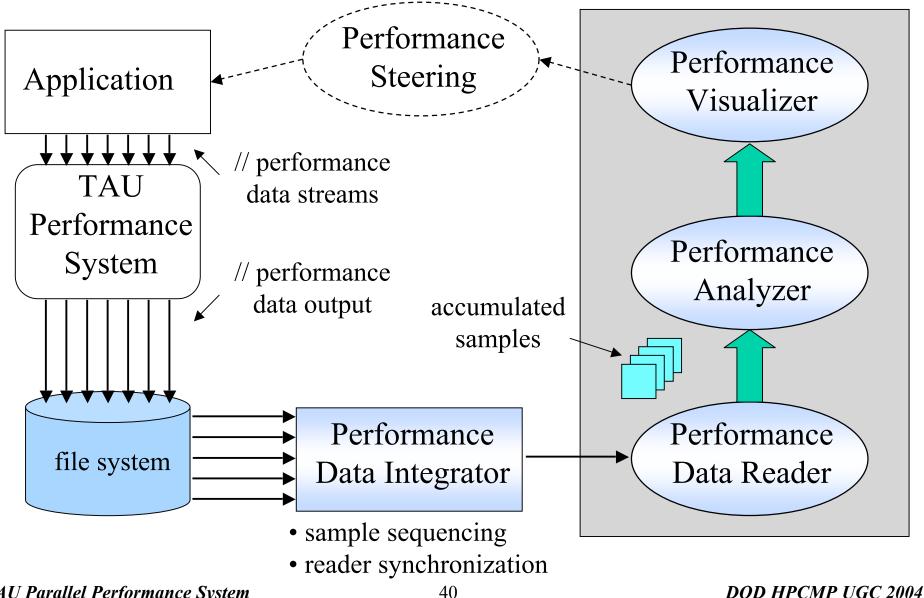
- This kind of load imbalance is a generic problem for structured-grid ocean models caused by variations in amount of ocean per tile
- HYCOM avoids all calculations over land, so the load imbalance leads to long intervals spent in MPI\_Waitall on processors that "own" little ocean
- A different MPI strategy is perhaps necessary to reduce the MPI overhead on those processors with the most computational overhead which is the main contributing factors for those processors waiting for non-blocking receive operations

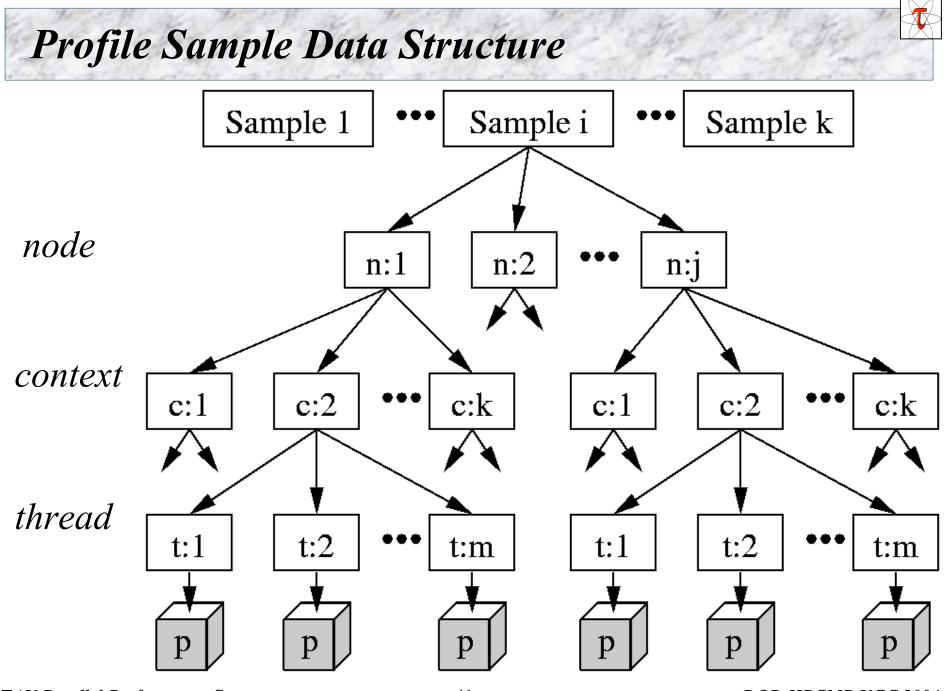
Online Profile Measurement and Analysis in TAU

- □ Standard TAU profiling
  - Per node/context/thread
- □ Profile "dump" routine
  - O Context-level
  - Profile file per each thread in context
  - O Appends to profile file
  - O Selective event dumping
- Analysis tools access files through shared file system
- Application-level profile
   "access" routine



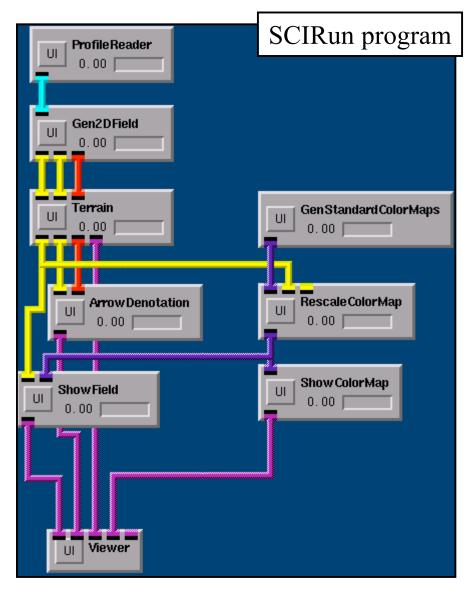
**Online Performance Analysis and Visualization** 

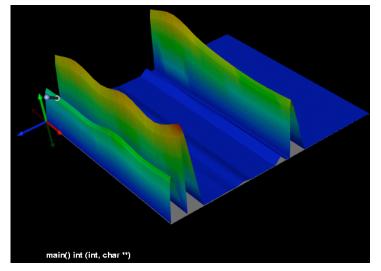


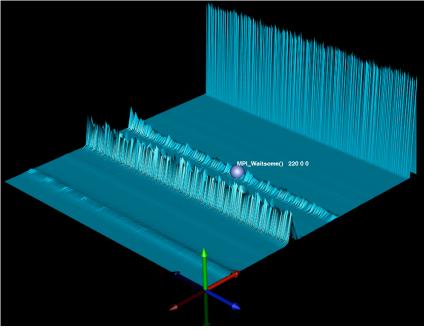


TAU Parallel Performance System

# Performance Analysis/Visualization in SCIRun







TAU Parallel Performance System

# Uintah Computational Framework (UCF)

ViewWindow

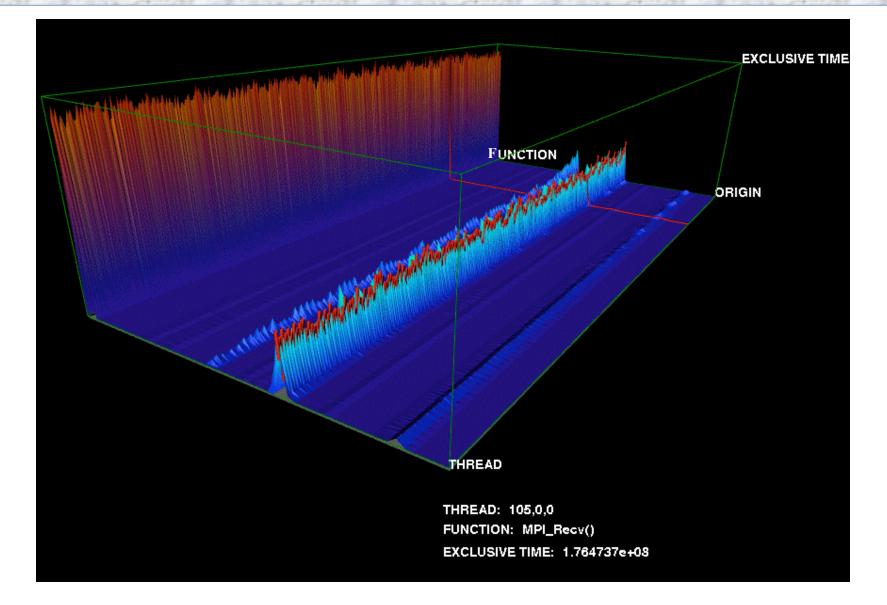
Edit Visual

□ University of Utah □ UCF analysis • Scheduling O MPI library • Components □ 500 processes □ Use for online and offline visualization □ Apply SCIRun steering

SerialMPM::interpolateParticlesToGrid [MPIScheduler::execute()] 403 0 0 55232 polygons in 0.6 seconds Autoview Set Home View 92053.3 polygons/second 0.1 frames/sec Go home Views

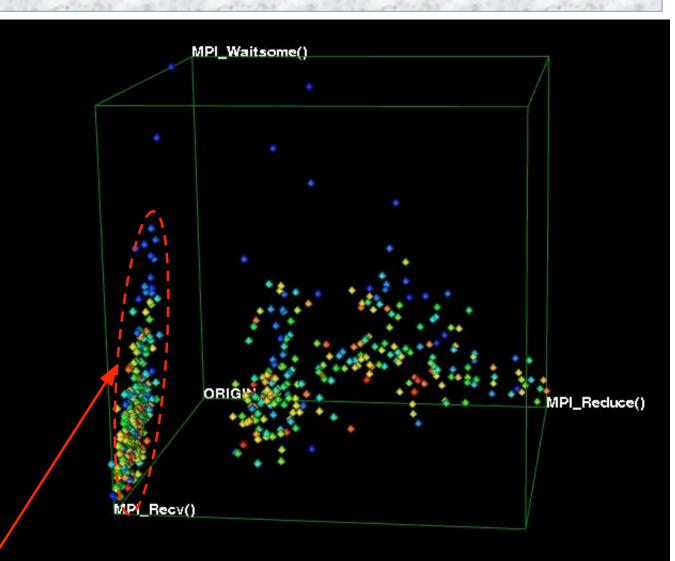
• 🗌

### "Terrain" Performance Visualization



### Scatterplot Displays

**Each** point coordinate determined by three values: MPI Reduce MPI Recv MPI Waitsome □ Min/Max value range □ Effective for cluster analysis

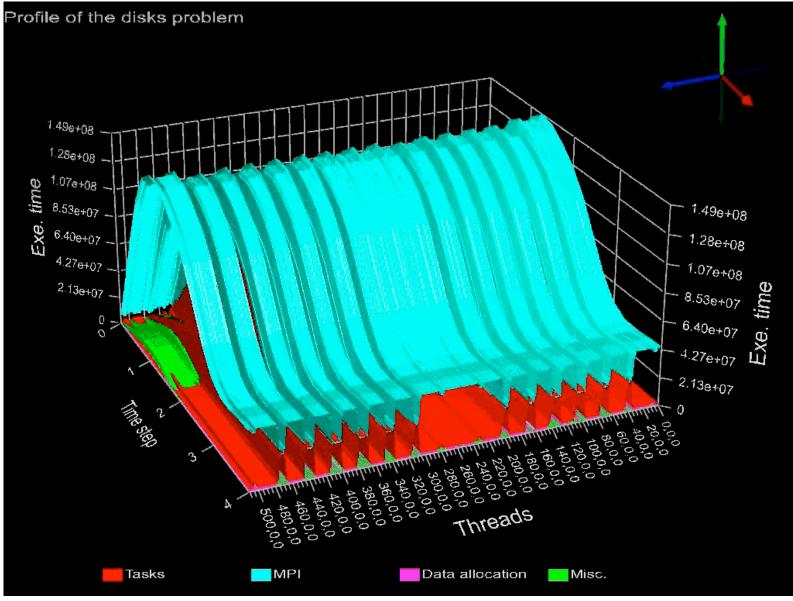


• Relation between MPI\_Recv and MPI\_Waitsome

### **Online Unitah Performance Profiling**

- Demonstration of online profiling capability
- Colliding elastic disks
  - Test material point method (MPM) code
  - Executed on 512 processors ASCI Blue Pacific at LLNL
- □ Example 1 (Terrain visualization)
  - Exclusive execution time across event groups
  - Multiple time steps
- □ Example 2 (Bargraph visualization)
  - MPI execution time and performance mapping
- □ Example 3 (Domain visualization)
  - Task time allocation to "patches"

# Example 1 (Event Groups)



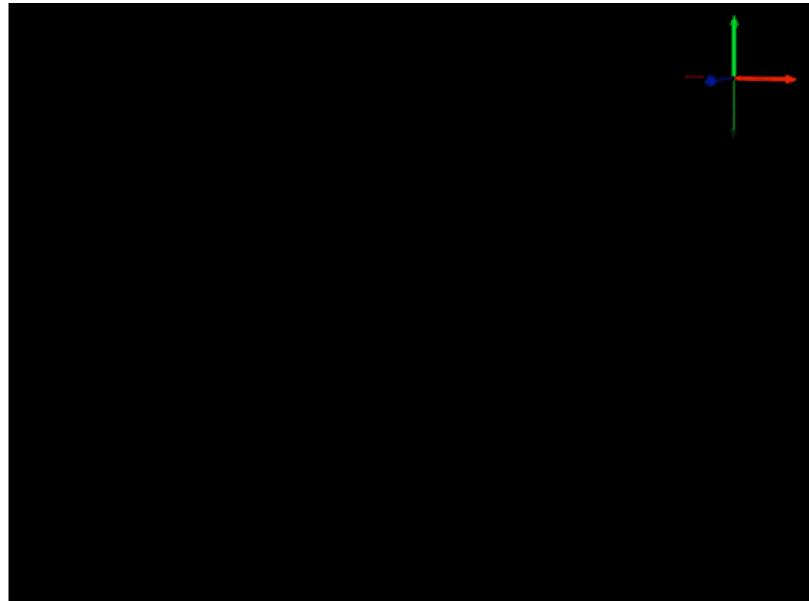
TAU Parallel Performance System

### Example 2 (MPI Performance)



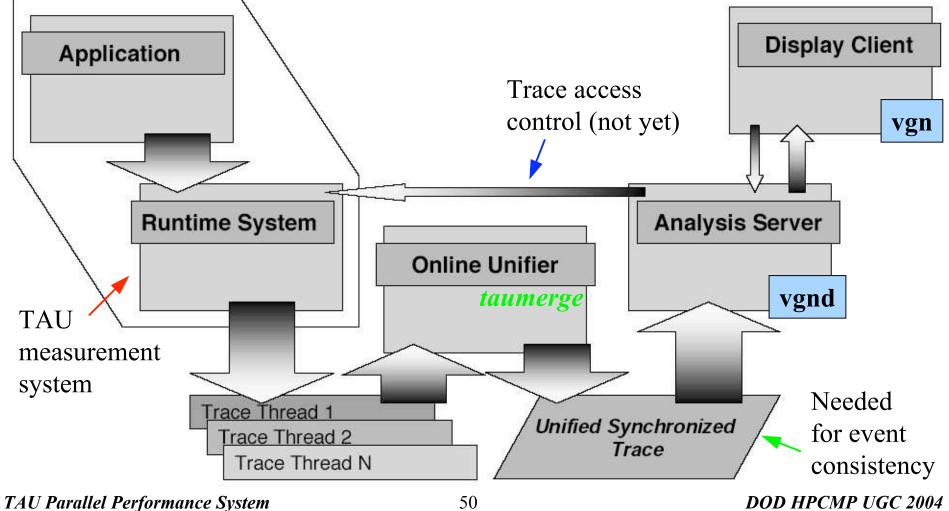


# **Example 3 (Domain-Specific Visualization)**

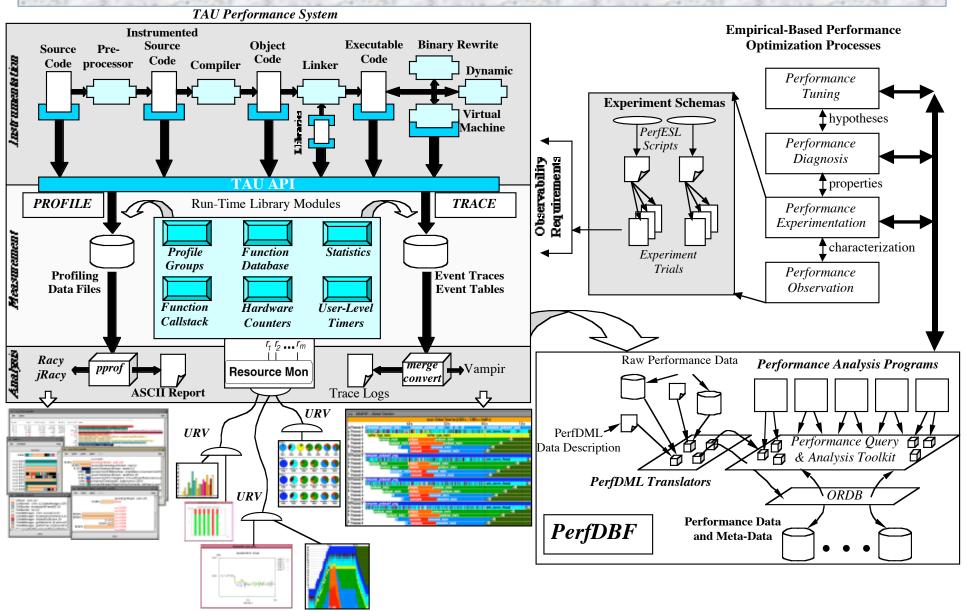


# **Online Trace Analysis with TAU and VNG**

- **X**
- □ TAU measurement of application to generate traces
- □ Write traces (currently) to NFS files and unify



### Integrated Performance Evaluation Environment



TAU Parallel Performance System