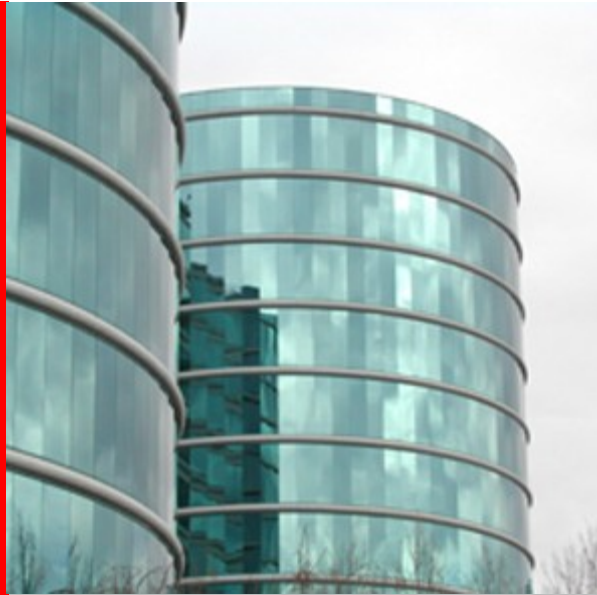


ORACLE®



ORACLE[®]

The Other HPC: Profiling Enterprise-scale Applications

Marty Itzkowitz
Senior Principal SW Engineer, Oracle
`marty.itzkowitz@oracle.com`



Agenda

- HPC Applications
 - Traditional HPC
 - The Other HPC
- Profiling Enterprise-Class Applications
 - SPECjbb, SPECjAppserver, SPECjEnterprise
 - SOA
 - Oracle Database

Traditional HPC

- Intensive numerical calculations
 - Fortran/C/C++
 - OpenMP/MPI
- Run on many CPUs, nodes
 - Many threads (OpenMP)
 - Many processes (MPI)
 - Hybrid runs
- Multiple processes tend to be uniform
- Computations are mostly loop-based

The Other HPC

- Transactions and web services
 - Java/C/C++
 - Ad hoc parallelism
- Also run on many CPUs, nodes
 - Long duration — web servers run forever
 - Many threads
 - Many processes
 - But not quite peta-scale (yet)
- Multiple processes are not uniform
- Often not loop-based

Profiling Enterprise-Class Applications

- Many processes, many threads; long duration
 - Need to track all
 - Typically have long initialization phase
- Multi-thread performance issues
 - Lock contention: lock-global vs. lock-local
 - Synchronization tracing (use `collect -s on`)
 - Key issue: scoping of locks
 - Load imbalance
 - Useful work matters, not CPU usage
 - Busy-waits use CPU resources, but are not useful work

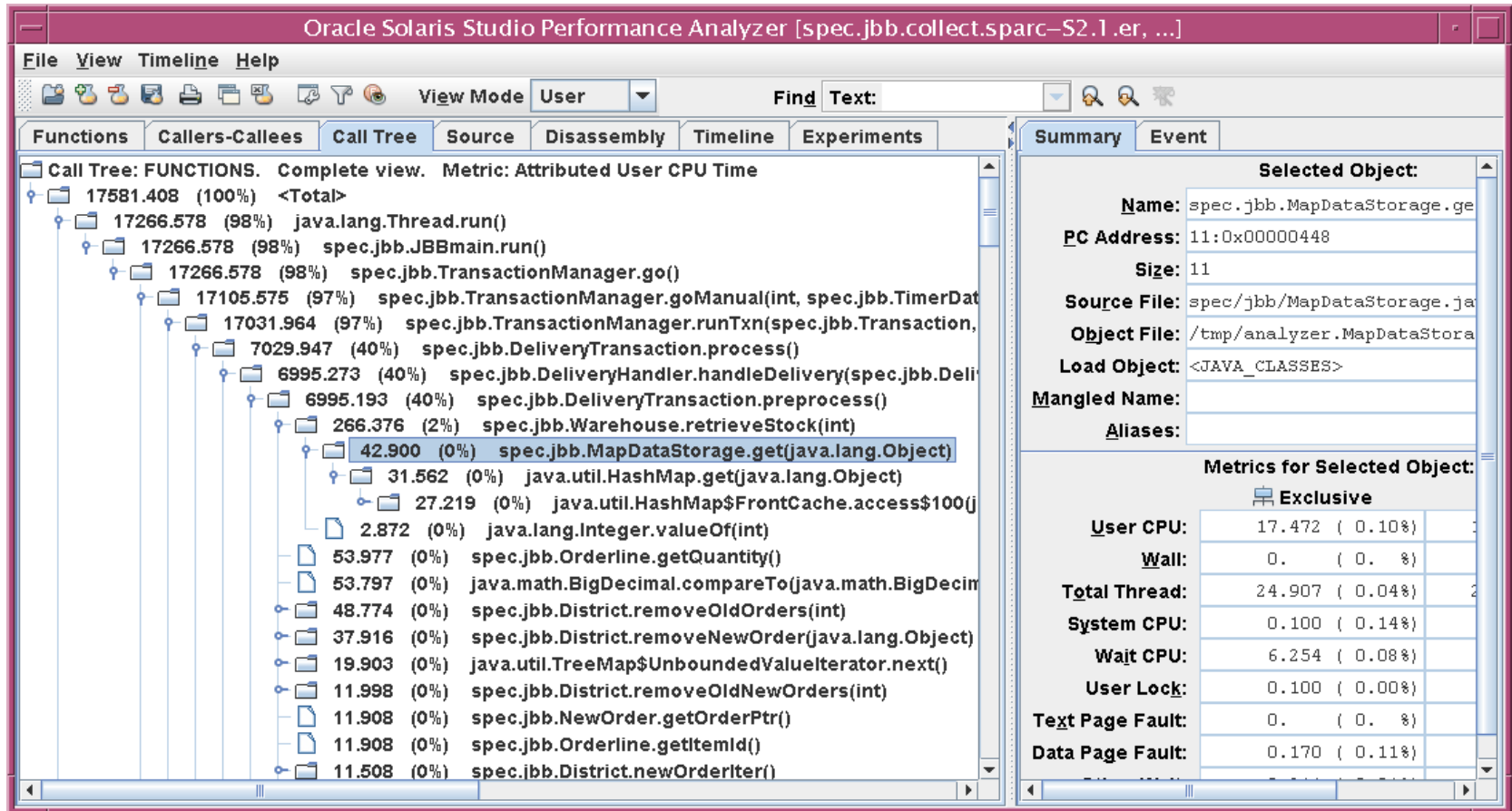
Profiling Enterprise-Class Applications (continued)

- Complex start up: launch by script
 - Add env.var. to prepend `collect` command to target invocation
 - No effect if not set; data collection if set
 - `-y` argument for data-collection control (e.g., skip initialization)
 - `-l` argument for event marking (e.g., mark transaction begin/end)
 - API calls in user code can be used to for markers, too
 - Calls ignored if no data being collected
- Filtering to drill down on problems
 - Based on function on stack
 - Based on threads, processes, CPUs
 - Between marked events

SpecJBB

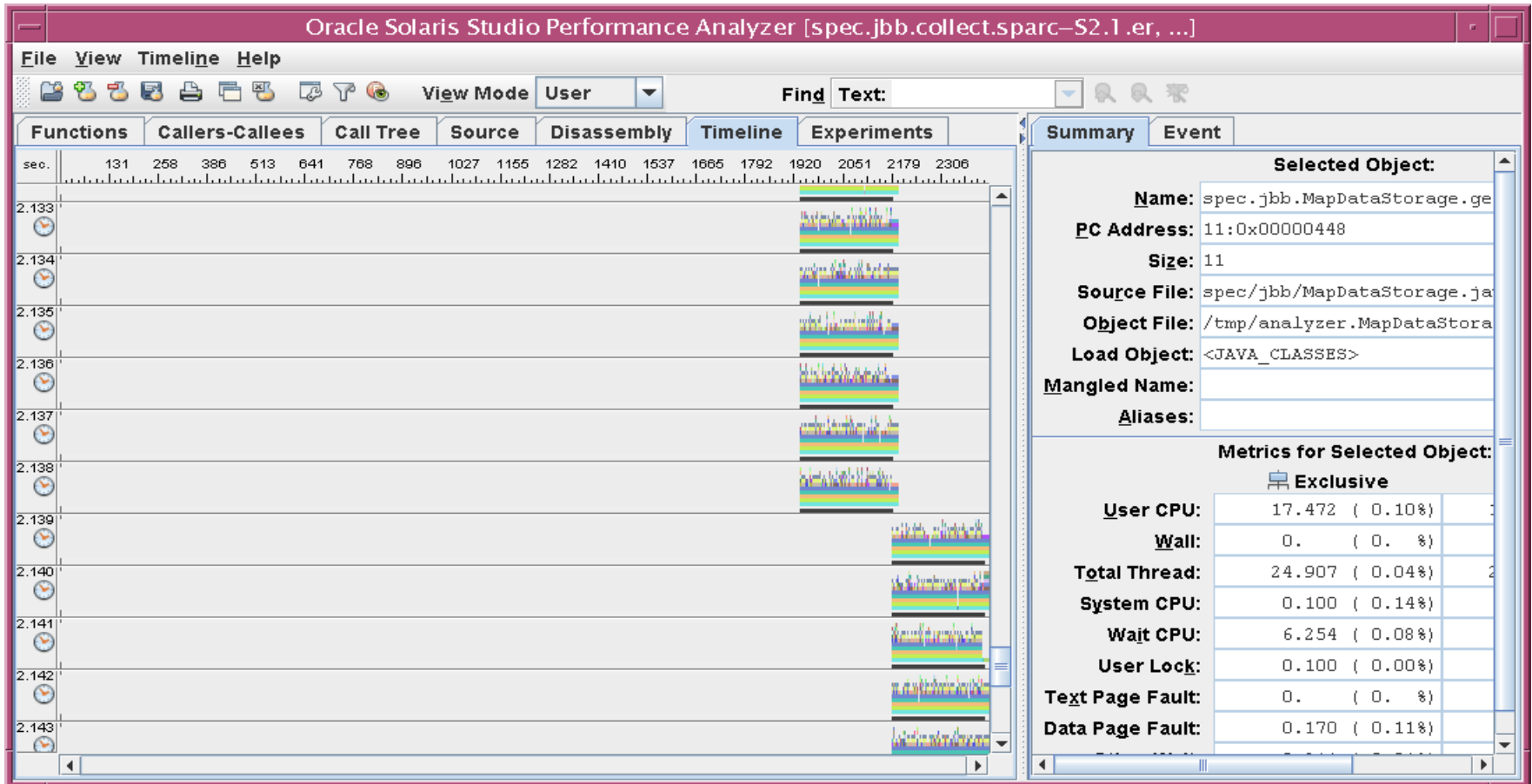
- Benchmark for three-tier enterprise system
 - Based on TPC-C
 - A small enterprise-scale application
- Models a wholesale company and order-entry system
 - Has warehouses that serve districts
 - Run does first 1, then 2, ..., 16 warehouses
 - Up to twice the number of CPUs detected
 - First eight ignored, last eight count for score
 - Processes orders, deliveries, payments, *etc.*
 - Has no real database interactions
 - Data records stored as HashMaps or TreeMap
- Run on 8-CPU machine, uses 156 threads
 - New set of $2N$ threads created for warehouse N
 - Completely CPU-bound

SpecJBB: Call Tree



Shows hottest path

SpecJBB: Timeline

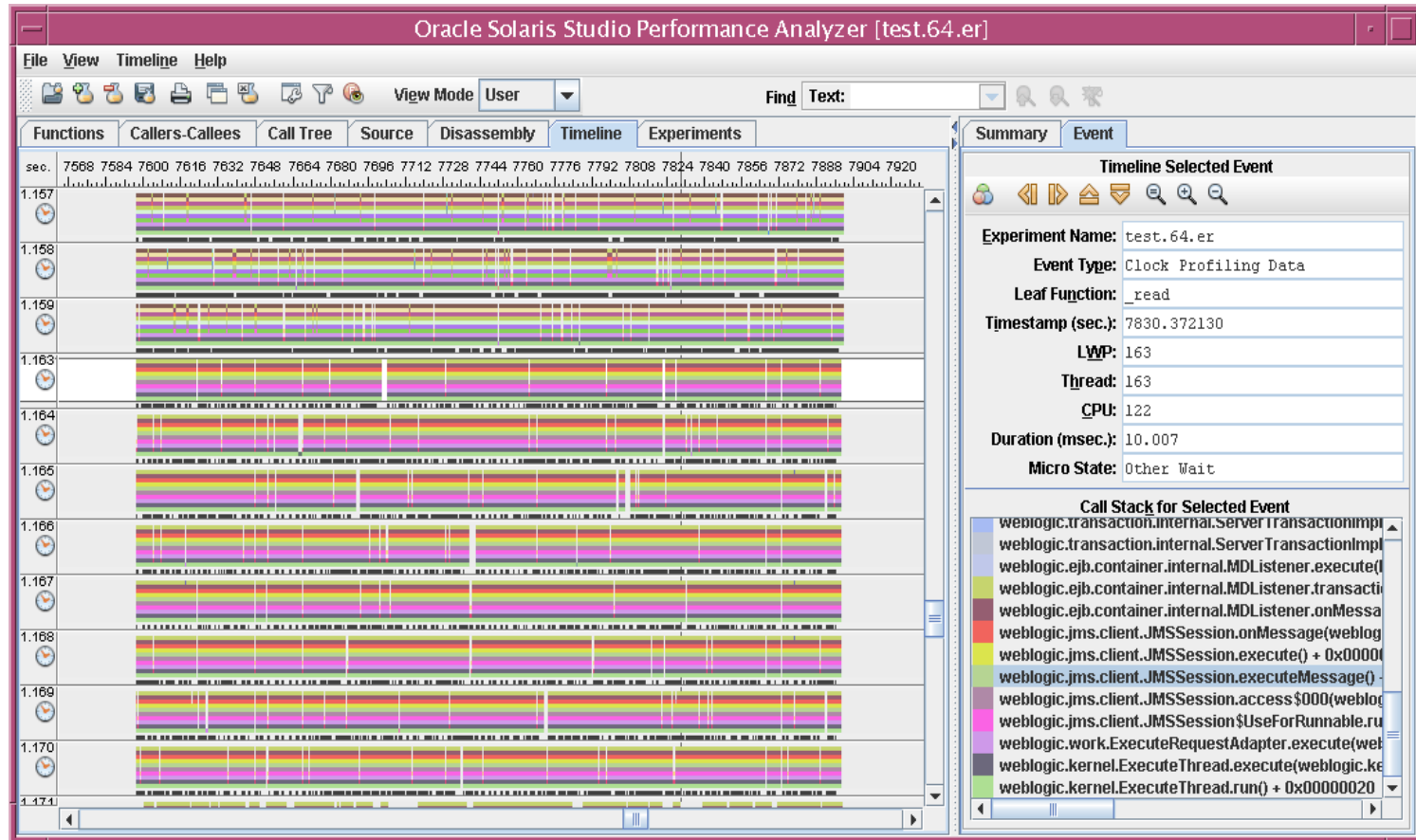


Transition from 15 warehouses to 16
Old threads terminate; new threads are created

SpecJAppServer

- Profile of WebLogic Application Server
 - Simulates standard e-commerce application
 - Processes requests from clients via browser for purchases
 - Processes requests via CORBA/IIOP to manage inventory
- Run on 128-CPU machine, uses ~280 threads
- Data collection paused during initialization phase
 - Recorded data shows active window ~400 seconds

SpecJAppServer: Timeline



Time from ~7500 – 7900 seconds
Threads 157-170; two different types of threads shown

SpecJAppServer: Function List

Oracle Solaris Studio Performance Analyzer [test.64.er]

File View Timeline Help

View Mode User Find Text:

Functions Callers-Callees Call Tree Source Disassembly Timeline Experiments

User CPU (sec.)	User CPU (sec.)	Sys. CPU (sec.)	Sys. CPU (sec.)	Name
4 886.288	4 886.288	1 925.457	1 925.457	<Total>
6.234	4 379.373	0.300	1 897.047	weblogic.kernel.ExecuteThread.run()
0.150	4 354.436	0.010	1 865.885	weblogic.kernel.ExecuteThread.execute(weblogic.kernel.Exe
0.530	4 295.785	0.050	1 689.292	weblogic.work.ExecuteRequestAdapter.execute(weblogic.kern
3.352	3 254.957	0.230	1 310.807	weblogic.servlet.internal.ServletRequestImpl.run()
1.461	3 177.653	0.050	1 281.997	weblogic.servlet.internal.WebAppServletContext.execute(we
2.121	3 169.797	0.140	1 281.516	weblogic.servlet.internal.WebAppServletContext.securedExe
0.791	3 156.558	0.040	1 280.586	weblogic.security.service.SecurityManager.runAs(weblogic.
7.615	3 155.888	0.470	1 280.566	weblogic.security.acl.internal.AuthenticatedSubject.doAs(
1.301	3 085.949	0.030	1 276.103	weblogic.servlet.internal.WebAppServletContext\$ServletInv
1.221	3 078.233	0.080	1 275.742	weblogic.servlet.internal.ServletStubImpl.execute(javax.s
11.308	3 077.803	0.610	1 275.702	weblogic.servlet.internal.ServletStubImpl.execute(javax.s
0.250	3 072.659	0.020	1 275.372	weblogic.servlet.internal.StubSecurityHelper.invokeServle
0.931	3 072.519	0.060	1 275.362	weblogic.servlet.internal.StubSecurityHelper\$ServletServi
1.051	3 072.109	0.040	1 275.332	javax.servlet.http.HttpServlet.service(javax.servlet.Serv
1.321	3 071.178	0.040	1 275.292	javax.servlet.http.HttpServlet.service(javax.servlet.http
0.030	3 069.817	0.	1 275.252	org.spec.jappserver.servlet.SpecAppServlet.doGet(javax.se
9.987	3 069.817	0.520	1 275.252	org.spec.jappserver.servlet.SpecAppServlet.performTask(ja
8.176	151.886	0.741	965.585	java.net.SocketOutputStream.socketWrite(byte[], int, int)
9.467	142.250	0.831	964.745	java.net.SocketOutputStream.socketWrite0(java.io.FileDesc
5.174	132.783	0.420	963.914	Java_java_net_SocketOutputStream_socketWrite0
5.084	72.070	0.280	960.722	JVM_Send

Summary Event

Selected Object:

Name: weblogic.security.service.SecurityM

PC Address: 2:0x00000000

Size: 4294967295

Source File: (unknown)

Object File: weblogic.security.service.SecurityM

Load Object: <JAVA_CLASSES>

Mangled Name: weblogic.security.service.SecurityM

Aliases:

Metrics for Selected Object:

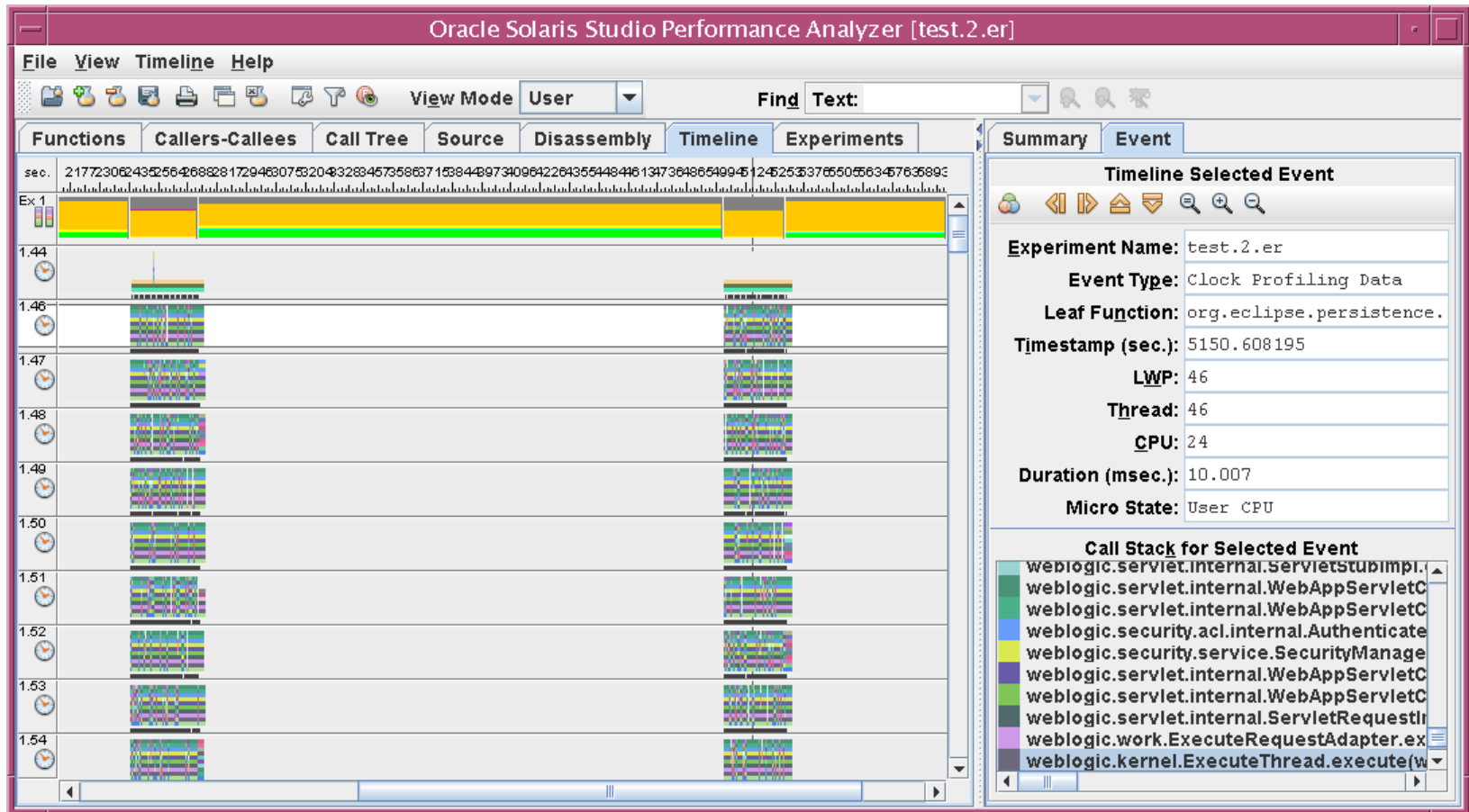
	Exclusive	Inclusi
User CPU:	0.030 (0.00%)	841.979 (
Wall:	0. (0. %)	0. (
Total Thread:	0.030 (0.00%)	9215.116 (
System CPU:	0. (0. %)	290.793 (
Wait CPU:	0. (0. %)	206.504 (
User Lock:	0. (0. %)	803.892 (
Text Page Fault:	0. (0. %)	0. (
Data Page Fault:	0. (0. %)	0.020 (
Other Wait:	0. (0. %)	7071.927 (

Sorted by system CPU time – implies I/O activity

SpecJEnterprise

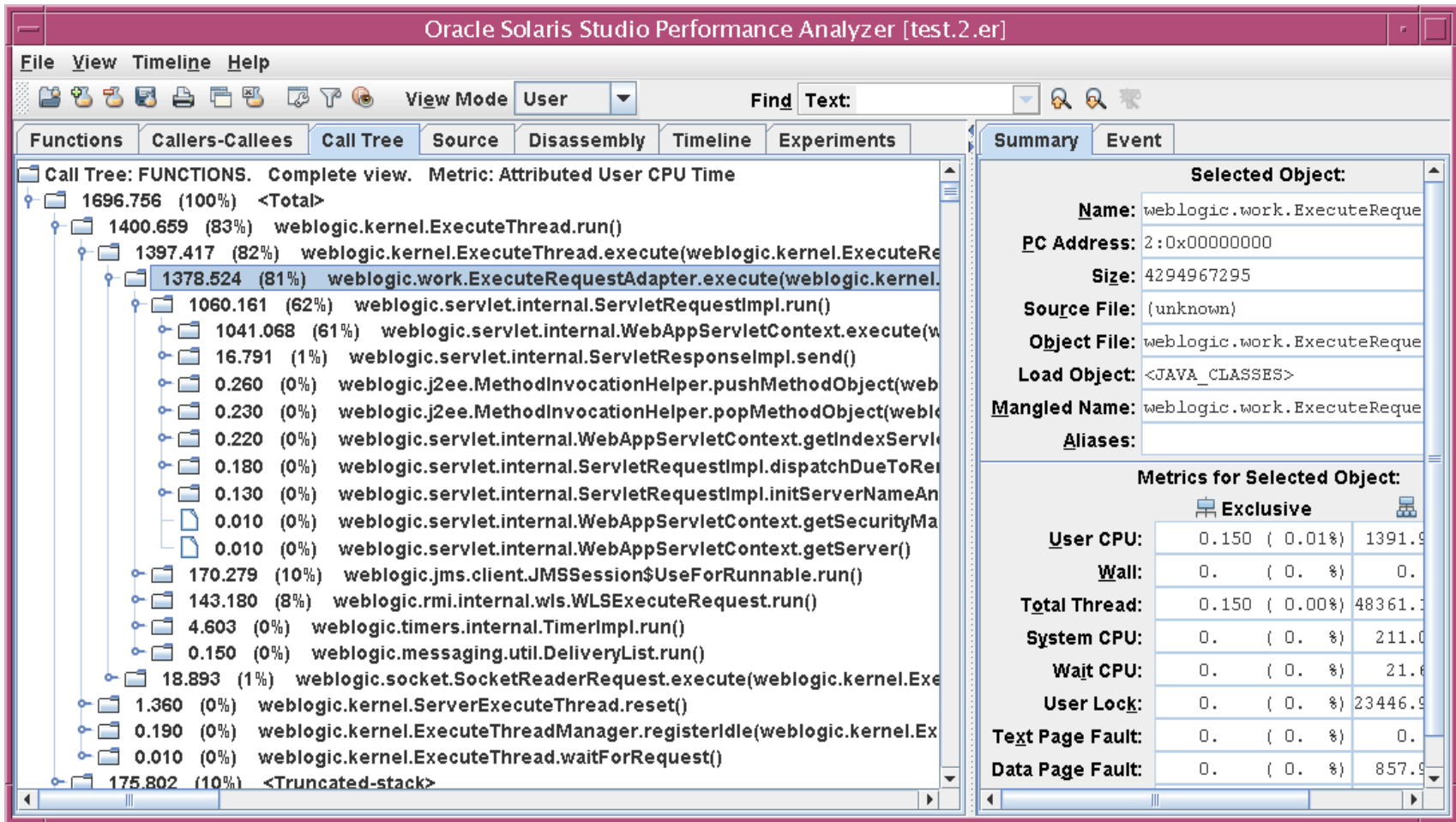
- Benchmark emulates automobile manufacturer
 - Stresses Java EE 5 servers, JVM, CPU, *etc.*
 - Three domains: Dealer, Manufacturing and Supplier
 - Driver drives the benchmark
 - Runs on different system
 - Successor benchmark to SPECjAppserver
- Run on 128-CPU machine, uses 282 threads
- Data collection enabled for two 300 second snaps
 - First at 2436 seconds, second at 5026 seconds
 - Data covers only those two intervals

SpecJEnterprise: Timeline



Data was collected only for two intervals

SpecJEnterprise: Call Tree



Most time spent in WebLogic middleware

Oracle SOA Suite

- SOA = Service-Oriented Architecture
- Single service component architecture
 - Based on Fusion Middleware and WebLogic
 - High throughput, low latency
 - Unified event-driven and service-oriented capabilities
 - Handles complex events
- Near real-time performance requirement
- Run on 64-CPU machine, using 166 threads
 - One run, collected clock- and cache-miss-profiles

SOA: Functions

Oracle Solaris Studio Performance Analyzer [test.3.er, ...]

File View Timeline Help

View Mode Expert Find Text:

Functions Callers-Callees Call Tree Source Disassembly DataObjects DataLayout Timeline Experiments

User CPU (sec.)	User CPU (sec.)	DC_miss Events	DC_miss Events	L2_dmiss Events	L2_dmiss Events	Name
1 170.749	1 170.749	9 941 029 823	9 941 029 823	891 002 673	891 002 673	<Total>
0.	536.425	0	5 301 015 903	0	428 001 284	_lwp_start
0.	533.363	0	5 283 015 849	0	428 001 284	java_start
0.	463.754	0	4 984 014 952	0	378 001 134	JavaThread::thread_main_inner()
0.	458.731	0	4 915 014 745	1 000 003	374 001 122	CompileBroker::compiler_thread_loop()
0.020	458.701	0	4 915 014 745	0	373 001 119	CompileBroker::invoke_compiler_on_method(CompileTask*)
0.010	458.251	0	4 915 014 745	0	373 001 119	C2Compiler::compile_method(ciEnv*,ciMethod*,int)
0.060	458.170	1 000 003	4 915 014 745	0	373 001 119	Compile::Compile(ciEnv*,C2Compiler*,ciMethod*,int,bool,bool)
0.020	446.122	0	3 352 010 056	0	343 001 029	weblogic.work.ExecuteThread.execute(java.lang.Runnable)
0.030	443.240	2 000 006	3 342 010 026	0	339 001 017	weblogic.work.ExecuteThread.run()
0.	439.557	0	3 272 009 816	0	334 001 002	weblogic.t3.srvr.SubsystemRequest.run()
0.	436.595	0	3 249 009 747	0	330 000 990	weblogic.application.utils.StateMachineDriver.nextState(weblogi
0.	410.677	0	3 085 009 255	0	312 000 936	weblogic.management.deploy.internal.ConfiguredDeployments.trans
0.010	403.883	0	3 047 009 141	0	311 000 933	weblogic.management.deploy.internal.ConfiguredDeployments.deplo
0.010	399.960	0	3 029 009 087	0	309 000 927	weblogic.management.deploy.internal.DeploymentServerService.res
0.	395.977	0	3 000 009 000	0	307 000 921	weblogic.management.deploy.internal.DeploymentServerService.sta
0.	326.899	0	3 302 009 906	0	302 000 906	Compile::Code_Gen()
0.	284.269	0	2 170 006 510	0	209 000 627	weblogic.security.acl.internal.AuthenticatedSubject.doAs(weblog
0.	283.859	0	2 167 006 501	0	209 000 627	weblogic.security.service.SecurityManager.runAs(weblogic.securi
0.	262.043	0	2 138 006 414	0	191 000 573	weblogic.application.internal.BaseDeployment\$2.next(java.lang.0
0.	259.201	0	2 118 006 354	0	191 000 573	weblogic.application.internal.BaseDeployment.activate(weblogic.
0.	255.869	0	2 097 006 291	0	189 000 567	weblogic.application.internal.DeploymentStateChecker.activate(w

Two main paths: HotSpot compiler and weblogic
(Inferred from function names)

SOA: Filter by Function in Stack

The screenshot displays the Oracle Solaris Studio Performance Analyzer interface. The main window shows a table of functions with columns for User CPU, DC_miss Events, L2_dmiss_Id Events, and Name. The function `weblogic.work.ExecuteThread.execute()` is highlighted. A **Filter Data** dialog box is open, showing the filter clause `((53893) IN XSTACK)` and the specified filter `((53893) IN XSTACK)`.

User CPU (sec.)	User CPU (sec.)	DC_miss Events	DC_miss Events	L2_dmiss_Id Events	L2_dmiss_Id Events	Name
446.122	446.122	3 352 010 056	3 352 010 056	343 001 029	343 001 029	<Total>
0.020	446.122	0	3 352 010 056	0	343 001 029	weblogic.work.ExecuteThread.execute(java.lang.Runnable)
0.	442.600	0	3 332 009 996	0	338 001 014	weblogic.work.ExecuteThread.run()
0.	436.355	0	3 253 009 759	0	332 000 996	weblogic.t3.srvr.SubsystemRequest.run()
0.	389.893	0	2 951 008 853	0	303 000 909	weblogic.management.deploy.internal.ConfiguredDeployments.transit
0.010	389.803	0	2 949 008 847	0	303 000 909	weblogic.management.deploy.internal.DeploymentServerService.resum
0.	389.803	0	2 949 008 847	0	303 000 909	weblogic.management.deploy.inte
0.010	389.793	0	2 949 008 847	0	303 000 909	weblogic.management.deploy.inte
0.	387.891	0	2 934 008 802	0	298 000 894	weblogic.application.utils.Stat
0.	232.743	0	1 830 005 490	0	175 000 525	weblogic.security.acl.internal.
0.	232.743	0	1 830 005 490	0	175 000 525	weblogic.security.service.Secur
0.	223.827	0	1 887 005 661	0	168 000 504	weblogic.application.internal.B
0.	223.827	0	1 887 005 661	0	168 000 504	weblogic.application.internal.B
0.	223.606	0	1 885 005 655	0	169 000 507	weblogic.deploy.internal.target
0.	223.456	0	1 884 005 652	0	168 000 504	weblogic.deploy.internal.target
0.	223.446	0	1 884 005 652	0	168 000 504	weblogic.deploy.internal.target
0.	223.436	0	1 884 005 652	0	168 000 504	weblogic.application.internal.D
0.	222.376	0	1 876 005 628	0	168 000 504	weblogic.management.deploy.inte
0.	222.376	0	1 876 005 628	0	168 000 504	weblogic.management.deploy.inte
0.	222.376	0	1 876 005 628	0	168 000 504	weblogic.management.deploy.inte
0.	222.366	0	1 876 005 628	0	168 000 504	weblogic.management.deploy.inte
0.	208.496	0	1 726 005 178	0	162 000 486	weblogic.application.internal.E

Function list shows data only from events with stacks containing `weblogic.work.ExecuteThread.execute()`

Oracle Database Profile

- Collected during TPC-H power test
- Script launches server, with `-y USR` flag
- Queries launched by a second script
 - Send `SIGUSR` to enable data collection
 - Run one query
 - Send `SIGUSR` to disable data collection
- Experiment has markers for each query
- Run on 128-CPU machine, uses 906 processes
 - Many are ephemeral, with no profile ticks
 - 256 processes do significant work

Oracle Database: Function List

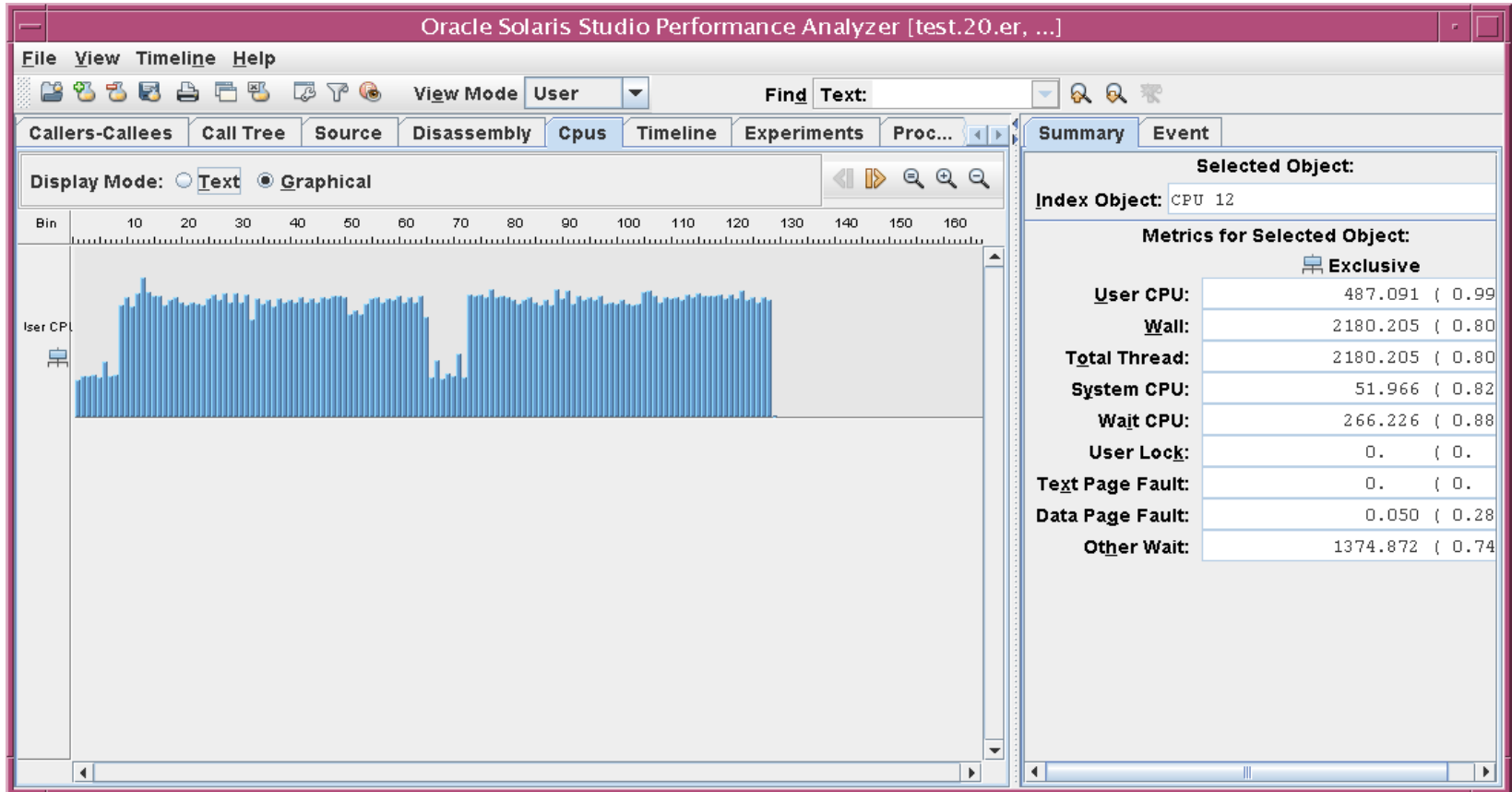
The screenshot displays the Oracle Solaris Studio Performance Analyzer interface. The main window shows a list of functions with columns for User CPU (sec.), User CPU (sec.), and Name. The right-hand pane shows the Summary for the selected object 'sskgs1cas'.

Functions	Callers-Callees	Call Tree	Source	Disassembly	Timeline	Experiments
49 288.428	49 288.428	<Total>				
0.	49 288.428	_start				
0.	49 288.428	main				
0.	49 288.428	opidrv				
0.	49 288.428	opimai_real				
0.	49 288.428	opirip				
0.	49 288.428	sou2o				
0.	49 288.428	ssthrdmain				
0.030	49 288.418	kxfprdp				
0.040	49 239.293	kxfxmai				
0.	49 225.224	kxfxsp				
0.040	49 221.921	upirtc				
0.030	49 221.851	kpoodr				
0.050	49 221.751	opiodr				
0.010	49 221.551	kxfxsExecute				
0.	49 221.541	OCISstmtExecute				
0.120	49 221.521	kpuexec				
0.	49 221.361	kpurcsc				
0.100	49 220.991	kpoal8				

Selected Object:			
Name:	sskgs1cas		
PC Address:	25:0x08F672F0		
Size:	16		
Source File:	(unknown)		
Object File:	est.20.er/archives/oracle.c		
Load Object:	<oracle.orig>		
Mangled Name:			
Aliases:			
Metrics for Selected Object:			
	Exclusive		
User CPU:	25422.994	(51.58%)	254
Wall:	33702.415	(12.40%)	337
Total Thread:	33702.415	(12.40%)	337
System CPU:	103.743	(1.64%)	
Wait CPU:	8145.278	(26.90%)	81
User Lock:	0.	(0. %)	
Text Page Fault:	0.	(0. %)	
Data Page Fault:	11.788	(65.85%)	

~40 minute run

Oracle Database: per-CPU Profile



Sorted by CPU Number

Oracle Database: per-Process Profile

The screenshot displays the Oracle Solaris Studio Performance Analyzer interface. The main window shows a list of processes with columns for CPU usage, Name, and other details. The top 5 processes are highlighted in blue, indicating they are the focus of the filter. A 'Filter Data' dialog box is open in the foreground, showing a filter clause set to '(LEAF IN (147417))' and a 'Specify filter' field containing '(Processes IN {14477, 14403, 14429, 14557, 14497})'. The dialog also includes buttons for 'AND', 'OR', 'Set', 'OK', 'Apply', 'Default', 'Close', and 'Help'.

User CPU (sec.)	Name
49288.428	<Total>
385.400	Processes Index Object 0x000000000000388d (14477)
381.367	Processes Index Object 0x0000000000003843 (14403)
377.464	Processes Index Object 0x000000000000385d (14429)
376.794	Processes Index Object 0x00000000000038dd (14557)
366.687	Processes Index Object 0x00000000000038a1 (14497)
364.585	Processes Index Object 0x000000000000386f (14447)
362.874	Processes Index Object 0x0000000000003823 (14371)
361.433	Processes Index Object 0x0000000000003837 (14391)
360.302	Processes Index Object 0x0000000000003817 (14359)
359.652	Processes Index Object 0x0000000000003863 (14435)
359.501	Processes Index Object 0x0000000000003899 (14489)
359.341	Processes Index Object 0x000000000000387d (14461)
359.121	Processes Index Object 0x00000000000038b3 (14515)
358.451	Processes Index Object 0x00000000000038b7 (14519)
358.361	Processes Index Object 0x000000000000386b (14443)
357.900	Processes Index Object 0x000000000000380f (14351)

Per-process profile; filter set for top 5 processes

Oracle Database: Top Five Processes

The screenshot displays the Oracle Solaris Studio Performance Analyzer interface. The main window shows a list of processes with columns for User CPU and Name. The process 'sskgsllcas' is highlighted, and its details are shown in the right-hand pane.

User CPU (sec.)	User CPU (sec.)	Name
1887.710	1887.710	<Total>
1252.626	1252.626	sskgsllcas
86.370	1399.009	kgxExclusive
28.470	29.130	kaf4ger9ir2srp0
25.548	45.632	kaf4ger9ir2srp1
24.447	24.467	lnxmul
21.345	32.183	kxfqfprFastPackRow
20.684	69.809	qerhjInnerProbeHashTable
20.364	20.464	lnxsum
19.444	170.229	qeshLoadRowForGBY
19.283	19.283	gethrtime
17.292	17.312	kaf4F0rst9ir2srp1
16.391	22.306	qeshrPUCmpare
14.270	14.270	memcpy
13.780	13.780	gettimeofday
13.509	13.509	yield
12.749	90.623	kdstf00101011000km
12.429	91.154	qesaOSMergeWA
11.768	12.489	qeshrGetUHash_Fast

Selected Object:

- Name: sskgsllcas
- PC Address: 25:0x08F672F0
- Size: 16
- Source File: (unknown)
- Object File: est.20.er/archives/oracle.o
- Load Object: <oracle.orig>
- Mangled Name:
- Aliases:

Metrics for Selected Object:

Exclusive		
User CPU:	1252.626 (66.36%)	12
Wall:	1258.610 (23.72%)	12
Total Thread:	1258.610 (23.72%)	12
System CPU:	3.933 (4.60%)	
Wait CPU:	2.001 (1.99%)	
User Lock:	0. (0. %)	
Text Page Fault:	0. (0. %)	
Data Page Fault:	0.020 (40.00%)	

Function list data filtered to show only the top 5 processes

SOFTWARE. HARDWARE. COMPLETE.

ORACLE®