

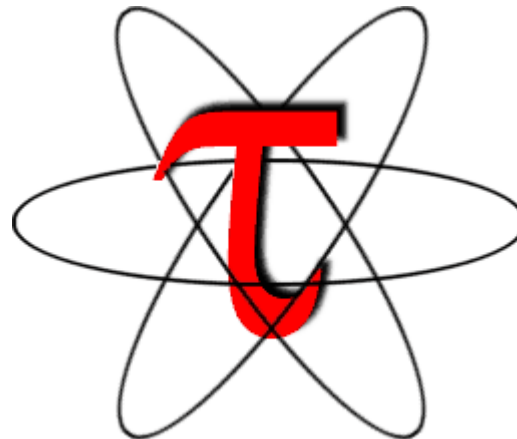
The TAU Performance Technology for Complex Parallel Systems

*(Performance Analysis Bring Your Own Code Workshop,
NRL Washington D.C.)*

Sameer Shende, Allen D. Malony, Robert Bell

University of Oregon

{sameer, malony, bertie}@cs.uoregon.edu



Tuning and Analysis Utilities



John von Neumann - Institut für Computing
Zentralinstitut für Angewandte Mathematik

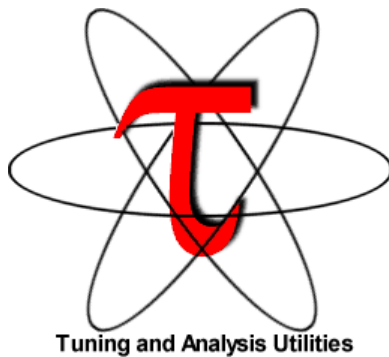


Outline



- ❑ Motivation
- ❑ Part I: Instrumentation
- ❑ Part II: Measurement
- ❑ Part III: Analysis Tools
- ❑ Conclusion

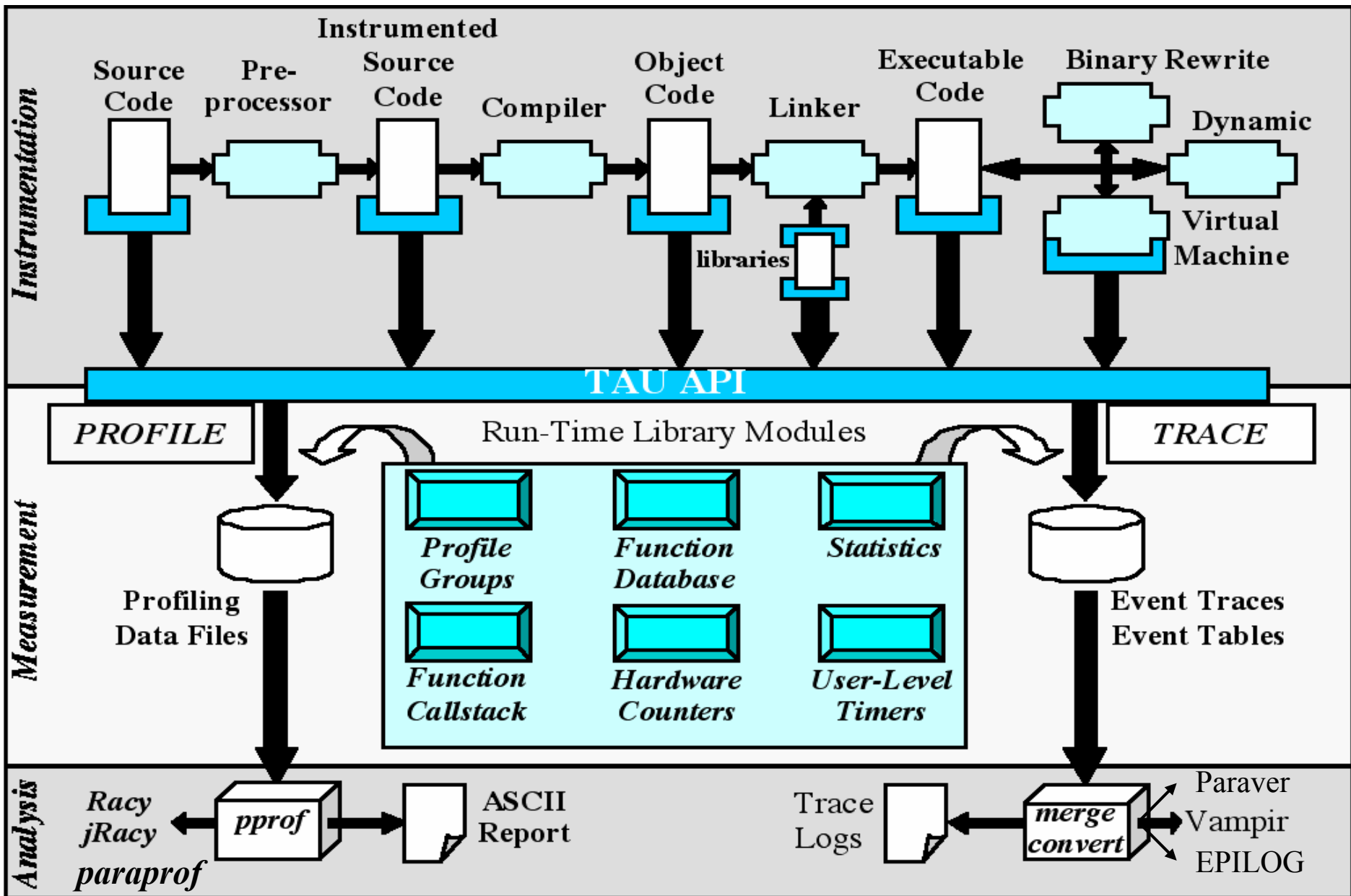
TAU Performance System Framework



- ❑ Tuning and Analysis Utilities
- ❑ Performance system framework for scalable parallel and distributed high-performance computing
- ❑ Targets a general complex system computation model
 - nodes / contexts / threads
 - Multi-level: system / software / parallelism
 - Measurement and analysis abstraction
- ❑ Integrated toolkit for performance instrumentation, measurement, analysis, and visualization
 - Portable, configurable **performance profiling/tracing facility**
 - Open software approach
- ❑ University of Oregon, LANL, FZJ Germany
- ❑ <http://www.cs.uoregon.edu/research/paracomp/tau>



TAU Performance System Architecture





□ Parallel profile analysis

○ *pprof*

- parallel profiler with text-based display

○ *paraprof*

- Graphical, scalable, parallel profile analysis and display

□ Trace analysis and visualization

- Trace merging and clock adjustment (if necessary)
- Trace format conversion (ALOG, SDDF, VTF, Paraver)
- Trace visualization using *Vampir* (Pallas/Intel)

Pprof Output (ESMF CoupledFlowSolver)



□ IBM AIX

□ F95,
C++,
C,
MPI

□ Profile

- Node

- Context

- Thread

□ Events

- code

- MPI

emacs@neuronic.nic.uoregon.edu

File Edit Options Buffers Tools Help

Reading Profile files in profile.*

NODE 0: CONTEXT 0: THREAD 0:

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive Name usec/call
100.0	3	42,766	1	24	42766029 ESMF_APPLICATIONWRAPPER
95.2	12	40,693	3612	3612	11266 void c_esmc_ftablecallentrypointvm(ESMC_VM **, ESMC_VMPlan **, void **, void **, ESMC_FTable **, char *, int *, int *, int *) C
95.2	172	40,693	3612	10836	11266 void *vmachine::vmachine_enter(vmplan &, void (*)(void *, void *), void *) vmachine
95.1	11	40,690	3612	10836	11265 void *vmachine_spawn(void *)
95.1	6	40,690	3612	3612	11265 void *ESMC_FTableCallEntryPointVMHop(void *, void *) C
95.1	25	40,690	3612	7224	11265 int ESMC_FTable::ESMC_FTableCallVFuncPtr(char *, ESMC_VM *, int *) ESMC_FTable
94.9	6	40,600	1801	1801	22543 ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPRUN
94.9	35	40,600	3601	18005	11275 ESMF_COMPMOD::ESMF_COMPRUN
94.9	16	40,599	1	5403	40599938 COUPLEDFLOWMOD::COUPLEDFLOW_RUN
61.4	264	26,260	900	18905	29179 FLOWSOLVERMOD::FLOWSOLVE
31.0	52	13,254	22500	589	ESMF_ROUTEMOD::ESMF_ROUTERUN
30.9	32	13,201	22500	22500	587 void c_esmc_routerunla(ESMC_Route **, ESMC_LocalArray **, ESMC_LocalArray **, int *) C
30.8	2,409	13,168	22500	150300	585 int ESMC_Route::ESMC_RouteRun(void *, void *, ESMC_DataKind)
30.2	6	12,901	1800	1800	7167 ESMF_CPLCOMPMOD::ESMF_CPLCOMPRUN
28.0	129	11,983	1800	14400	6658 COUPLERMOD::COUPLER_RUN
27.7	66	11,850	12600	25200	941 ESMF_FIELDCOMMOD::ESMF_FIELDEDIST
27.4	62	11,736	12600	25200	931 ESMF_ARRAYCOMMOD::ESMF_ARRAYREDISTNEW
25.0	104	10,700	49896	98568	214 int ESMC_Delayout::ESMC_DelayoutCopy(void **, void **, int, int, int, ESMC_Logical)
25.0	43	10,682	22500	45000	475 int ESMC_Delayout::ESMC_DelayoutExchange(void **, void **, void **, void **, int, int, int, int, ESMC_Logical)
14.7	0,945	6,276	5	66	1255337 FLOWSOLVERMOD::FLOWPRINT
14.5	0,161	6,200	17	17	364757 ESMF_ARRAYMOD::ESMF_ARRAYWRITE
14.5	0,143	6,200	17	17	364748 void c_esmc_arraywrite(ESMC_Array **, char *, char *, int *, int, int) C
14.5	6,200	6,200	17	0	364739 int ESMC_Array::ESMC_ArrayWrite(const char *, const char *) const
12.6	75	5,373	26172	26172	205 void vmachine::vmachine_recv(void *, int, int) vmachine
12.4	4,690	5,314	900	1800	5905 FLOWSOLVERMOD::FLOWSTATE
12.4	5,297	5,297	26172	0	202 MPI_Recv()
12.1	71	5,195	22500	22500	231 void vmachine::vmachine_send(void *, int, int) vmachine
12.0	5,123	5,123	22500	0	228 MPI_Send()
10.8	4,229	4,599	900	900	5111 FLOWSOLVERMOD::FLOWRHOI
9.9	3,032	4,236	900	3600	4707 FLOWSOLVERMOD::FLOWVEL
8.1	328	3,466	9900	118802	350 ESMF_FIELDCOMMOD::ESMF_FIELDHALODEPRECAT
7.8	2,719	3,345	900	1800	3717 FLOWSOLVERMOD::FLOWRHOVEL
5.1	1,529	2,172	900	1800	2414 FLOWSOLVERMOD::FLOWRHO

txt (Fundamental)--L19--Top



Terminology – Example

- ❑ For routine “int main()”:
- ❑ Exclusive time
 - $100 - 20 - 50 - 20 = 10$ secs
- ❑ Inclusive time
 - 100 secs
- ❑ Calls
 - 1 call
- ❑ Subrs (no. of child routines called)
 - 3
- ❑ Inclusive time/call
 - 100secs

```
int main( )  
{ /* takes 100 secs */  
  
    f1(); /* takes 20 secs */  
    f2(); /* takes 50 secs */  
    f1(); /* takes 20 secs */  
  
    /* other work */  
}  
  
/*  
Time can be replaced by counts  
*/
```

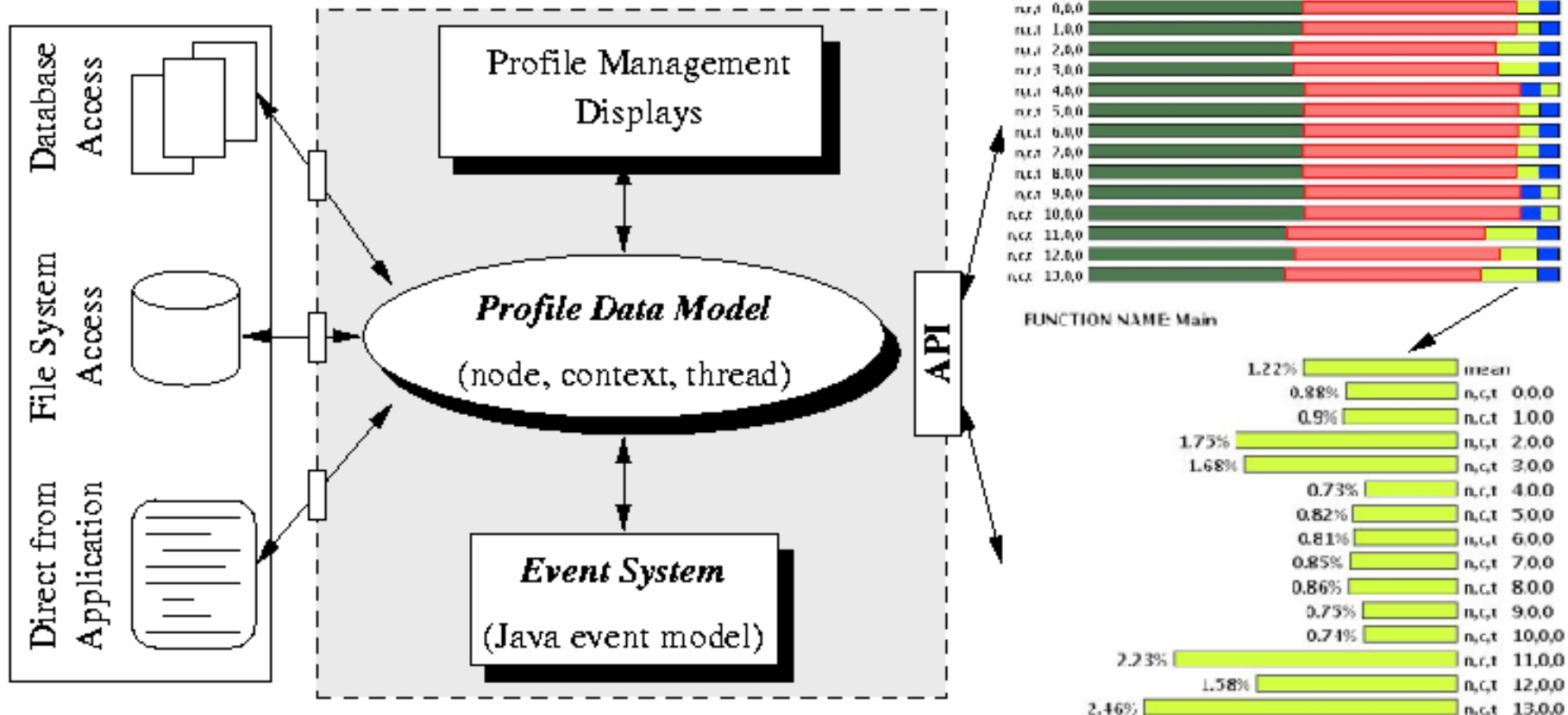
Performance Analysis and Visualization



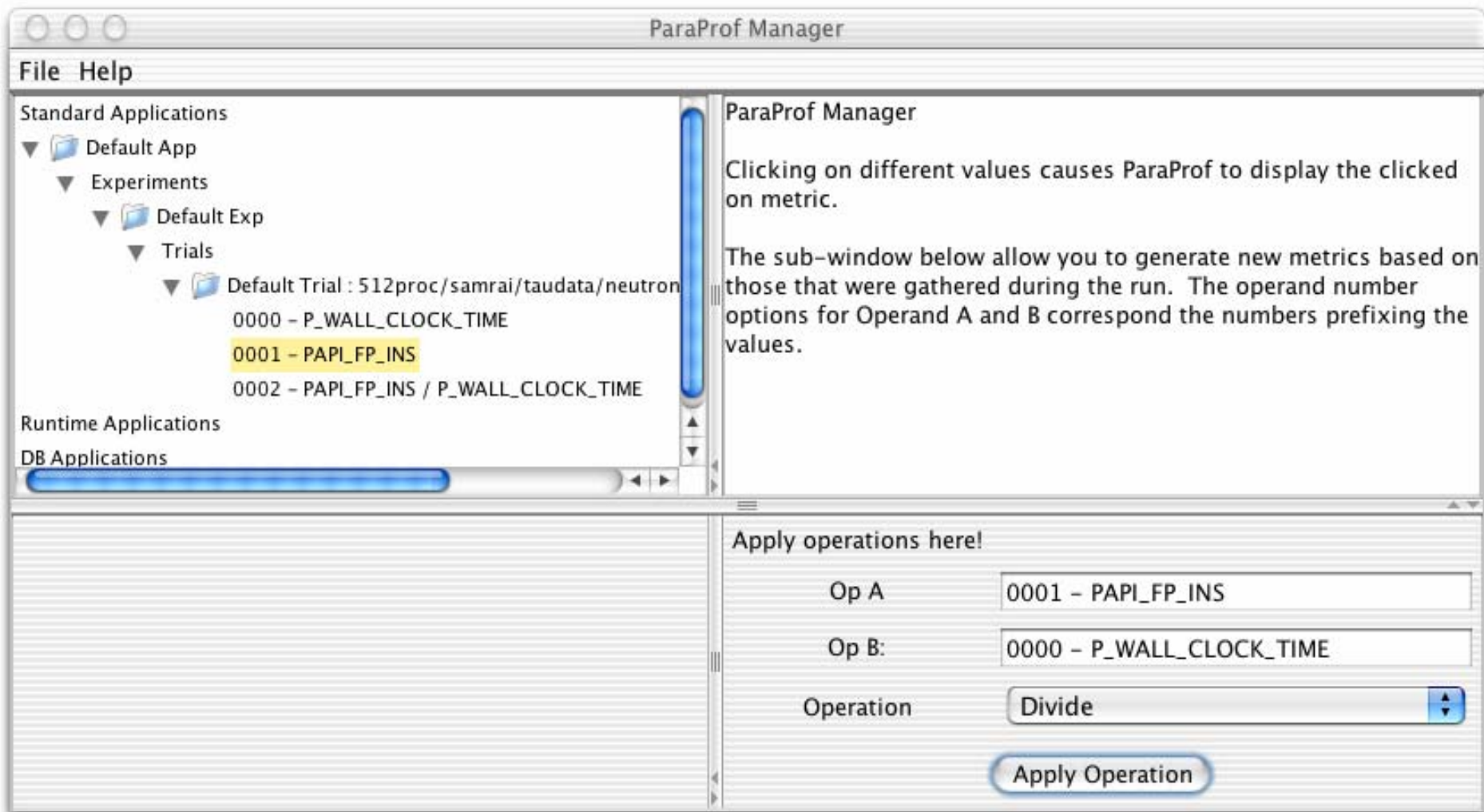
- ❑ Analysis of parallel profile and trace measurement
- ❑ Parallel profile analysis
 - ParaProf
 - Cube Profile Browser (UTK, FZJ)
 - Profile generation from trace data
- ❑ Performance data management framework (PerfDMF)
- ❑ Parallel trace analysis
 - Translation to VTF 3.0 and EPILOG
 - Integration with VNG (Technical University of Dresden)
- ❑ Online parallel analysis and visualization

TAU's ParaProf Framework Architecture

- Portable, extensible, and scalable tool for profile analysis
- Try to offer “best of breed” capabilities to analysts
- Build as profile analysis framework for extensibility

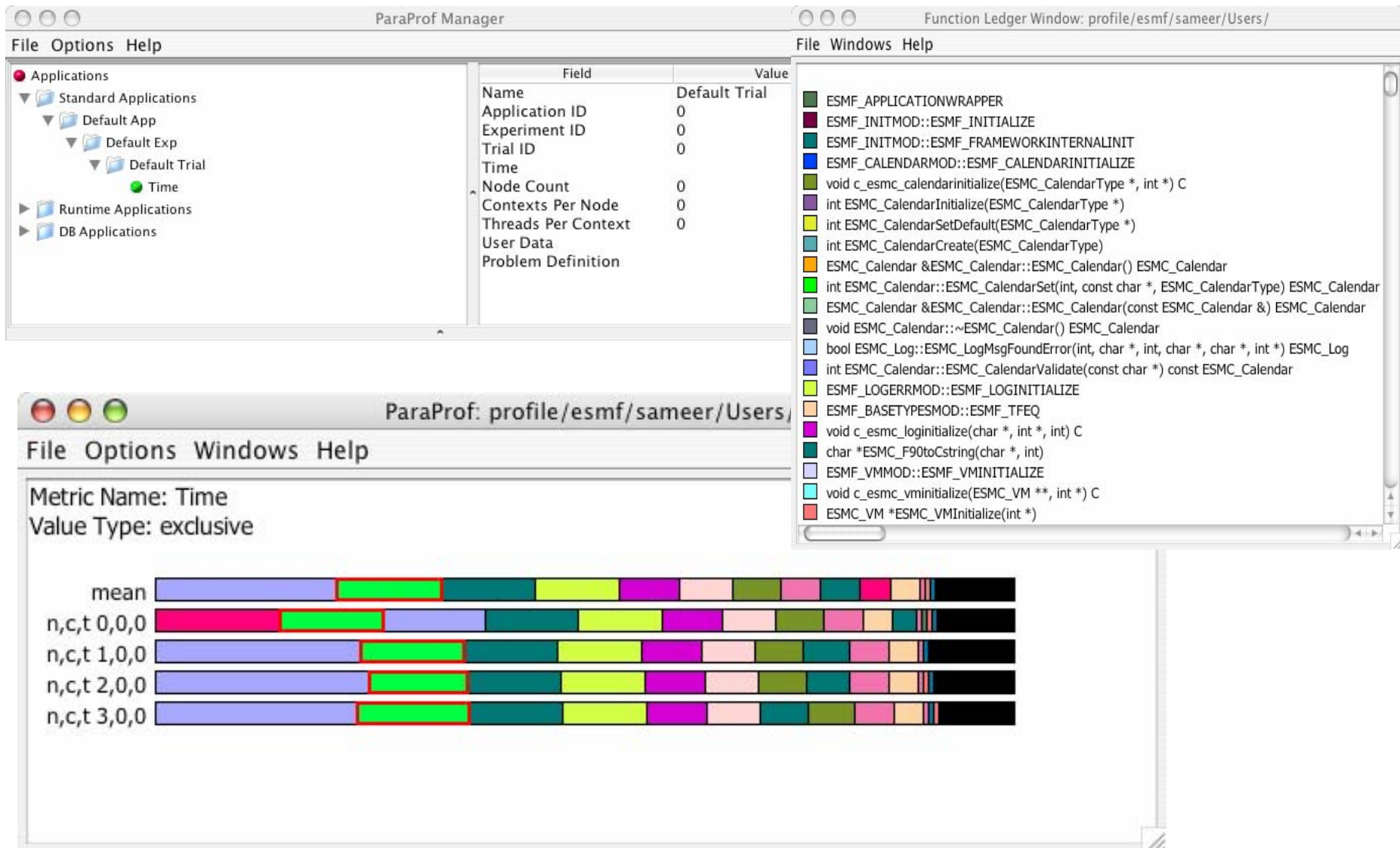


Profile Manager Window

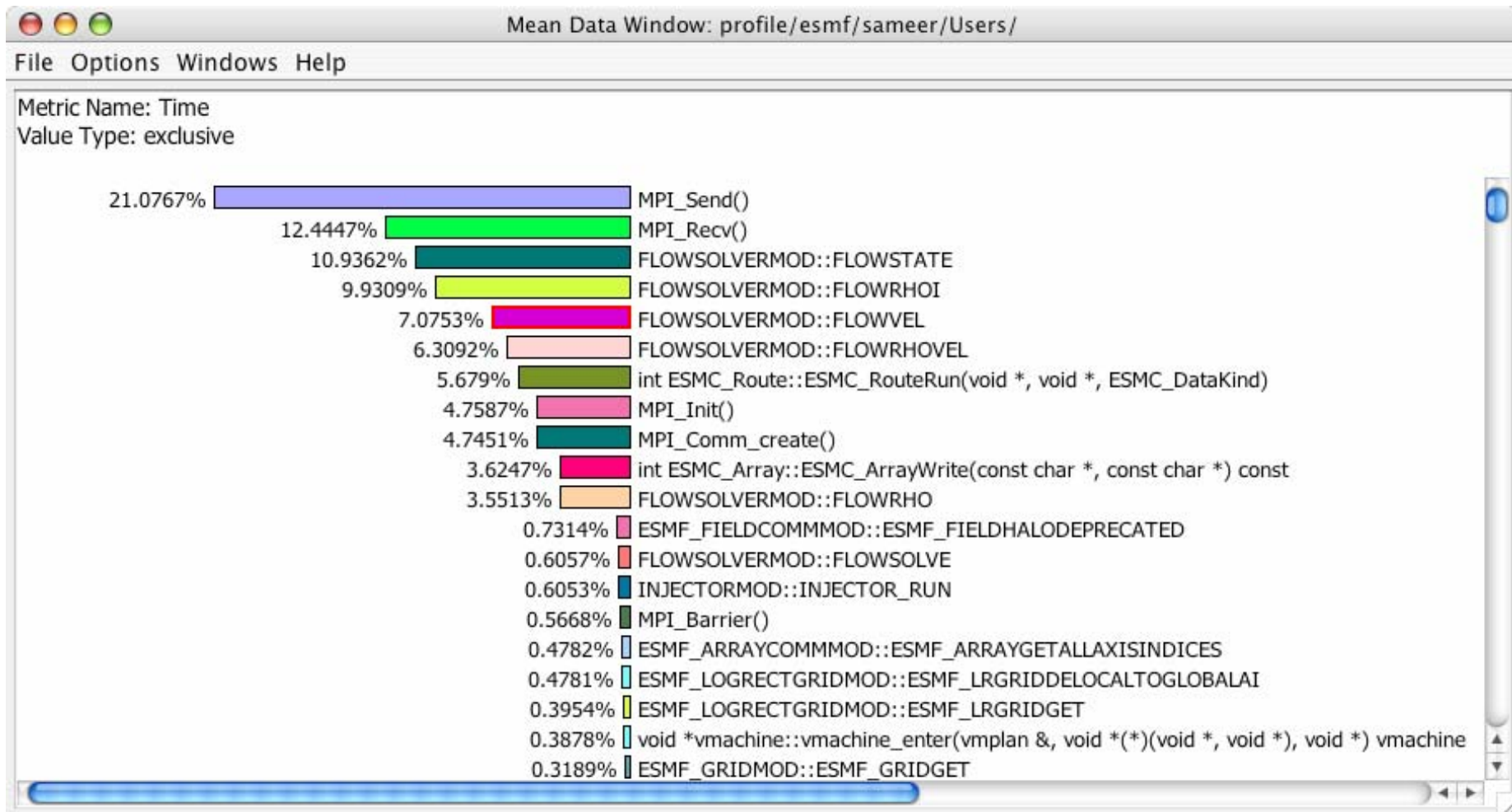


❑ Structured AMR toolkit (SAMRAI++), LLNL

Paraprof: CoupledFlowApp (ESMF) on 4 Nodes

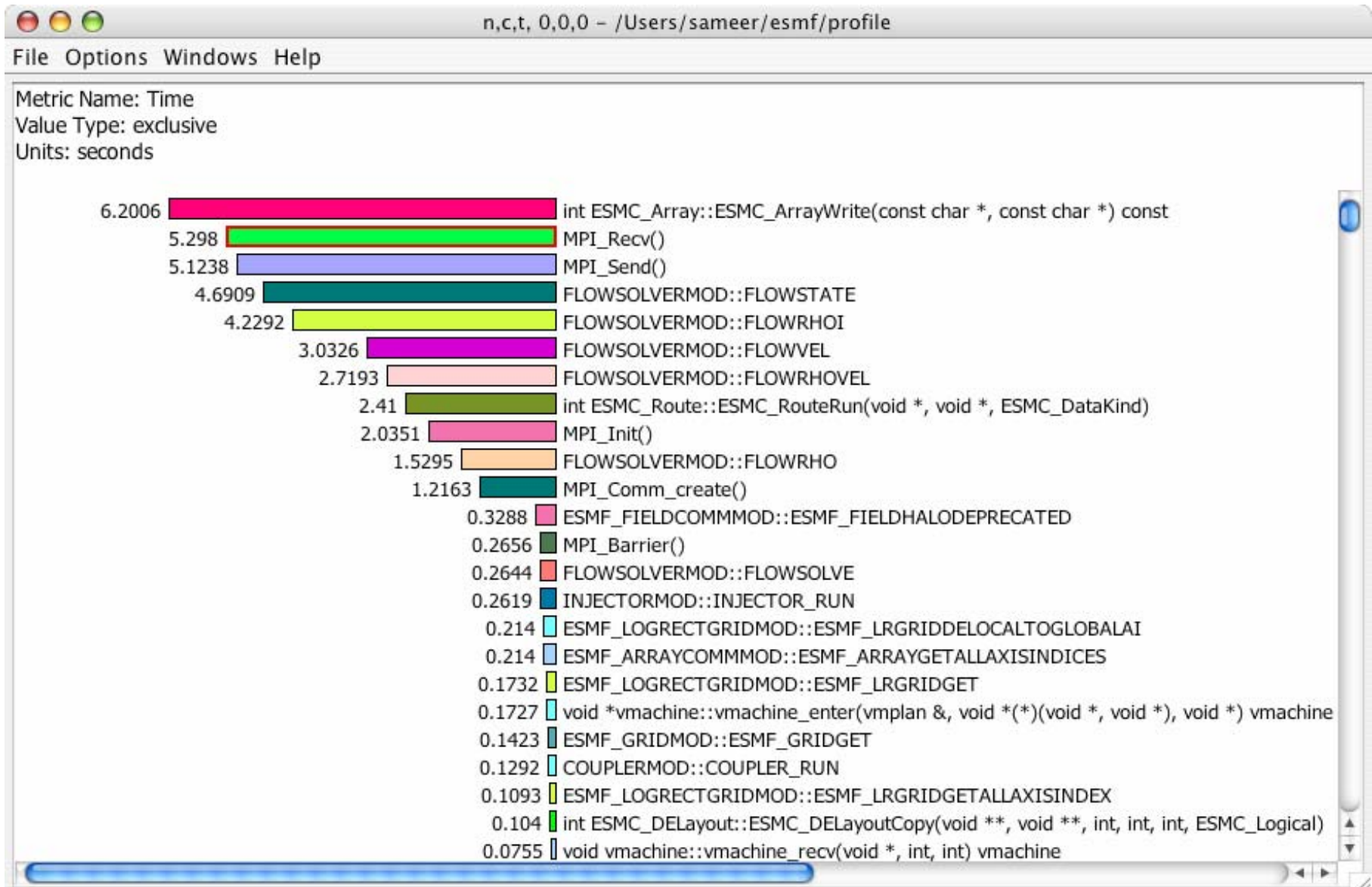


Paraprof Mean Profile (4 nodes)

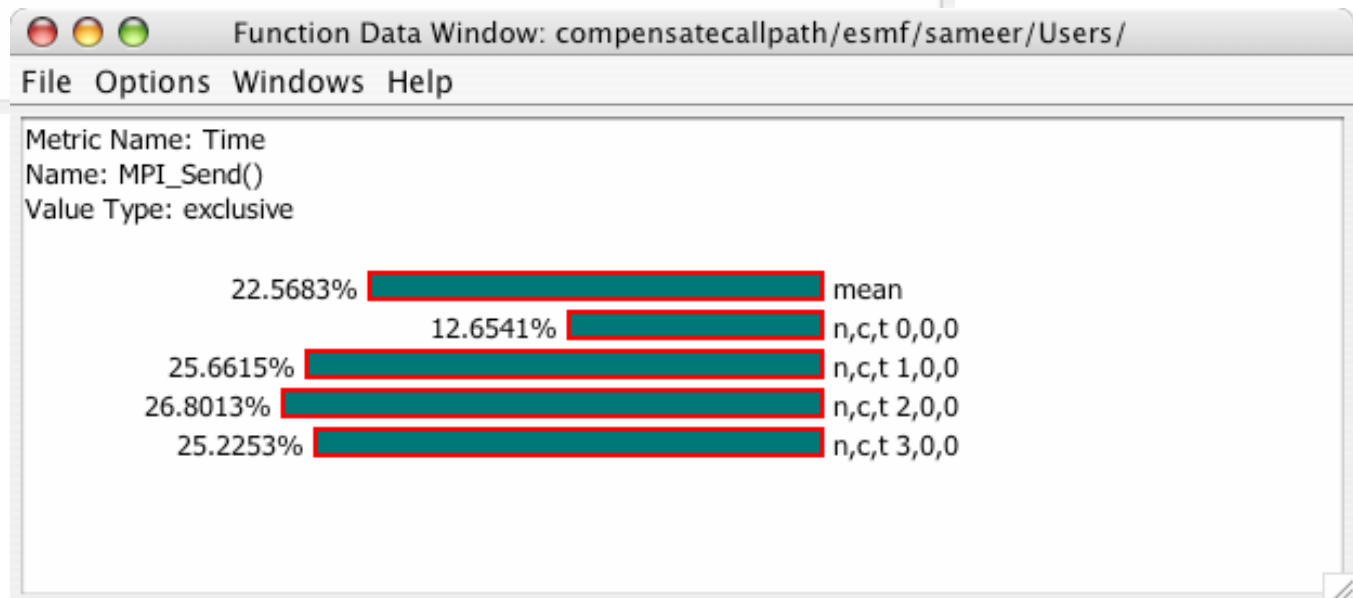
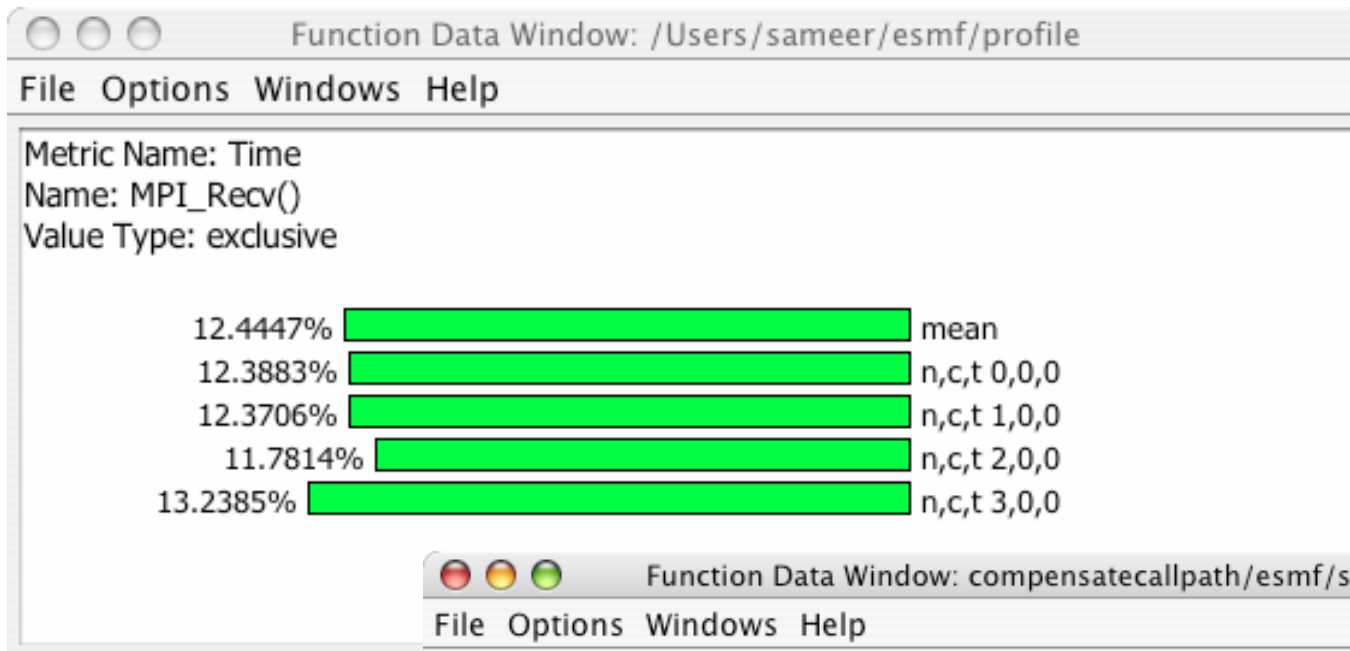




Individual Node (0) Profile in Paraprof



MPI Routines



Text Profile Window



File Options Windows Help

Total n,c,t, 0,0,0 - /Users/sameer/esmf/profile

Metric Name: Time
Sorted By: exclusive
Units: seconds

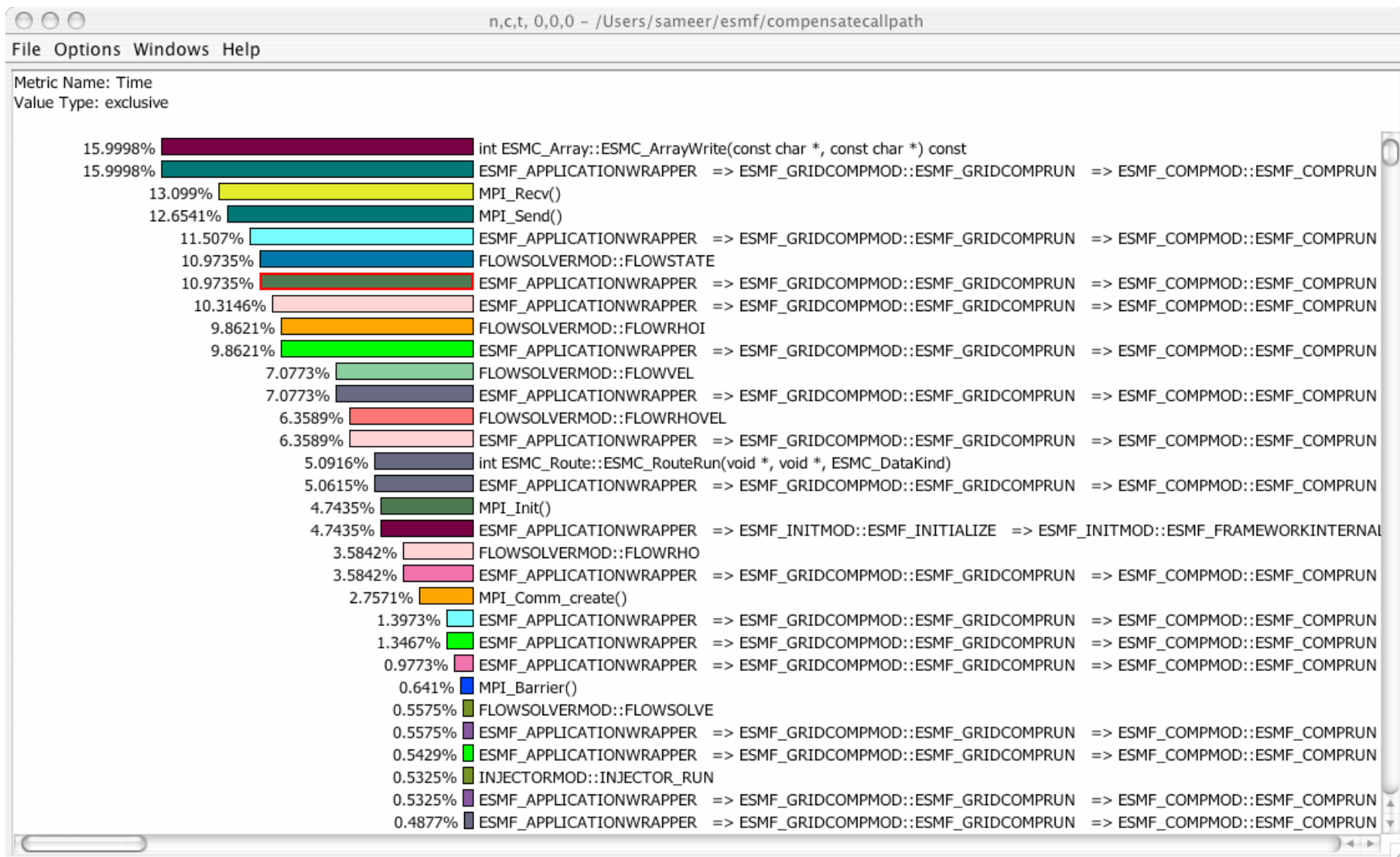
%Time	Time	total Time	#calls	#subrs	total Time/call	name
14.5	6.2006	6.2006	17.0	0.0	0.3647	int ESMC_Array::ESMC_ArrayWrite(const char *, const char *) const
12.4	5.298	5.298	26172.0	0.0	2.0243E-4	MPI_Recv()
12.0	5.1238	5.1238	22500.0	0.0	2.2772E-4	MPI_Send()
12.4	4.6909	5.3147	900.0	1800.0	0.0052	FLWSOLVERMOD::FLOWSTATE
10.8	4.2292	4.5995	900.0	900.0	0.0047	FLWSOLVERMOD::FLOWRHGI
9.9	3.0326	4.2362	900.0	3600.0	0.0034	FLWSOLVERMOD::FLOWVEL
7.8	2.7193	3.3453	900.0	1800.0	0.0030	FLWSOLVERMOD::FLOWRHVEL
30.8	2.41	13.169	22500.0	150300.0	1.0711E-4	int ESMC_Route::ESMC_RouteRun(void *, void *, ESMC_DataKind)
4.8	2.0351	2.0351	1.0	0.0	2.0351	MPI_Init()
5.1	1.5295	2.1724	900.0	1800.0	0.0017	FLWSOLVERMOD::FLOWRHQ
2.8	1.2163	1.2163	3613.0	0.0	3.3664E-4	MPI_Comm_create()
8.1	0.3288	3.4665	9900.0	118802.0	3.321E-5	ESMF_FIELDCOMM::ESMF_FIELDHALODEPRECATED
0.6	0.2656	0.2656	3612.0	0.0	7.352E-5	MPI_Barrier()
61.4	0.2644	26.2607	900.0	18905.0	2.9376E-4	FLWSOLVERMOD::FLWSOLVE
0.8	0.2619	0.3317	900.0	23400.0	2.9095E-4	INJECTORMOD::INJECTOR_RUN
0.7	0.214	0.2787	19808.0	79232.0	1.081E-5	ESMF_LOGRECTGRIDMOD::ESMF_LRGRIDDELOCALTOGLOBALAI

k-Level Callpath Implementation in TAU

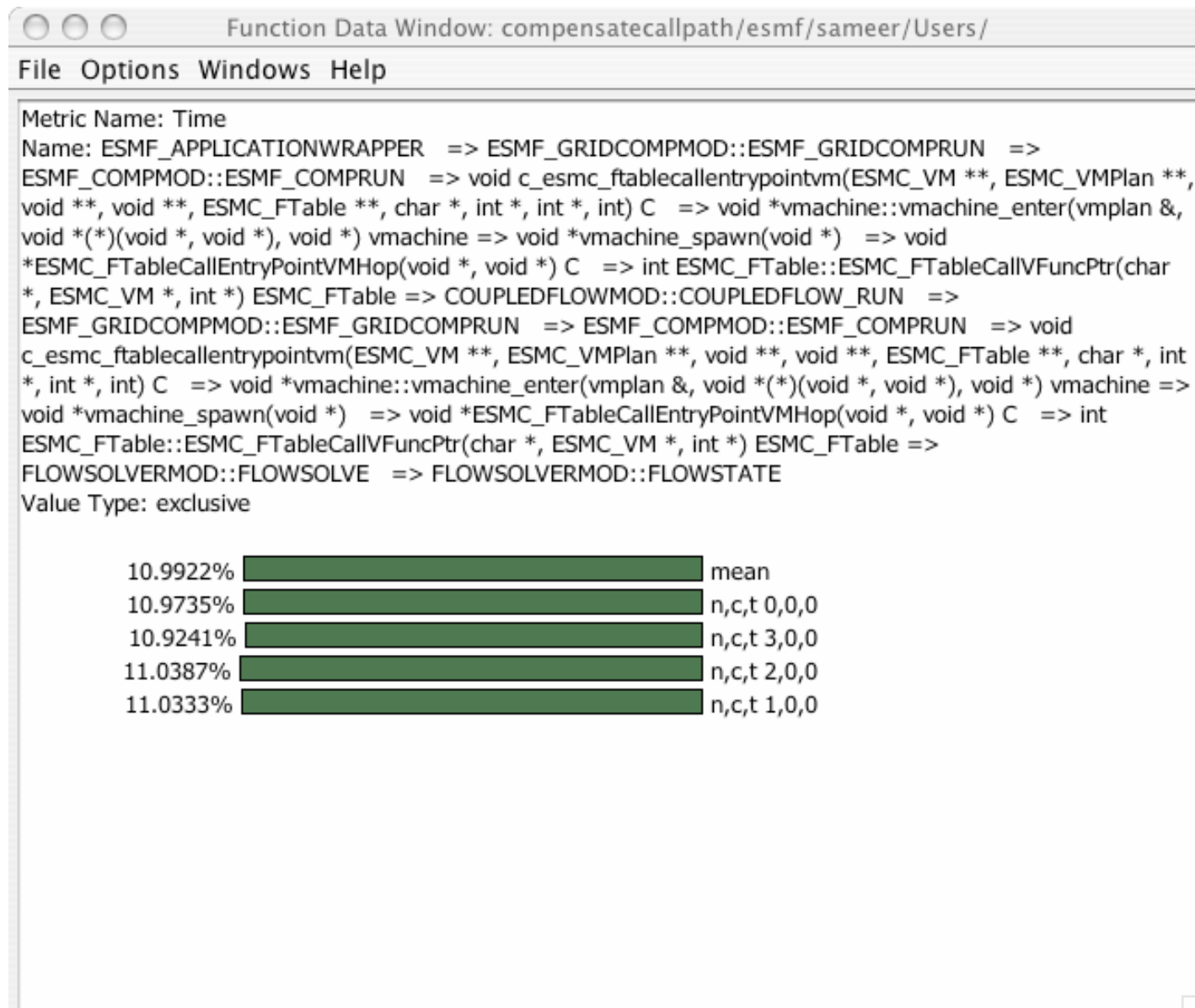


- ❑ TAU maintains a performance event (routine) callstack
- ❑ Profiled routine (child) looks in callstack for parent
 - Previous profiled performance event is the parent
 - A *callpath profile structure* created first time parent calls
 - TAU records parent in a *callgraph map* for child
 - String representing k-level callpath used as its key
 - “a()=>b()=>c()” : name for time spent in “c” when called by “b” when “b” is called by “a”
- ❑ Map returns pointer to callpath profile structure
 - k-level callpath is profiled using this profiling data
 - Set environment variable **TAU_CALLPATH_DEPTH** to depth
- ❑ Build upon TAU’s performance mapping technology
- ❑ Measurement is independent of instrumentation
- ❑ Use **-PROFILECALLPATH** to configure TAU

k-Level Callpath Implementation in TAU



Examining Callpaths



Unique Callpaths



```
Call Path Data Relations - compensatcallpath/esmf/sameer/Users/
File Options Windows Help

Metric Name: Time
Sorted By: exclusive
Units: microseconds

Name[id]
-----

--> ESMF_APPLICATIONWRAPPER [0]
    ESMF_INITMOD::ESMF_INITIALIZE [1]
    ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPCREATEVM [66]
    void esmf_gridcompsetservices(void *, int (*)() C, int *) C [97]
    ESMF_CALENDARMOD::ESMF_CALENDARCREATEBUILTIN [115]
    ESMF_TIMEINTERVALMOD::ESMF_TIMEINTERVALSET [127]
    ESMF_TIMEMOD::ESMF_TIMESSET [152]
    ESMF_CLOCKMOD::ESMF_CLOCKCREATENEW [169]
    ESMF_DELAYOUTMOD::ESMF_DELAYOUTCREATEND [238]
    ESMF_LOGRECTGRIDMOD::ESMF_GRIDCREATEHORZXYUNI [260]
    ESMF_GRIDMOD::ESMF_GRIDDISTRIBUTE [273]
    ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPSET [498]
    ESMF_BASEMOD::ESMF_BASECREATE [70]
    ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPINITIALIZE [507]
    ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPRUN [1281]
    ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPFINALIZE [1932]
    ESMF_BASEMOD::ESMF_BASEDESTROY [1982]
    ESMF_GRIDMOD::ESMF_GRIDDESTROY [2096]
    ESMF_CLOCKMOD::ESMF_CLOCKDESTROY [2132]
    ESMF_CALENDARMOD::ESMF_CALENDARDESTROY [1990]
    ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPDESTROY [2057]
    ESMF_DELAYOUTMOD::ESMF_DELAYOUTDESTROY [2158]
    ESMF_VMMOD::ESMF_VMGET [550]
    ESMF_INITMOD::ESMF_FINALIZE [2173]

    ESMF_APPLICATIONWRAPPER [0]
--> ESMF_INITMOD::ESMF_INITIALIZE [1]
    ESMF_INITMOD::ESMF_FRAMEWORKINTERNALINIT [3]
    ESMF_VMMOD::ESMF_VMGETGLOBAL [63]

    ESMF_INITMOD::ESMF_INITIALIZE [1]
--> ESMF_INITMOD::ESMF_FRAMEWORKINTERNALINIT [3]
    ESMF_CALENDARMOD::ESMF_CALENDARINITIALIZE [5]
    ESMF_LOGERRMOD::ESMF_LOGINITIALIZE [32]
    ESMF_VMMOD::ESMF_VMINITIALIZE [40]
    ESMF_VMMOD::ESMF_VMGETGLOBAL [63]

    ESMF_INITMOD::ESMF_FRAMEWORKINTERNALINIT [3]
--> ESMF_CALENDARMOD::ESMF_CALENDARINITIALIZE [5]
    void c_esmc_calendarinitialize(ESMC_CalendarType *, int *) C [7]
```

Gprof Style Parent, Routine, Children Display



Call Path Data n,c,t, 1,0,0 - compensatcallpath/esmf/sameer/Users/

File Options Windows Help

Metric Name: Time
Sorted By: exclusive
Units: seconds

Exclusive	Inclusive	Calls/Tot.Calls	Name[id]
10.9057	10.9057	33624/33624	void vmachine::vmachine_send(void *, int, int) vmachine[1464]
--> 10.9057	10.9057	33624	MPI_Send() [1466]
5.6126	5.6126	32400/32400	void vmachine::vmachine_recv(void *, int, int) vmachine[1468]
--> 5.6126	5.6126	32400	MPI_Recv() [1470]
4.689	5.1791	900/900	FLowsolvermod::flowsolve [1348]
--> 4.689	5.1791	900	FLowsolvermod::flowstate [1655]
0.0269	0.4901	1800/1800	ESMF_Fieldcommmod::ESMF_Fieldhalodeprecated [1366]
4.2568	4.5577	900/900	FLowsolvermod::flowsolve [1348]
--> 4.2568	4.5577	900	FLowsolvermod::flowrhoi [1472]
0.0162	0.3008	900/900	ESMF_Fieldcommmod::ESMF_Fieldhalodeprecated [1366]
3.0292	3.8199	900/900	FLowsolvermod::flowsolve [1348]
--> 3.0292	3.8199	900	FLowsolvermod::flowvel [1594]
0.0424	0.7906	3600/3600	ESMF_Fieldcommmod::ESMF_Fieldhalodeprecated [1366]
2.7077	3.3302	900/900	FLowsolvermod::flowsolve [1348]
--> 2.7077	3.3302	900	FLowsolvermod::flowrhoel [1364]
0.0311	0.6225	1800/1800	ESMF_Fieldcommmod::ESMF_Fieldhalodeprecated [1366]
4.2375E-4	4.2375E-4	1/3613	void vmachine::vmachine_init() vmachine[46]
2.5014	2.5014	3612/3613	void *vmachine::vmachine_enter(vmplan &, void *(*)(void *, void *), void *)
--> 2.5018	2.5018	3613	MPI_Comm_create() [56]

Clickable Callpath Entities



Call Path Data n,c,t, 1,0,0 - compensatcallpath/esmf/sameer/Users/			
File Options Windows Help			
Metric Name: Time			
Sorted By: exclusive			
Units: seconds			
3.0292	3.8199	900/900	FLWSOLVERMOD::FLWSOLVE [1348]
--> 3.0292	3.8199	900	FLWSOLVERMOD::FLOWVEL [1594]
0.0424	0.7906	3600/3600	ESMF_FIELDCOMM::ESMF_FIELDHALODEPRECATED [1366]
2.7077	3.3302	900/900	FLWSOLVERMOD::FLWSOLVE [1348]
--> 2.7077	3.3302	900	FLWSOLVERMOD::FLOWRHOVEL [1364]
0.0311	0.6225	1800/1800	ESMF_FIELDCOMM::ESMF_FIELDHALODEPRECATED [1366]
4.2375E-4	4.2375E-4	1/3613	void vmachine::vmachine_init() vmachine[46]
2.5014	2.5014	3612/3613	void *vmachine::vmachine_enter(vmplan &, void (*)(void *, void *), void *) vmach
--> 2.5018	2.5018	3613	MPI_Comm_create() [56]
2.1652	13.3299	22500/22500	void c_esmc_routerunla(ESMC_Route **, ESMC_LocalArray **, ESMC_LocalArray **, int
--> 2.1652	13.3299	22500	int ESMC_Route::ESMC_RouteRun(void *, void *, ESMC_DataKind) [1451]
0.0365	0.0365	22500/12600	int ESMC_CommTable::ESMC_CommTableGetCount(int *) const [1453]
0.0891	0.0891	90000/50400	int ESMC_XPacket::ESMC_XPacketGet(int *, int *, int *, int *, int *) ESMC_XPacket
0.0485	0.0485	45000/25200	int ESMC_DataKindSize(ESMC_DataKind) [1457]
0.0	10.9906	32400/12600	int ESMC_DELayout::ESMC_DELayoutExchange(void **, void **, void **, void **, int,
2.0354	2.0354	1/1	void vmachine::vmachine_init() vmachine[46]
--> 2.0354	2.0354	1	MPI_Init() [48]
1.528	1.9931	900/900	FLWSOLVERMOD::FLWSOLVE [1348]
--> 1.528	1.9931	900	FLWSOLVERMOD::FLOWRHO [1533]
0.0262	0.4652	1800/1800	ESMF_FIELDCOMM::ESMF_FIELDHALODEPRECATED [1366]
0.2606	0.2606	3612/3612	void vmachine::vmachine_barrier() vmachine[529]
--> 0.2606	0.2606	3612	MPI_Barrier() [531]
0.2303	24.7214	900/900	int ESMC_FTable::ESMC_FTableCallVFuncPtr(char *, ESMC_VM *, int *) ESMC_FTable[53
--> 0.2303	24.7214	900	FLWSOLVERMOD::FLWSOLVE [1348]
9.761E-4	0.0020	900/900	ESMF_CLOCKMOD::ESMF_CLOCKGET [1048]
0.0013	0.0072	900/900	ESMF_TIMEINTERVALMOD::ESMF_TIMEINTERVALGET [1055]
2.7077	3.3302	900/900	FLWSOLVERMOD::FLOWRHOVEL [1364]
4.2568	4.5577	900/900	FLWSOLVERMOD::FLOWRHOI [1472]
1.528	1.9931	900/900	FLWSOLVERMOD::FLOWRHO [1533]
3.0292	3.8199	900/900	FLWSOLVERMOD::FLOWVEL [1594]
4.689	5.1791	900/900	FLWSOLVERMOD::FLOWSTATE [1655]
0.0062	0.0062	6300/6300	ESMF_FIELDMOD::ESMF_FIELDVALIDATE [943]
0.0020	0.0064	6300/6300	ESMF_FIELDMOD::ESMF_FIELDGET [945]
5.8664E-4	5.5894	5/5	FLWSOLVERMOD::FLOWPRINT [1828]

Call Path Data n,c,t, 1,0,0 - compensatcallpath/esmf/sameer/Users/

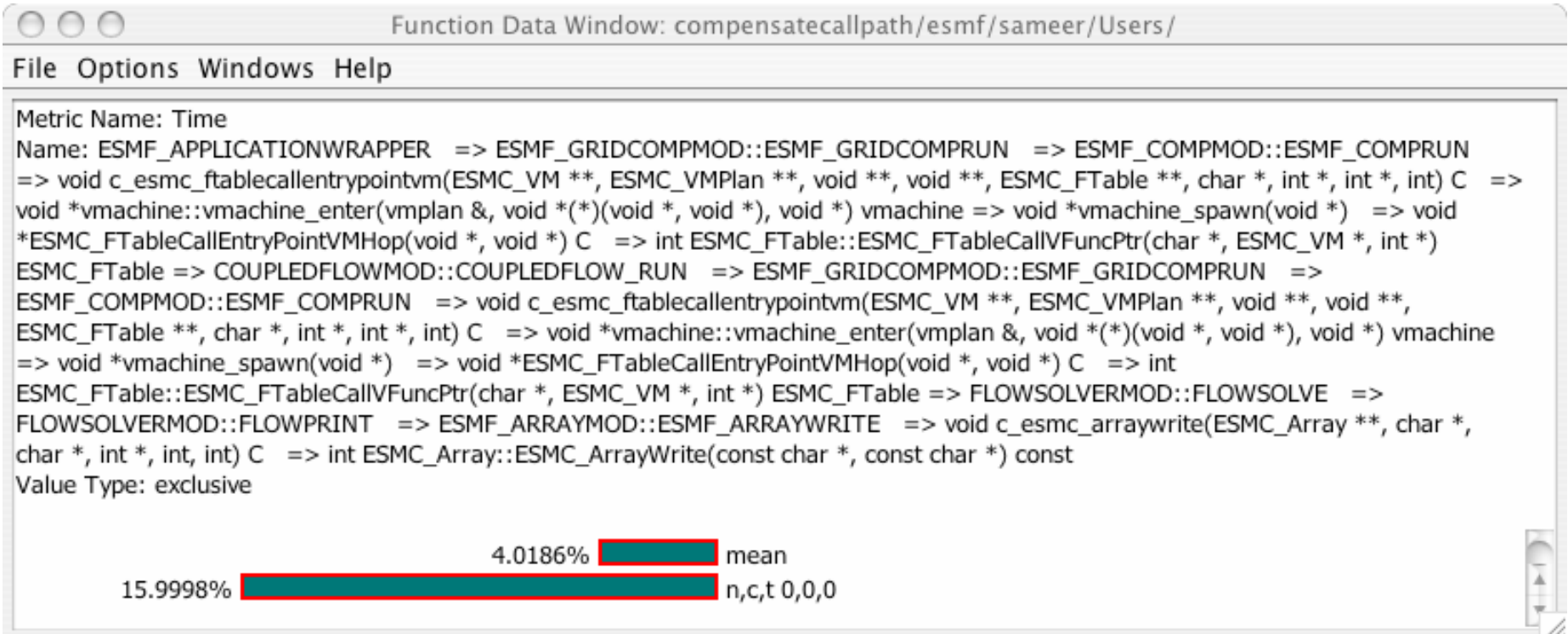
File Options Windows Help

Metric Name: Time
Sorted By: inclusive
Units: seconds

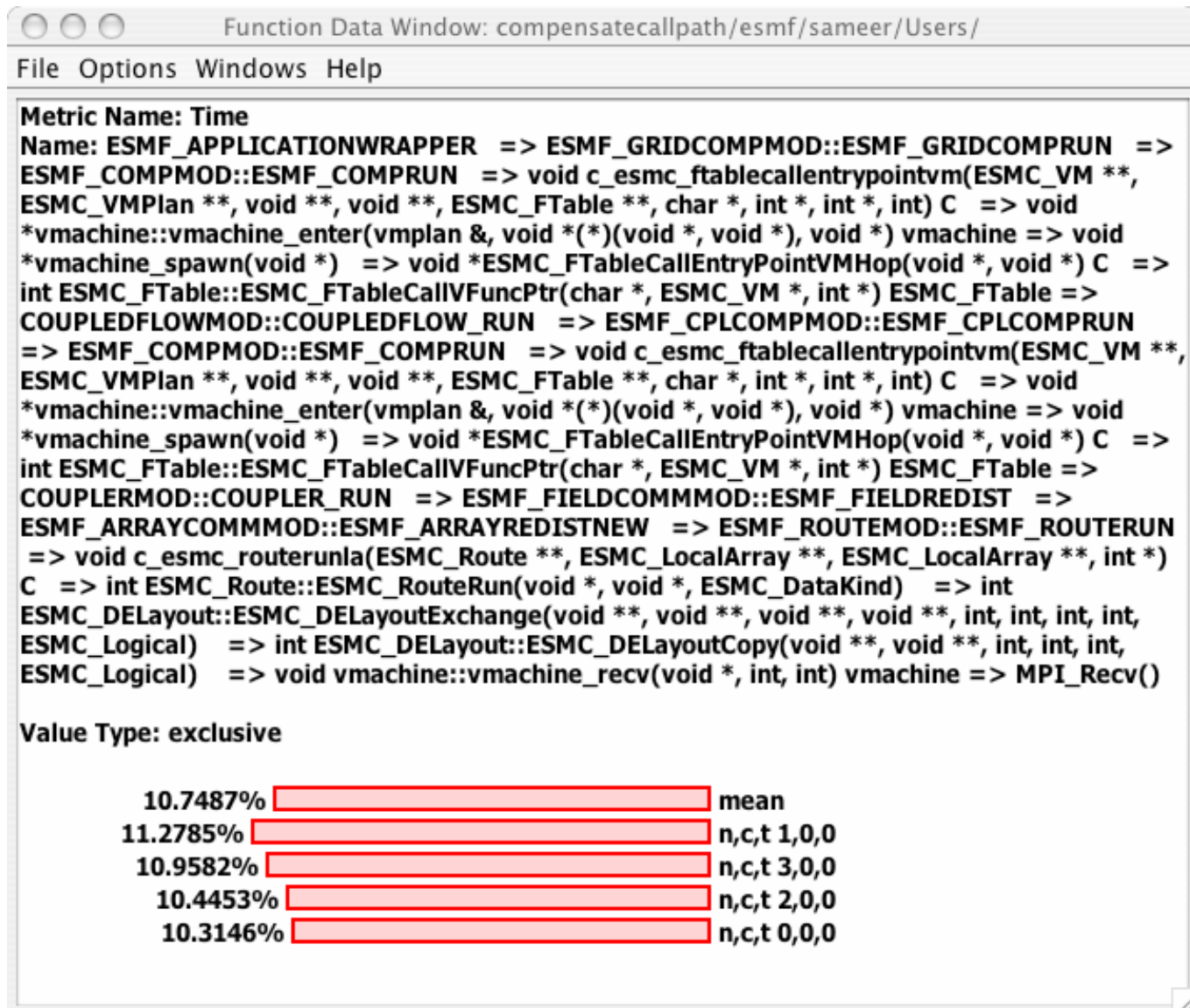
Exclusive	Inclusive	Calls/Tot.Calls	Name[id]
--> 0.0031	42.4985	1	ESMF_APPLICATIONWRAPPER [0]
5.78E-5	2.0391	1/1	ESMF_INITMOD::ESMF_INITIALIZE [1]
5.728E-5	0.0013	1/1	ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPCREATEVM [66]
3.028E-5	5.7762E-4	1/1	void esmf_gridcompsetservices(void *, int (*)() C, int *) C [97]
7.928E-5	5.4623E-4	1/1	ESMF_CALENDARMOD::ESMF_CALENDARCREATEBUILTIN [115]
4.628E-5	0.0011	1/1	ESMF_TIMEINTERVALMOD::ESMF_TIMEINTERVALSET [127]
2.228E-5	7.5134E-4	2/2	ESMF_TIMEMOD::ESMF_TIMESSET [152]
4.928E-5	0.0037	1/1	ESMF_CLOCKMOD::ESMF_CLOCKCREATENEW [169]
4.828E-5	7.8662E-4	1/1	ESMF_DELAYOUTMOD::ESMF_DELAYOUTCREATEND [238]
3.528E-5	5.5623E-4	1/1	ESMF_LOGRECTGRIDMOD::ESMF_GRIDCREATEHORZXYUNI [260]
3.128E-5	0.0378	1/1	ESMF_GRIDMOD::ESMF_GRIDDISTRIBUTE [273]
2.728E-5	6.202E-5	1/1	ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPSET [498]
2.128E-5	2.1613E-4	1/1	ESMF_BASEMOD::ESMF_BASECREATE [70]
3.128E-5	0.2064	1/1	ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPINITIALIZE [507]
3.428E-5	40.1724	1/1	ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPRUN [1281]
2.828E-5	0.0249	1/1	ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPPFINALIZE [1932]
2.628E-5	1.123E-4	1/1	ESMF_BASEMOD::ESMF_BASEDESTROY [1982]
2.98E-5	0.0017	1/1	ESMF_GRIDMOD::ESMF_GRIDDESTROY [2096]
3.628E-5	3.42E-4	1/1	ESMF_CLOCKMOD::ESMF_CLOCKDESTROY [2132]
2.428E-5	1.6658E-4	1/1	ESMF_CALENDARMOD::ESMF_CALENDARDESTROY [1990]
2.428E-5	6.2151E-4	1/1	ESMF_GRIDCOMPMOD::ESMF_GRIDCOMPDESTROY [2057]
2.828E-5	2.9813E-4	1/1	ESMF_DELAYOUTMOD::ESMF_DELAYOUTDESTROY [2158]
2.328E-5	2.1085E-4	1/1	ESMF_VVMOD::ESMF_VMGET [550]
3.433E-5	0.0019	1/1	ESMF_INITMOD::ESMF_FINALIZE [2173]
3.4893E-4	0.312	7/3612	ESMF_COMPMOD::ESMF_COMPINITIALIZE [509]
0.0106	80.2093	3601/3612	ESMF_COMPMOD::ESMF_COMPRUN [1283]
3.4511E-4	0.0395	4/3612	ESMF_COMPMOD::ESMF_COMPPFINALIZE [1934]
--> 0.0113	40.4022	3612	void c_esmc_ftablecallentrypointvm(ESMC_VM **, ESMC_VMPlan **, void **, void **, ESMC_FTable
0.1594	80.5494	3612/1	void *vmachine::vmachine_enter(vmplan &, void *(*)(void *, void *), void *) vmachine[520]
0.1594	80.5494	3612/3612	void c_esmc_ftablecallentrypointvm(ESMC_VM **, ESMC_VMPlan **, void **, void **, ESMC_FTable
--> 0.1608	40.402	3612	void *vmachine::vmachine_enter(vmplan &, void *(*)(void *, void *), void *) vmachine[520]
0.0176	0.0176	3612/1	MPI_Group_incl() [522]
2.5014	2.5014	3612/1	MPI_Comm_create() [56]
8.9999E-4	77.8711	3612/1	void *vmachine_spawn(void *) [525]



Tracking I/O on Node 0 in ESMF



Calling Path for MPI_Recv()





Using TAU with Vampir (Intel Trace Analyzer)



- ❑ Configure TAU with **-TRACE** option

```
% configure -TRACE -mpi ...
```

- ❑ Execute application

```
% poe CoupledFlowApp -procs 4
```

- ❑ This generates TAU traces and event descriptors

- ❑ Merge all traces using **tau_merge**

```
% tau_merge *.trc app.trc
```

- ❑ Convert traces to Vampir Trace format using **tau_convert**

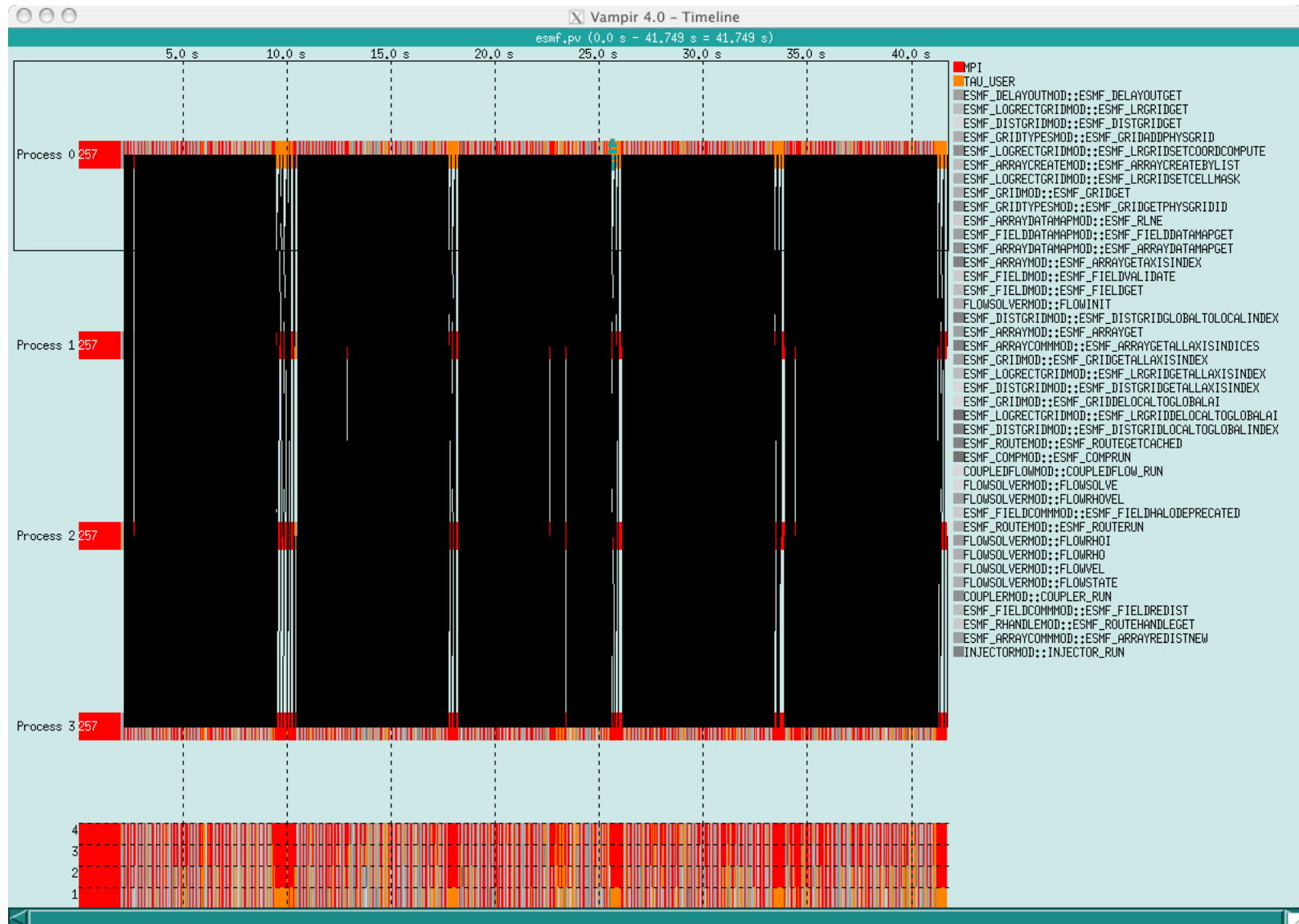
```
% tau_convert -pv app.trc tau.edf app.pv
```

Note: Use **-vampir** instead of **-pv** for multi-threaded traces

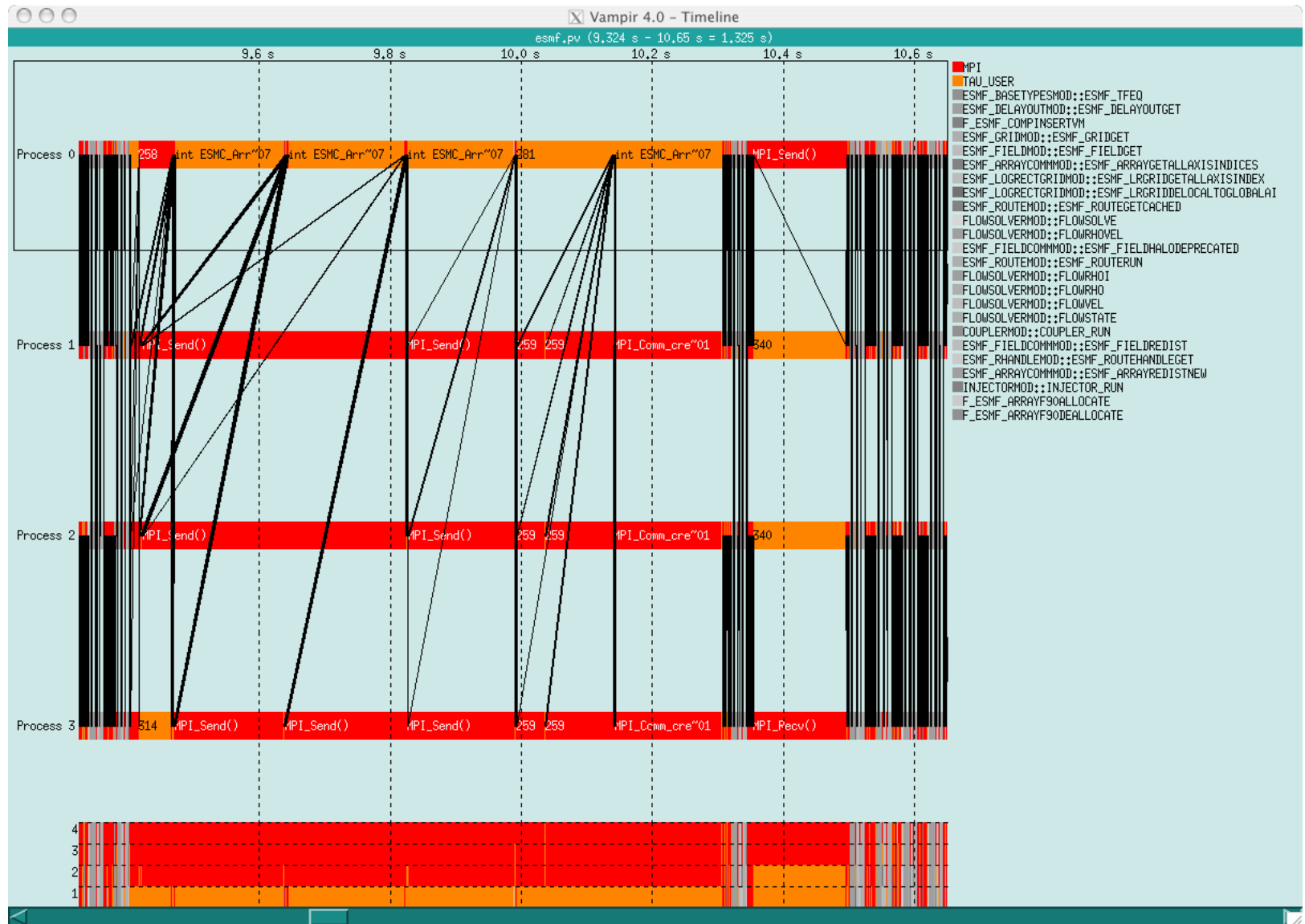
- ❑ Load generated trace file in **Vampir**

```
% vampir app.pv
```

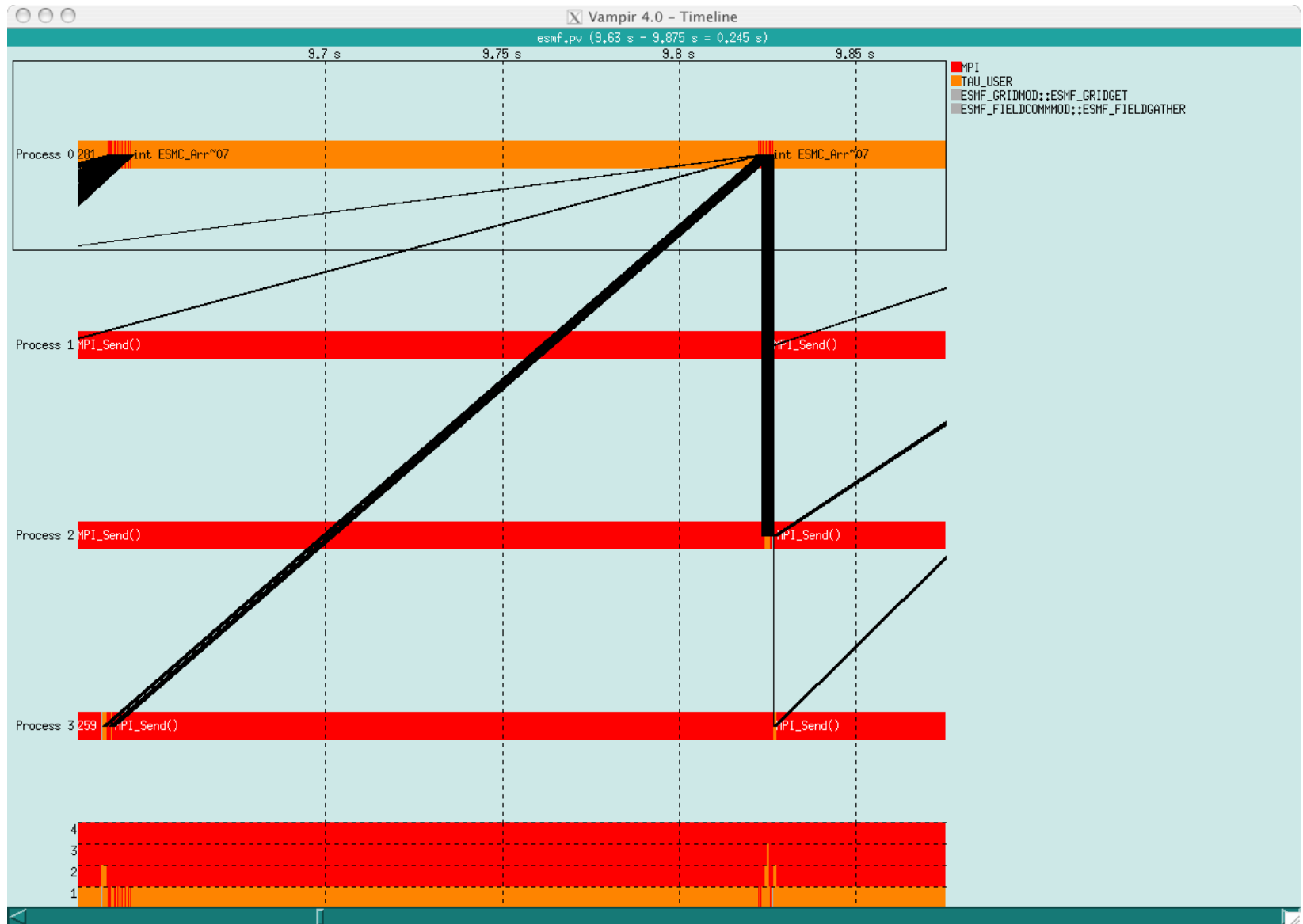
Global Timeline Display with Parallelism View



Vampir: Zooming In...

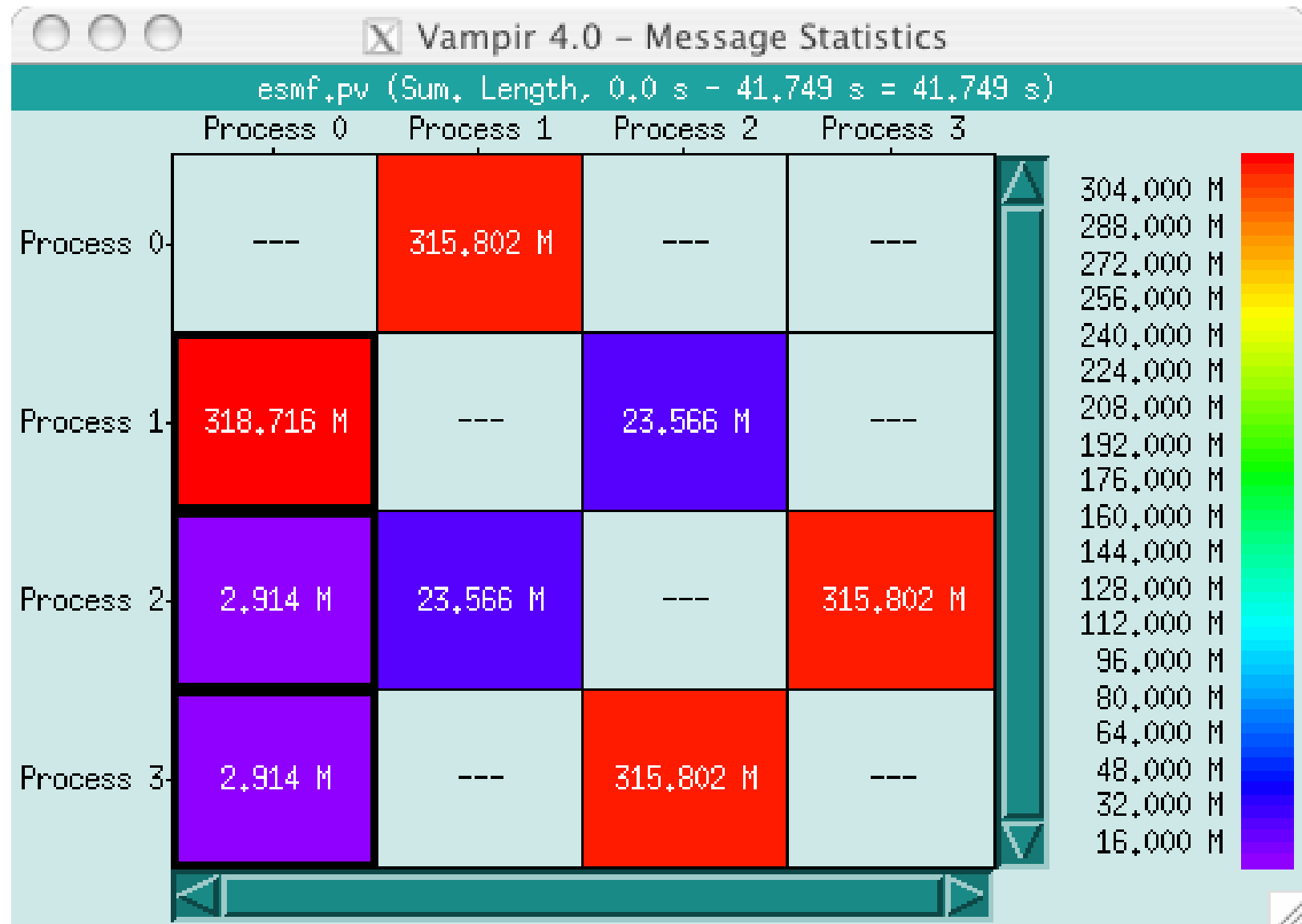


Vampir: IO on Node 0





Vampir: Communication Matrix Display

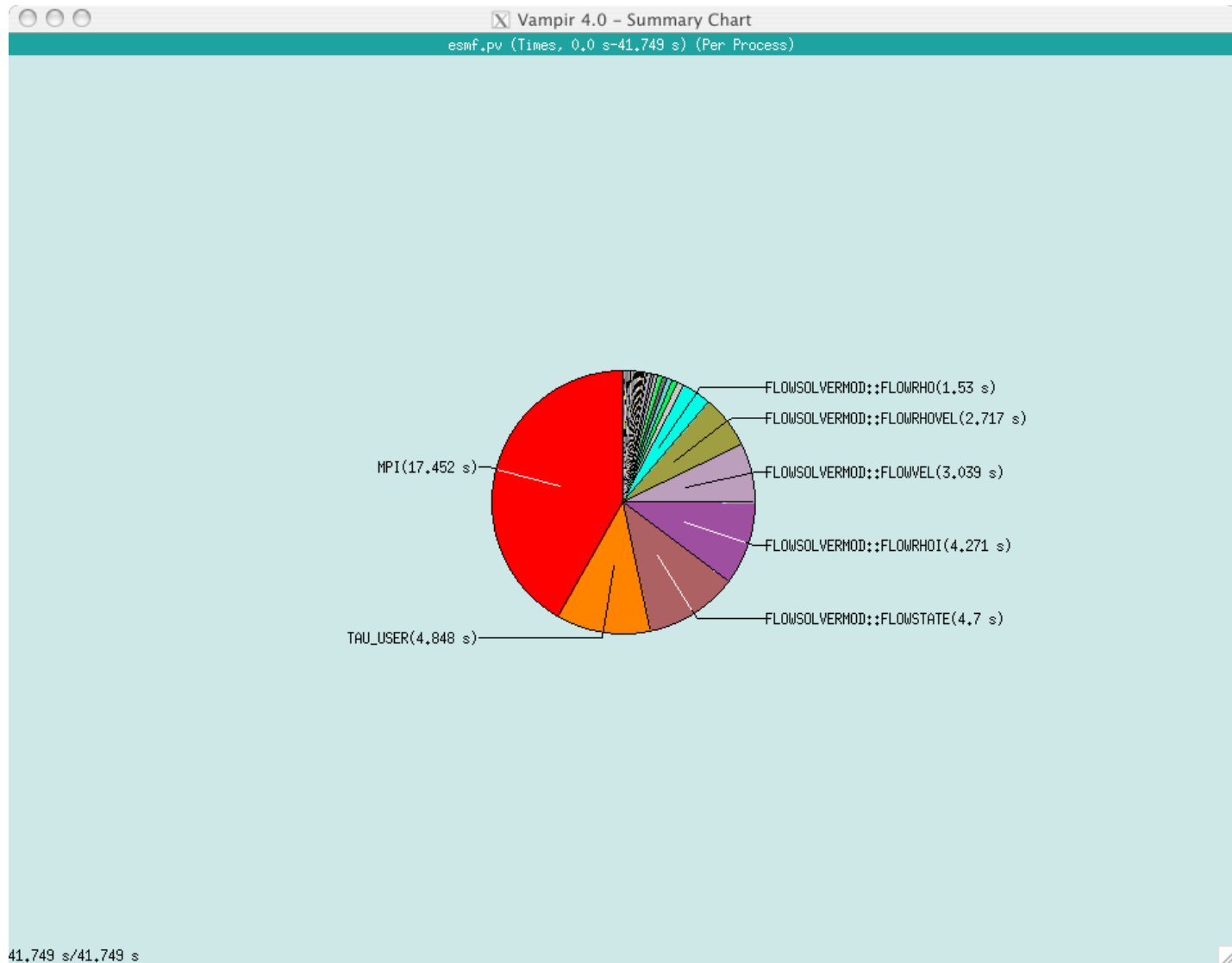


Vampir: Calltree View



```
Vampir 4.0 - Call Tree
esmf.pv
esmf.pv: Global/Times incl./Sorted by:NamesUp
->ESMF_APPLICA~01 (1 : 41.749 s..41.749 s)
  ->ESMF_BASEMOD~01 (1 : 11.0 µs)
    ->void c_esmc~04 (1 : 9.0 µs..10.0 µs)
      ->ESMC_Base &E~01 ... (1 : 2.0 µs..3.0 µs)
      ->char *ESMC_F~01 (2 : 2.0 µs..4.0 µs)
    ->ESMF_BASEMOD~04 (1 : 4.0 µs..5.0 µs)
      ->void c_esmc~49 (1 : 3.0 µs..4.0 µs)
        ->void ESMC_Ba~02 (1 : 1.0 µs)
      ->ESMF_CALENDAR~02 (1 : 74.0 µs..83.0 µs)
        ->void c_esmc~07 (1 : 39.0 µs..40.0 µs)
          ->ESMC_Calendar~03 ... (1 : 12.0 µs..14.0 µs)
        ->ESMF_CALENDAR~03 (1 : 6.0 µs..7.0 µs)
          ->void c_esmc~50 (1 : 4.0 µs..5.0 µs)
            ->int ESMC_Cal~08 ... (1 : 3.0 µs)
      ->ESMF_CLOCKMO~01 (1 : 0.722 ms..0.723 ms)
        ->void c_esmc~10 (1 : 0.687 ms..0.69 ms)
          ->ESMC_Clock *~01 ... (1 : 0.649 ms..0.653 ms)
      ->ESMF_CLOCKMO~06 (1 : 85.0 µs..89.0 µs)
        ->void c_esmc~57 (1 : 65.0 µs..69.0 µs)
          ->int ESMC_Clo~05 ... (1 : 46.0 µs..47.0 µs)
      ->ESMF_DELAYOU~01 (1 : 0.233 ms..0.235 ms)
        ->void c_esmc~11 (1 : 0.197 ms..0.198 ms)
          ->ESMC_DELAYou~01 ... (1 : 0.16 ms..0.162 ms)
          ->bool ESMC_Lo~01 (1 : 0.0 s..1.0 µs)
      ->ESMF_DELAYOU~05 (1 : 75.0 µs..79.0 µs)
        ->void c_esmc~58 (1 : 55.0 µs..58.0 µs)
          ->bool ESMC_Lo~01 (1 : 0.0 s..1.0 µs)
          ->int ESMC_DEL~12 ... (1 : 34.0 µs..35.0 µs)
      ->ESMF_GRIDCOM~01 (1 : 0.457 ms..0.48 ms)
        ->ESMF_COMPMOD~01 (1 : 0.421 ms..0.441 ms)
          ->ESMF_BASEMOD~01 ... (1 : 0.111 ms..0.12 ms)
          ->ESMF_VMOD::~03 ... (1 : 0.153 ms..0.167 ms)
          ->void c_esmc~06 ... (1 : 55.0 µs..58.0 µs)
      ->ESMF_GRIDCOM~02 (1 : 23.0 µs..25.0 µs)
        ->ESMF_COMPMOD~02 (1 : 1.0 µs..2.0 µs)
      ->ESMF_GRIDCOM~03 (1 : 96.182 ms..96.248 ms)
        ->ESMF_COMPMOD~03 (1 : 96.159 ms..96.224 ms)
          ->ESMF_BASEMOD~02 ... (1 : 4.0 µs..5.0 µs)
          ->ESMF_BASETY~04 (1 : 1.0 µs)
          ->void c_esmc~22 ... (1 : 24.0 µs..25.0 µs)
          ->void c_esmc~23 ... (1 : 96.073 ms..96.139 ms)
          ->void c_esmc~26 ... (1 : 12.0 µs..13.0 µs)
      ->ESMF_GRIDCOM~05 (1 : 39.567 s..39.568 s)
        ->ESMF_COMPMOD~05 (1 : 39.567 s..39.568 s)
          ->ESMF_BASEMOD~02 ... (1 : 4.0 µs..5.0 µs)
          ->ESMF_BASETY~04 (1 : 0.0 s..1.0 µs)
          ->void c_esmc~22 ... (1 : 4.0 µs..5.0 µs)
          ->void c_esmc~23 ... (1 : 39.567 s..39.567 s)
          ->void c_esmc~26 ... (1 : 14.0 µs..16.0 µs)
      ->ESMF_GRIDCOM~06 (1 : 3.2 ms..3.335 ms)
        ->ESMF_COMPMOD~06 (1 : 3.177 ms..3.31 ms)
          ->ESMF_BASEMOD~02 ... (1 : 5.0 µs..6.0 µs)
          ->ESMF_BASETY~04 (1 : 1.0 µs)
          ->void c_esmc~22 ... (1 : 6.0 µs..7.0 µs)
          ->void c_esmc~23 ... (1 : 3.139 ms..3.271 ms)
          ->void c_esmc~26 ... (1 : 14.0 µs..15.0 µs)
      ->ESMF_GRIDCOM~07 (1 : 27.0 µs..28.0 µs)
        ->ESMF_COMPMOD~07 (1 : 24.0 µs..25.0 µs)
```

Summary Chart



TAU Performance System Status



□ Computing platforms (selected)

- IBM SP / pSeries, SGI Origin 2K/3K, Cray T3E / SV-1 / X1, HP (Compaq) SC (Tru64), Sun, Hitachi SR8000, NEC SX-5/6, Linux clusters (IA-32/64, Alpha, PPC, PA-RISC, Power, Opteron), Apple (G4/5, OS X), Windows

□ Programming languages

- C, C++, Fortran 77/90/95, HPF, Java, OpenMP, Python

□ Thread libraries

- pthreads, SGI sproc, Java, Windows, OpenMP

□ Compilers (selected)

- Intel KAI (KCC, KAP/Pro), PGI, GNU, Fujitsu, Sun, Microsoft, SGI, Cray, IBM (xlc, xlf), Compaq, NEC, Intel

Concluding Remarks



- ❑ Complex parallel systems and software pose challenging performance analysis problems that require robust methodologies and tools
- ❑ To build more sophisticated performance tools, existing proven performance technology must be utilized
- ❑ Performance tools must be integrated with software and systems models and technology
 - Performance engineered software
 - Function consistently and coherently in software and system environments
- ❑ TAU performance system offers robust performance technology that can be broadly integrated

Support Acknowledgements



❑ Department of Energy (DOE)



- Office of Science contracts
- University of Utah DOE ASCI Level 1 sub-contract
- DOE ASCI Level 3 (LANL, LLNL)



❑ NSF National Young Investigator (NYI) award



❑ Research Centre Juelich

- John von Neumann Institute for Computing
- Dr. Bernd Mohr



❑ Los Alamos National Laboratory



UNIVERSITY
OF OREGON