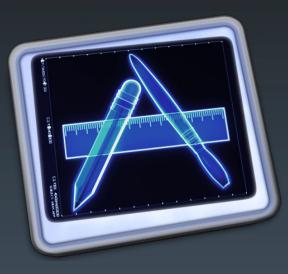
What's New in Instruments

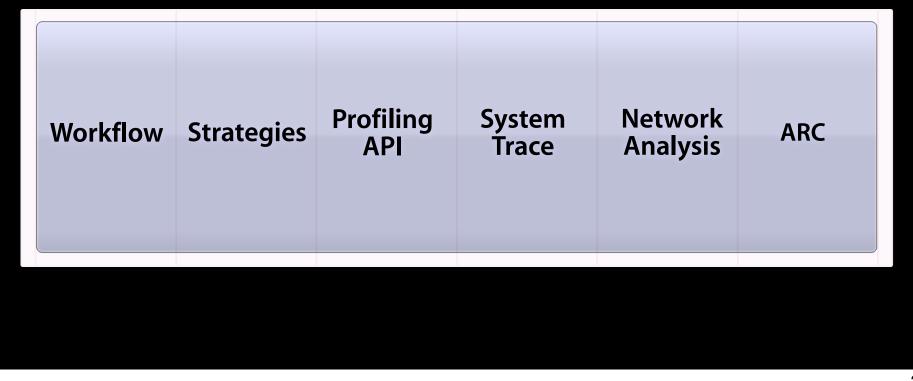


Session 310

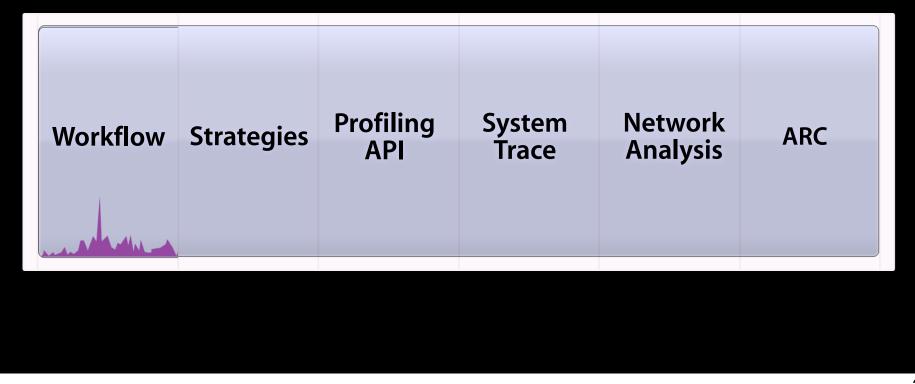
Steve Lewallen Performance Tools Engineering Manager

These are confidential sessions—please refrain from streaming, blogging, or taking pictures





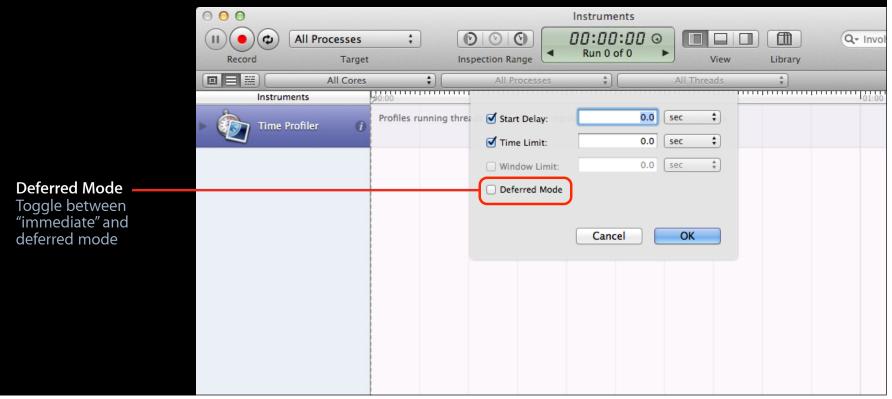
Workflow Improvements From recording to data mining



Record Options

	0 0 0					Instruments			
		ocesses	;	0		00:00:00 (Run 0 of 0			Q- Invol
	Record	Target		Inspection	Range	Kunooro	View	Library	
		All Cores	\$		ll Processes	Å (All Threads	* *	
	Instruments		20:00						01:00
Start Delay		0	Profiles running	three 🗹 s	Start Delay:	oula 0.0	sec 🗘		
Add delay between	Time Profiler	()				0.0	sec 🛟		
nitting record and					Time Limit:	0.0			
record starting				□ v	Nindow Limit:	. 0.0	sec 🛊		
					Deferred Mode	e			
						Cancel	ОК		

0 0 0				Instruments			
(I) (C) (C) (C) (C) (C) (C) (C) (C) (C) (All Processes Target		DOC	CO:CO:CO @ Run 0 of 0	View	Library	Q- Invol
	All Cores	\$	All Processes	÷)	All Threads	÷	
Instrume		20:00					01:00
Time Pr	ofiler 👔	Profiles running thre	a 🔄 🗹 Start Delay:	0.0	sec 🗧		
Time Limit			Time Limit:	0.0	sec 🗘		
Set a maximum			Window Limit:	0.0	sec 🗍		
recording length			Deferred Mode				
				Cancel	ОК		



	O O Instruments Preferences
	General Display Quick Start DTrace Dock Profiling Search Paths
Deferred Mode	Tracing behavior: Monitor all spins
Set deferred mode as the defaults for	Always use deferred mode Startup preferences: Suppress template chooser
all new recordings	Saving: ☑ Save current run only ☑ Compress run data
	Default document location: Default Temporary Directory
	Toggle Front Document Trace: ^り
	Active Processor Cores (8): I Hardware Multi-Threading
	Modifications to the 'Active Processor Cores' preferences will persist until the system is rebooted, or until the preferences are changed again.

General Display Quick Start DTrace Dock Profiling Search Paths Tracing behavior: Monitor all spins Always use deferred mode Startup preferences: Suppress template chooser Saving: Save current run only Compress run data Default document location: Default Temporary Directory Toggle Recording of front-most trace document Instruments may be in background Active Processor Cores (8): Modifications to the 'Active Processor Cores' preferences will persist until the system is rebooted, or until the preferences are changed again.	$\Theta \circ \circ$	Instruments Preferences
 Always use deferred mode Startup preferences: Suppress template chooser Saving: Save current run only Compress run data Default document location: Default Temporary Directory + Toggle Recording of front-most trace document Instruments may be 		General Display Quick Start DTrace Dock Profiling Search Paths
Saving: Save current run only Compress run data Default document location: Default Temporary Directory + Toggle Recording of front-most trace document Instruments may be Modifications to the 'Active Processor Cores' preferences will persist until the		
Instruments may be		Startup preferences: 🗌 Suppress template chooser
Toggle Recording Default document location: Default Temporary Directory + Toggle Recording of front-most trace document Toggle Front Document Trace: >> Instruments may be Modifications to the 'Active Processor Cores' preferences will persist until the		
Toggle Recording Toggle Front Document Trace: Start/Stop recording of Active Processor Cores (8): Instruments may be Modifications to the 'Active Processor Cores' preferences will persist until the		Compress run data
Start/Stop recording of front-most trace document Instruments may be Modifications to the 'Active Processor Cores' preferences will persist until the		Default document location: 📄 Default Temporary Directory 💠
front-most trace document Active Processor Cores (8): I Hardware Multi-Threading Instruments may be Modifications to the 'Active Processor Cores' preferences will persist until the		Toggle Front Document Trace: ^り
Instruments may be Modifications to the 'Active Processor Cores' preferences will persist until the		
		Modifications to the 'Active Processor Cores' preferences will persist until the

$\Theta \bigcirc \bigcirc$	Instruments Preferences
	General Display Quick Start DTrace Dock Profiling Search Paths
	Tracing behavior: Monitor all spins Always use deferred mode
	Startup preferences: 🗌 Suppress template chooser
	Saving: ☑ Save current run only ☑ Compress run data
	Default document location: Default Temporary Directory
	Toggle Front Document Trace: ^り
ggle Hardware Threading —	Active Processor Cores (8): 🗹 Hardware Multi-Threading
rns hyper-threading on or f for supported systems	Modifications to the 'Active Processor Cores' preferences will persist until the system is rebooted, or until the preferences are changed again.

General Display Quick Start DTrace Dock Profil Tracing behavior: Monitor all spins Always use deferred mo Startup preferences: Suppress template choo	
Always use deferred mo Startup preferences: Suppress template choo	de
	ser
Saving: 🗹 Save current run only Saving: 🗹 Compress run data	
Default document location: 🔲 Default Temporary Dir	ectory 🛟
Toggle Front Document Trace: ^り	
Active Processor Cores (8): V Hardware Multi-Thread	ing
Active Processor Cores Active Processor Cores' preferences will processor Cores' preferences will processor Cores' preferences are change and the core of the Active Processor Cores' preferences are change and the core of t	

Track Gestures

Track Gestures

Working with high-resolution traces

000			Instruments			R _M
	ocesses ‡	$\bigcirc \bigcirc \bigcirc \bigcirc$	00:00:00 0			Q- Involves Symbol
Record	Target	Inspection Range	Run 0 of 0 🕨	View	Library	Search
		All Processes		All Threads	*	
Instruments	20:00					01:00
► 🤖 Time Profiler	Profiles rur	ning threads on all cores at re	egular intervals for one or a	Ill processes.		

Track Gestures

Working with high-resolution traces

0 0	an an an Arian an Arian an Arian An Arian an Ari	an an an tao an tao an Tao an tao an		Instruments							R _M
Record Targe	t	1	nspection Range	CO:CO:CO (Run 1 of 1	· · · · · · · · · · · · · · · · · · ·	/iew	Library	Q.	Process	Search	
(III 🔳 🗮) 🛛 🛛 All Cores	;) [All Processes	\$)	All Threa	ds	\$				
Instruments	20:00										
Fime Profiler (i	the canal is denoted			har	(AALWASHI)		ANANAV		la akakan		
🎃 Time Profiler 🗧	E Call Tre	ee 🗢 🔪	Call Tree								≡
Sample Perspective	Running (Self)	Symbol Name								
O All Sample Counts	3484.0ms	51.4%	▶Terminal (774)	D							
 Running Sample Times 	2045.0ms		▶find (29064)								
Call Tree	660.0ms	9.7%	▶WindowServer (10)	06)							
V Call Tree				,							
Separate by Thread	367.0ms	5.4%	▶mach_kernel (0)								
□ Separate by Thread ✓ Invert Call Tree	367.0ms 51.0ms	5.4% 0.7%	▶Instruments (289								
 □ Separate by Thread ☑ Invert Call Tree □ Hide Missing Symbols 	367.0ms 51.0ms 42.0ms	5.4% 0.7% 0.6%	 Instruments (289 hidd (82) 								
 Separate by Thread ✓ Invert Call Tree Hide Missing Symbols Hide System Libraries 	367.0ms 51.0ms 42.0ms 32.0ms	5.4% 0.7% 0.6% 0.4%	 Instruments (289 hidd (82) Preview (28813) 								
 Separate by Thread Invert Call Tree Hide Missing Symbols 	367.0ms 51.0ms 42.0ms	5.4% 0.7% 0.6%	 Instruments (289 hidd (82) 								

Zoom In Shift + Mouse Drag

0 0	n an tha tha Barnata da ann Abhailte 1916 - Bhailte an Lidhean agus an San A Martha Gaile ann a daon a sao ann an 191	Inst	ruments	i an an an Araba an Araba an Araba Anna an Araba an Araba an Araba An Araba an Araba an Araba an Araba	an an Indonesia an Anna Anna A Fan Anna Anna Anna Pantananan Anta Marina an Antana Anna Anna Anna	H.
(II) (All Processes	;	$\bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc$:00:00 0			Q- Process
Record Target	t		in 1 of 1 🔹 🕨	View	Library	Search
		All December 2		All Thursda		
All Cores		All Processes	•) (All Threads	÷	
Instruments	00:00		∇			
Time Profiler	Mar and an address of the second		hasa		ain a a à h-bh dha	A firsh at the second second second second second second
					Zoom in : 2.51 Duration : 1.9	
🎕 Time Profiler 🔷 🖨	\ge \equiv Call Tree \Rightarrow	Call Tree				=
Sample Perspective	Running (Self)	Symbol Name				
 All Sample Counts 	3484.0ms 51.49					
 Running Sample Times 	2045.0ms 30.19					
Call Tree	660.0ms 9.79					
Separate by Thread	367.0ms 5.49	- · · · · · · · · · · · · · · · · · · ·				
✓ Invert Call Tree	51.0ms 0.79					
Hide Missing Symbols	42.0ms 0.69	· · · · · · · · · · · · · · · · · · ·				
Hide System Libraries	32.0ms 0.49	,				
Show Obj-C Only	19.0ms 0.29					
Flatten Recursion	18.0ms 0.29	· · · · · · · · · · · · · · · · · · ·				

Zoom In Shift + Mouse Drag

000		Instruments k ²
(II) (All Processes	;	
Record Target		Inspection Range Run 1 of 1 View Library Search
(🔳 🗮) 🦳 All Cores	÷)[All Processes
Instruments		
Fime Profiler (i)		Anin product hilled while allow the new pression of the state of a state of a state of the state of the state of
🎕 Time Profiler 🔶	\ge Call Tree 🗢	Call Tree =
Sample Perspective	Running (Self)	Symbol Name
 All Sample Counts 	3484.0ms 51.4%	▶Terminal (774)
Running Sample Times	2045.0ms 30.1%	▶ find (29064)
Call Tree	660.0ms 9.7% 367.0ms 5.4%	WindowServer (106)
Separate by Thread	51.0ms 5.4%	mach_kernel (0) Instruments (28958)
✓ Invert Call Tree	42.0ms 0.6%	► hidd (82)
Hide Missing Symbols	32.0ms 0.4%	▶Preview (28813)
Hide System Libraries Show Obi C Only	19.0ms 0.2%	▶iCal (633)
Show Obj-C Only Flatten Recursion	18.0ms 0.2%	▶Dock (148)
I Hatten Kecursion	18.0mc 0.2%	Dock (137)

Zoom Out Control + Mouse Drag

000		Instruments R ⁷
(II) (All Processes	;) [
Record Target		nspection Range Run 1 of 1 View Library Search
All Cores	\$]	All Processes
Instruments		
F ime Profiler (1)		المعين الالاني والفليلي والمليلين والملحلين وتعاصيني وتعاصين والملحون والالان والمليس والالان
		om Out : 4.83 ms/pixel Iration : 1.00 sec
🄕 Time Profiler 🔶	\geq E Call Tree \Rightarrow	Call Tree =
▼ Sample Perspective	Running (Self)	Symbol Name
O All Sample Counts	3484.0ms 51.4%	▶Terminal (774) 💿
 Running Sample Times 	2045.0ms 30.1%	▶find (29064)
▼ Call Tree	660.0ms 9.7%	►WindowServer (106)
Separate by Thread	367.0ms 5.4%	▶mach_kernel (0)
🗹 Invert Call Tree	51.0ms 0.7%	▶Instruments (28958)
Hide Missing Symbols	42.0ms 0.6%	▶hidd (82)
Hide System Libraries	32.0ms 0.4%	▶ Preview (28813)
Show Obj-C Only	19.0ms 0.2%	▶iCal (633)
Flatten Recursion	18.0ms 0.2%	Dock (148)

Zoom Out Control + Mouse Drag

00	and a fair to a state of a state of the desire of the desire of the state of the			Instruments	in an an ann a fhair an an ann. Na bhliann 16 Anna ann ann ann	ala dan basil basa dan seri da Katalan Basara basar Basara Manangan Basara dan seri dan basar		and a first standard and an Indonesia internet and first standard a first standard	R ₂₁
(II) (All Processes	;	[000	00:00:00 G			Q- Proces	S	
Record Target		in the second	nspection Range	Run 1 of 1	► View	Library		Search	
All Cores		•	All Processes	\$]	All Threads	+			
Instruments		111111	\bigtriangledown						
Fime Profiler (i)		L .		handle bill and a street					
鎏 Time Profiler 🔶 🗢	∃ Call Tr	ee 🕈 🔪	Call Tree						=
Sample Perspective	Running	(Self)	Symbol Name						
O All Sample Counts	3484.0ms		▶Terminal (774)						
 Running Sample Times 	2045.0ms		▶find (29064)						
Call Tree	660.0ms		▶WindowServer (10)	06)					
Separate by Thread	367.0ms		▶mach_kernel (0)						
✓ Invert Call Tree	51.0ms	0.7%	▶Instruments (289	58)					
Hide Missing Symbols	42.0ms	0.6%	▶hidd (82)						
Hide System Libraries	32.0ms	0.4%	▶Preview (28813)						
	19.0ms	0.2%	▶iCal (633)						
Show Obj-C Only	18.0ms	0.2%	▶Dock (148)						

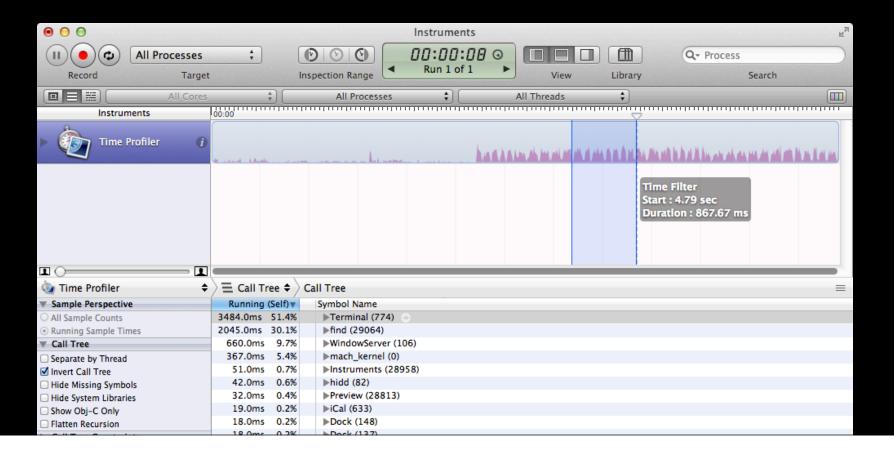
Snap Track to Fit Control + Command + Z

• • •		Instruments	R _M
II C C All Processes Record Target	÷ (Inspection Range Image: Construction Range Image: Con	
🔳 🗮 🧮 🔛 All Cores	÷) [All Processes	
Instruments			
Fime Profiler ()		الاس بدعام الحل الم المنه الله والمحد والتليم المحدي بالتلك ومحمل الملكي والم الم الله الم الم الم الم	
		control + command + z	
🄄 Time Profiler 🔶 🗢	\ge E Call Tree \Rightarrow	Call Tree	=
Sample Perspective	Running (Self)	Symbol Name	
O All Sample Counts	3484.0ms 51.4%	▶Terminal (774)	
 Running Sample Times 	2045.0ms 30.1%	▶find (29064)	
Call Tree	660.0ms 9.7%	▶WindowServer (106)	
Separate by Thread	367.0ms 5.4%	▶mach_kernel (0)	
✓ Invert Call Tree	51.0ms 0.7%	►Instruments (28958)	
Hide Missing Symbols	42.0ms 0.6%	▶hidd (82)	
Hide System Libraries	32.0ms 0.4%	▶ Preview (28813)	
Show Obj-C Only	19.0ms 0.2%	▶iCal (633)	
Flatten Recursion	18.0ms 0.2%	▶Dock (148)	
Coll Trace Constanting	18 0mc 0.2%	▶ Dock (127)	

Snap Track to Fit Control + Command + Z

00				Instruments				R _M
Record Tar		(nspection Range	00:00:00 ⊙ Run 1 of 1 ►	View	Library	Q- Process Search	
	res	÷)	All Processes	\$)	All Threads	\$		
Instruments	90:00							1
Time Profiler	()	h		hants		"TATUTURA	ia mili hultur lik bu yang ak ang manan ani ani ak hung ikan m	
	1							
🤕 Time Profiler	♦ E Call T	ree \Rightarrow	Call Tree					=
Sample Perspective	Running	(Self)	Symbol Name					
O All Sample Counts	3484.0ms	51.4%	▶Terminal (774)					
 Running Sample Times 	2045.0ms		▶find (29064)					
Call Tree	660.0ms		▶WindowServer (10)	06)				
Separate by Thread	367.0ms		▶mach_kernel (0)					
Invert Call Tree	51.0ms		Instruments (289)	58)				
Hide Missing Symbols	42.0ms		▶hidd (82)					
Hide System Libraries	32.0ms		▶Preview (28813)					
Show Obj-C Only	19.0ms		▶iCal (633)					
Flatten Recursion	18.0ms		▶Dock (148)					
	18.0mc	0.2%	▶ Dock (137)					

Select Time Range Option + Mouse Drag



Select Time Range Option + Mouse Drag

● ● ●					Instrument	s				R _M
Record All	Processes Target	;	a second beau	Inspection Range	00:00:0 Run 1 of 1		View	Librar	States a second second second second second	Search
	All Cores		+	All Processes	\$,	All Threads	\$		
Instruments	;	00:00							~	
🕨 🤖 Time Profil	ler ()	the second to the state			den 1	hanta	nyi (maini		an ain ng la la la la na an air a	a na da adi ah li na li a na
	I (\ \							
🍓 Time Profiler	+	\ge \equiv Call Tr	ree 🕈 🖯	Call Tree						≡
Sample Perspective		Running		Symbol Name						
O All Sample Counts		656.0ms		►Terminal (774)						
 Running Sample Times 		372.0ms		▶find (29064)						
Call Tree		88.0ms 61.0ms		 WindowServer (1) mach_kernel (0) 						
Separate by Thread		8.0ms	0.6%	► hidd (82)						
Invert Call Tree Hide Missing Symbols		5.0ms	0.4%	Finder (28750)						
Hide Missing Symbols		4.0ms	0.3%	Preview (28813))					
Show Obj-C Only		3.0ms	0.2%	▶iCal (633)						
Flatten Recursion		2.0ms		▶Dock (148)						
		1.0mc	0.0%	► System III Server	(128)					

Call Tree Data Mining

Focus call tree to isolate factors

O O Instruments													
(II) (C) 🙀 MIPMap.app	p :	$\bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc$	0:00:10 0			Q- Involves Syr	mbol						
Record Target		Inspection Range	Run 1 of 1 📃 🕨	View	Library	Se	arch						
				All Three de									
All Cores	¢ [All Processes	;][All Threads		01:00							
Instruments	90:00					01:00							
Fime Profiler (1)													
🔹 🗍	E Call Tree +	Call Tree					=						
Sample Perspective	Running (Self)						_						
All Sample Counts	13446.0ms 35.3%		on vector CoreGran	nics									
Running Sample Times	13446.0ms 35.3%												
Call Tree	13446.0ms 35.3%	s 35.3% 💽 wimg_interpolate_read CoreGraphics											
Separate by Thread	13446.0ms 35.3%												
Separate by Thread ✓ Invert Call Tree	13446.0ms 35.3%												
Hide Missing Symbols	13446.0ms 35.3%	-	▼ripc_AcquireImage libRIP.A.dylib										
Hide System Libraries	13446.0ms 35.3%												
Show Obj-C Only	13446.0ms 35.3%			ranhier									
✓ Flatten Recursion	13446.0ms 35.3%		Charge 'scaleImag	e' to callers									
Call Tree Constraints	11992.0ms 31.4%		Prune 'scaleImage	' and subtrees	5 MIPMap								
Specific Data Mining	11992.0ms 31.4%		Charge 'MiPMap' 1										
p specific bata mining	11992.0ms 31.4%		5 1	o boundary frames	ock_invoke_1	MIPMap							
	11992.0ms 31.4%	A V	natten miridap t	o boundary ridilles									
	11992.0ms 31.4%		Focus on subtree										
	11992.0ms 31.4%	0	Focus on calls ma	de by 'scaleImage'									
	1454.0ms 3.8%	▼(Ir	Focus on callers o		2 MIPMap								
	1454.0ms 3.8%			5									
	1454.0ms 3.8%	<u>л</u> т_	Focus on callers o	мичиар	ock_invoke_1	MIPMap							
	1454.0ms 3.8%	₽	Reveal in Xcode										
	1454.0ms 3.8%	0	nevea in Acoue										
	1454.0ms 3.8%	0	thread_start libsy	/stem_c.dylib									
	4263.0ms 11.1%	► IphaProviderGetByte	s ImagelO										

Focus call tree to isolate relevant subtrees

Charge 'scaleImage' to callers Prune 'scaleImage' and subtrees Charge 'MiPMap' to callers Flatten 'MiPMap' to boundary frames

Focus on subtree

Focus on calls made by 'scaleImage' Focus on callers of 'scaleImage' Focus on callers of 'MiPMap'

Reveal in Xcode

- Focus on entire tree beneath selected symbol Used to eliminate noise in call tree

Focus call tree to isolate relevant subtrees

Charge 'scaleImage' to callers Prune 'scaleImage' and subtrees Charge 'MiPMap' to callers Flatten 'MiPMap' to boundary frames

Focus on subtree

Focus on calls made by 'scaleImage' Focus on callers of 'scaleImage' Focus on callers of 'MiPMap'

Reveal in Xcode

Focus on all calls made by symbol
 Used to identify all calls made by symbol

Focus call tree to isolate relevant subtrees

Charge 'scaleImage' to callers Prune 'scaleImage' and subtrees Charge 'MiPMap' to callers Flatten 'MiPMap' to boundary frames

Focus on subtree Focus on calls made by 'scaleImage' Focus on callers of 'scaleImage' Focus on callers of 'MiPMap'

Reveal in Xcode

Focus on all callers of selected symbol
 Useful to identify who all calls the selected symbol

Focus call tree to isolate relevant subtrees

Charge 'scaleImage' to callers Prune 'scaleImage' and subtrees Charge 'MiPMap' to callers Flatten 'MiPMap' to boundary frames

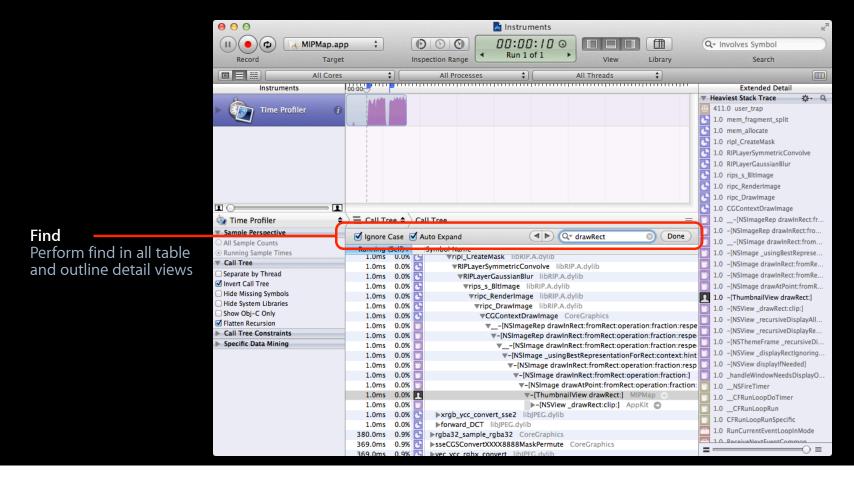
Focus on subtree Focus on calls made by 'scaleImage' Focus on callers of 'scaleImage' Focus on callers of 'MiPMap'

Reveal in Xcode

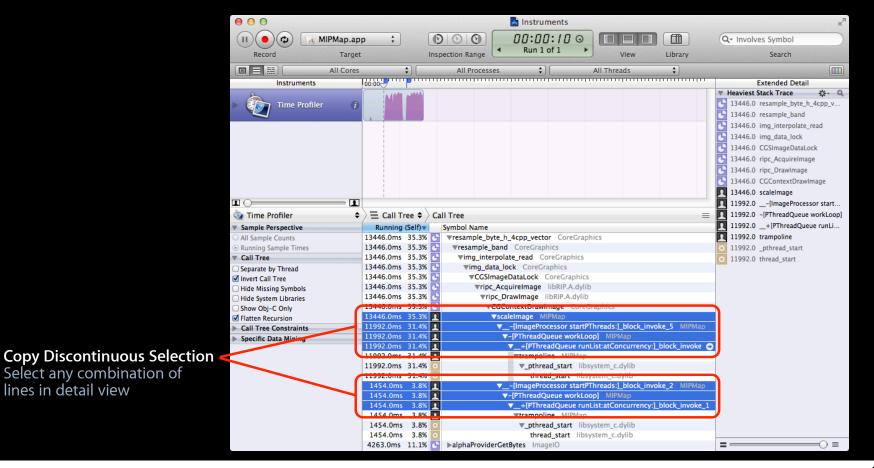
- Focus on all callers into the selected library Useful to see who all uses a given library

Find in Detail Views

Find data without filter out surrounding context

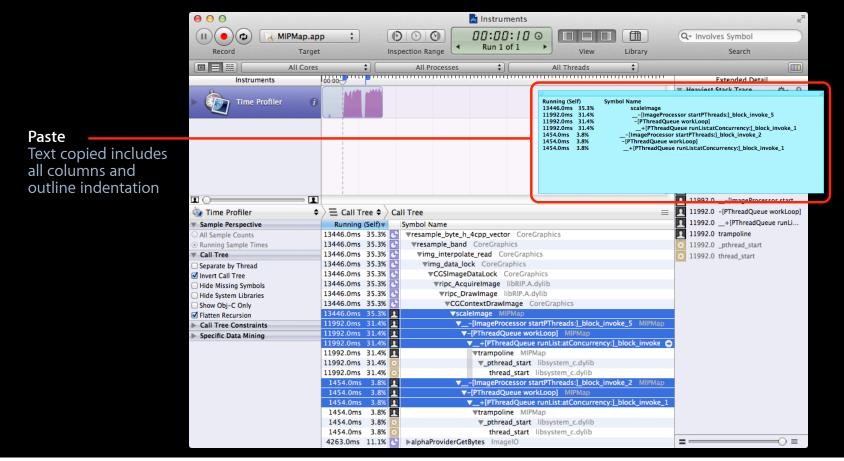


Stack Copy Discontinuous selection with shallow or deep copy

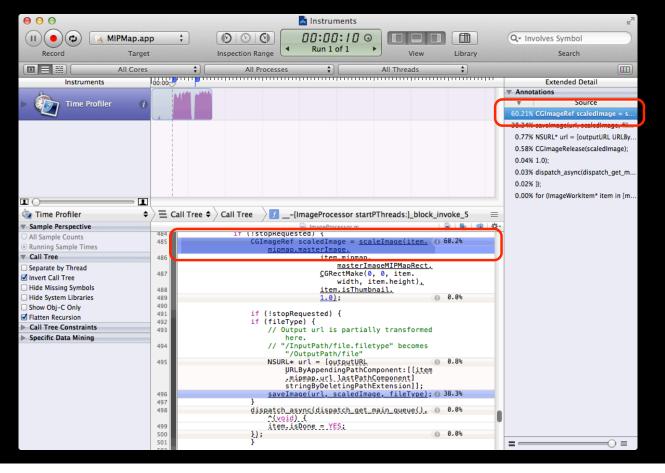


Stack Copy User Interface

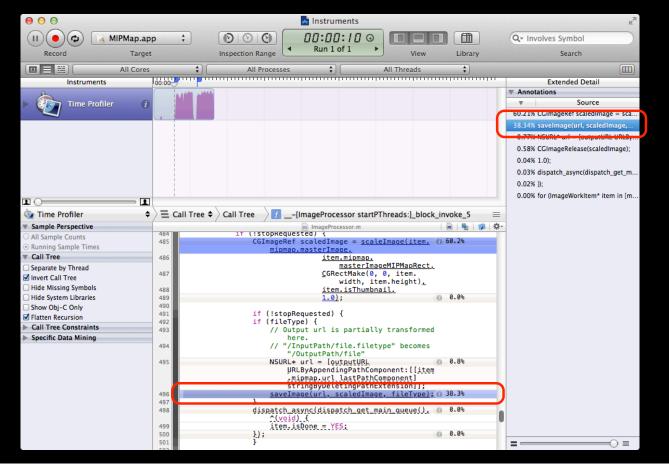
Discontinuous selection with shallow or deep copy



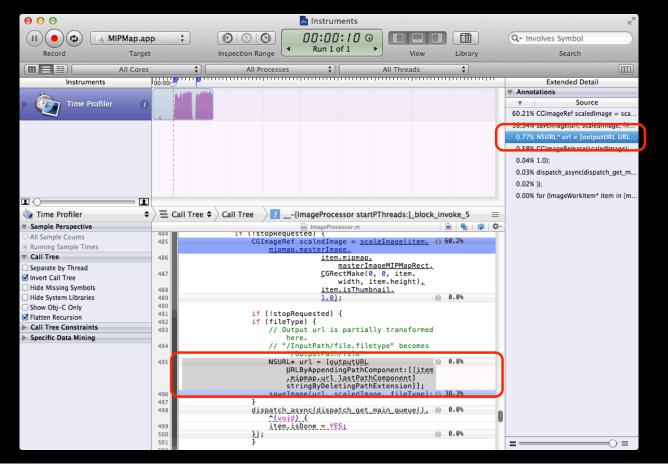
Navigation of Source Annotations Navigate by hottest to coolest hits

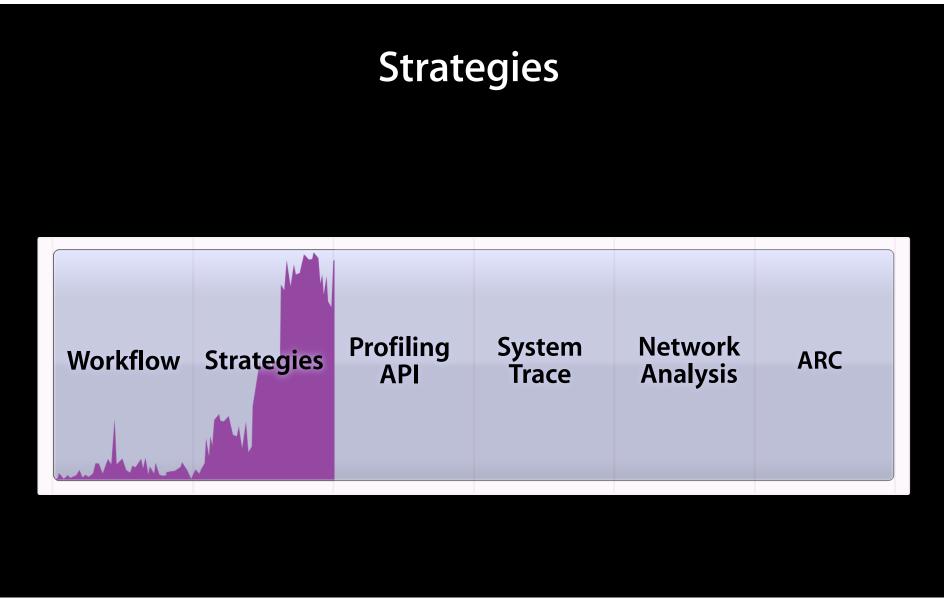


Navigation of Source Annotations Navigate by hottest to coolest hits



Navigation of Source Annotations Navigate by hottest to coolest hits





What Is a "Strategy?"

A method to categorize, display, and highlight data gathered from multiple instruments along a common axis

Strategies

The "Instruments" strategy

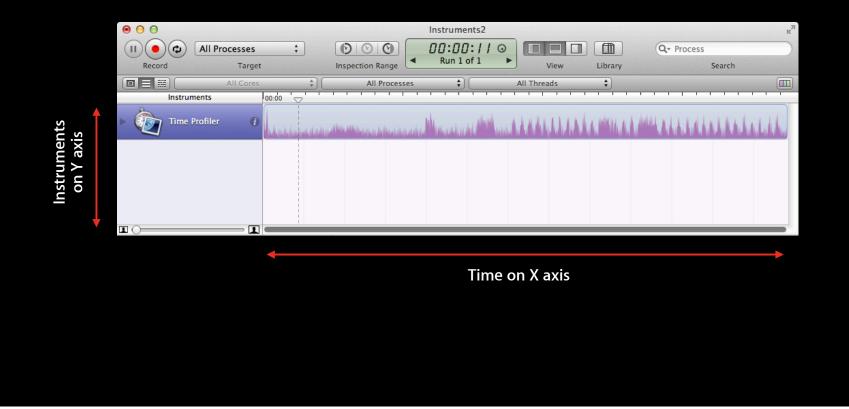
• • •				Instrum	ents2							R _M
All Processes	;	[୭ାତାତା	00:01	0:110				Q-	Process		
Record Target			spection Range			>	View	Library		Se	arch	
All Cores		÷)	All Processes	\$		All Threa	ads	\$				
Instruments	00:00 🗢									1 1 1		
► 🐑 Time Profiler →		data da		an Muran	all the bas		MM.	A.PHUA	7	Mun	MAG	١.
								-				
🔕 Time Profiler 🔷 🖨		ree 🕈 🔾 🕻	all Tree									=
Sample Perspective	Running	(Self)	Symbol Name									
O All Sample Counts	4762.0ms	51.0%	▶iTunes (13111) 🕤								
 Running Sample Times 	3083.0ms	33.0%	▶Keynote (1042)	5)								
▼ Call Tree	610.0ms	6.5%	▶WindowServer	(105)								
Separate by Thread	439.0ms	4.7%	▶coreaudiod (12)	24)								
Invert Call Tree	129.0ms	1.3%	▶mach_kernel (0)								
Hide Missing Symbols	67.0ms	0.7%	▶Xcode (12657))								
Hide System Libraries	52.0ms	0.5%	▶Instruments (1)									
Show Obj-C Only	32.0ms		▶Firefox (12499))								
Flatten Recursion	31.0ms		▶hidd (86)									
Call Tree Constraints	29.0ms	0.3%	▶iCal (143)									
Specific Data Mining	23.0ms	0.2%	▶Stickies (1016									
	15.0ms	0.1%	▶mDNSRespond	ler (11)								
	13.0ms		▶Dock (147)									
	9.0ms		▶Finder (1602)									
	7.0ms		▶SystemUIServe	r (148)								
	7.0ms	0.0%	▶mds (77)									
	4.0ms	0.0%	Istnoted (13)	1)								
	3.0ms	0.0%	▶Terminal (142))								
	3.0ms	0.0%	▶fseventsd (16)									
	2.0ms	0.0%	▶fontd (153)									
	2.0ms	0.0%	▶netbiosd (107)	0								
	2.0ms	0.0%	▶Pages (1500)									

Strategies

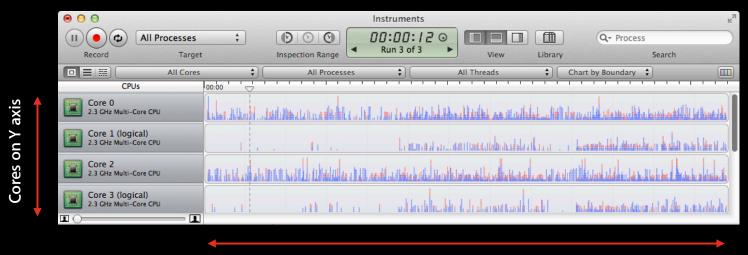
The "Instruments" strategy

. ● ● ●				Instruments2				н ^л
II O C All Pr	rocesses Target	:	Inspection Range	00:00:// © Run 1 of 1	View	Library	Q- Process	Search
	All Cores	\$	All Processes	•	All Threads	\$		
Instruments		00:00 🗢						
► 🤯 Time Profiler	0		al kunatti da ayan da kuna dan araka d	Manalust		W. MINNY	IN THE WAY	MARALIA
	— I							

Strategies The "Instruments" strategy



Strategies The "CPU" strategy



Time on X axis

Strategies The "Threads" strategy



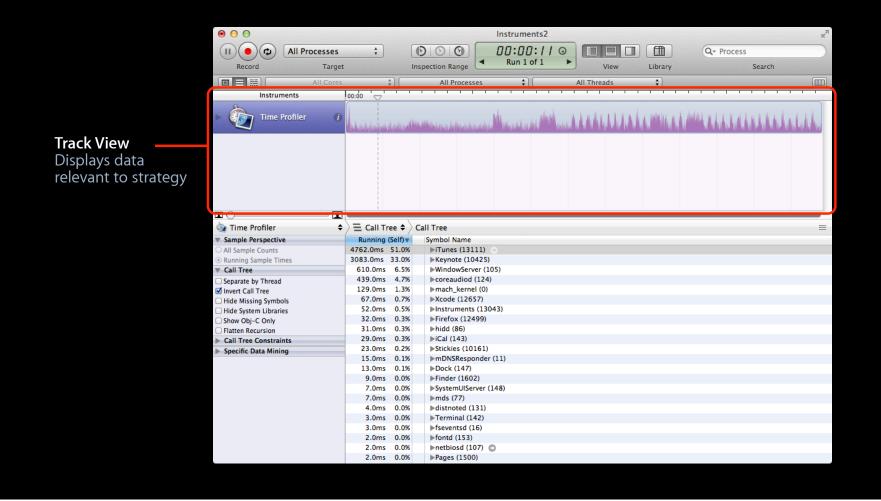
What Are the Elements of a Strategy?

Strategy Chooser Controls which strategy to display

00			Instruments2				к ^л
(II) (All Processes	;	\square	00:00:11	0		Q- Process	
Record Target		spection Range	Run 1 of 1	► View	Library	Sear	ch
			•) (Jean	
All Cores	÷) [All Processes	÷) [All Threads	<u></u>		
Instruments	00:00						
Fime Profiler (i)	-		Margareter Mark	ALLIANALA	A PHILAD	MANANANA	Valuation
🄄 Time Profiler 🔶 🖨	E Call Tree 🕈	Call Tree					=
Sample Perspective	Running (Self)	Symbol Name					
O All Sample Counts	4762.0ms 51.0%	▶iTunes (13111)	ə				
 Running Sample Times 	3083.0ms 33.0%	▶Keynote (10425)					
▼ Call Tree	610.0ms 6.5%	▶WindowServer (10)5)				
Separate by Thread	439.0ms 4.7%	▶coreaudiod (124)					
Invert Call Tree	129.0ms 1.3%	▶mach_kernel (0)					
Hide Missing Symbols	67.0ms 0.7%	▶Xcode (12657)					
Hide System Libraries	52.0ms 0.5%	Instruments (130	43)				
Show Obj-C Only	32.0ms 0.3%	▶Firefox (12499)					
Flatten Recursion	31.0ms 0.3%	▶hidd (86)					
Call Tree Constraints	29.0ms 0.3%	▶iCal (143)					
Specific Data Mining	23.0ms 0.2%	▶Stickies (10161)					
	15.0ms 0.1%	▶mDNSResponder	(11)				
	13.0ms 0.1%	▶Dock (147)					
	9.0ms 0.0%	▶Finder (1602)					
	7.0ms 0.0%	SystemUIServer (2)	148)				
	7.0ms 0.0%	▶mds (77)					
	4.0ms 0.0%	distnoted (131)					
	3.0ms 0.0%	▶Terminal (142)					
	3.0ms 0.0%	▶fseventsd (16)					
	2.0ms 0.0%	▶fontd (153)					
	2.0ms 0.0%	netbiosd (107)	D				
	2.0ms 0.0%	▶Pages (1500)					

Highlight Controls Controls highlighting of data in track and detail views

● ○ ○			an an an Aria. Tha man an ann an Araba an Araba Ann Araba ann an Araba an Araba an Araba Araba Canada ann an Araba an Araba an Araba	Instruments2				м ^л
(II) (C) All Processes	;	E	\mathbf{O}	00:00:11	3		Q- Process	
Record Target		and a state of the state of the	pection Range	Run 1 of 1	► Vie	the second s	c	Search
				20				
All Cores	÷		All Processes	‡	All Threads	: +		
Instruments	00:00 🗢							
Fine Profiler (i)	Internet descent of		Malatana kata pinana kata kata kata kata kata kata kata	Manager		DIN M ^{an} tan	V"INTERVER	
	i	_						
🔄 Time Profiler 🔶	E Call Tre		all Tree					=
		1						=
Sample Perspective	Running (S 4762.0ms 5		Symbol Name	•				
All Sample Counts	4762.0ms 3 3083.0ms 3		 iTunes (13111) Keynote (10425) 					
Running Sample Times Call Tree		6.5%	WindowServer (1					
		4.7%	► coreaudiod (124)					
Separate by Thread		1.3%	▶mach_kernel (0))				
☑ Invert Call Tree		0.7%	►Xcode (12657)					
Hide Missing Symbols Hide System Libraries		0.5%	▶Instruments (130	143)				
Show Obj-C Only		0.3%	Firefox (12499)	,,,,,				
Show Obj-C Only Flatten Recursion		0.3%	▶hidd (86)					
Call Tree Constraints		0.3%	▶iCal (143)					
Specific Data Mining		0.2%	▶Stickies (10161)					
P Specific Data Mining	15.0ms	0.1%	▶mDNSResponder					
	13.0ms	0.1%	▶Dock (147)					
	9.0ms	0.0%	▶Finder (1602)					
	7.0ms	0.0%	▶SystemUlServer (148)				
	7.0ms	0.0%	▶mds (77)					
	4.0ms	0.0%	▶distnoted (131)					
	3.0ms	0.0%	▶Terminal (142)					
	3.0ms	0.0%	▶fseventsd (16)					
		0.0%	▶fontd (153)					
		0.0%	▶netbiosd (107)	0				
		0.0%	▶Pages (1500)					



Legend Displays a key legend for the selected strategy

00	and de la Transford an a Transford de la Camport			Instruments2			Sulta Sultana Sulta		and the second sec	H
(II) (All Processes	;		000	00:00:11	0			Q- Proces		
	•			Run 1 of 1				Q+ Proces		-
Record Target		Ins	pection Range	Kull I OF I		View	Library		Search	
All Cores	4	;)	All Processes	\$	All Thr	eads	\$			
Instruments	00:00 🗢								1 1 1	
Fime Profiler ()	histophysic		Alf de la combe de la comptete de la	Instantia (h	Land AV		W.	V"WWWW	in the	
🤕 Time Profiler 🔶	\ge E Call Tr	ee 🗢 🖯 C	all Tree							=
Sample Perspective	Running (Self) 🖷	Symbol Name							
O All Sample Counts	4762.0ms	51.0%	▶iTunes (13111) 💮							
 Running Sample Times 	3083.0ms	33.0%	Keynote (10425)							
▼ Call Tree	610.0ms	6.5%	>WindowServer (105))						
Separate by Thread	439.0ms	4.7%	▶coreaudiod (124)							
✓ Invert Call Tree	129.0ms	1.3%	▶mach_kernel (0)							
Hide Missing Symbols		0.7%	▶Xcode (12657)							
Hide System Libraries	52.0ms	0.5%	►Instruments (1304)	3)						
Show Obj-C Only	32.0ms	0.3%	▶Firefox (12499)							
Flatten Recursion	31.0ms	0.3%	▶hidd (86)							
Call Tree Constraints	29.0ms	0.3%	▶iCal (143)							
Specific Data Mining	23.0ms	0.2%	▶Stickies (10161)							
	15.0ms	0.1%	▶mDNSResponder (1)	1)						
	13.0ms	0.1%	▶Dock (147)							
	9.0ms	0.0%	▶Finder (1602)							
	7.0ms	0.0%	SystemUIServer (14)	8)						
	7.0ms	0.0%	▶mds (77)							
	4.0ms	0.0%	▶distnoted (131)							
	3.0ms	0.0%	▶Terminal (142)							
	3.0ms	0.0%	▶fseventsd (16)							
	2.0ms	0.0%	▶fontd (153)							
	2.0ms	0.0%	▶netbiosd (107) 🔘							
	2.0ms	0.0%	▶Pages (1500)							

CPU Strategy In-Depth

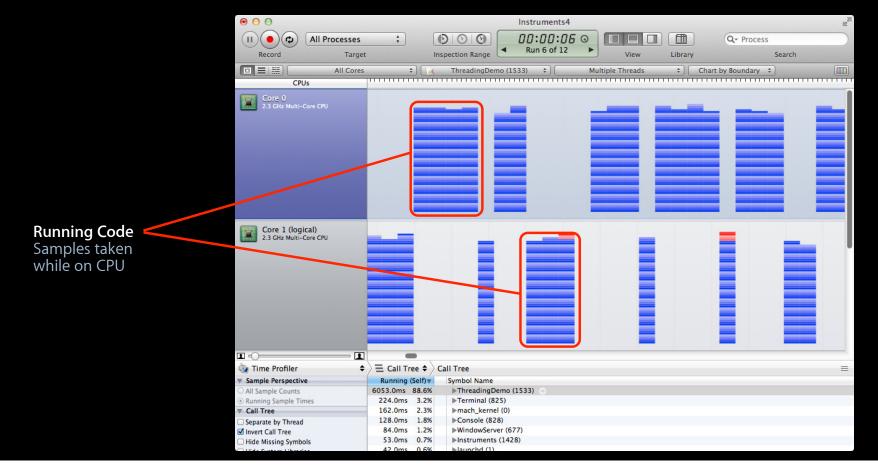
•••••	

Why Use the CPU Strategy?

- Determine frequency and duration that your code is on CPU
- Identify what keeps your threads off CPU
 - Poor concurrency
 - Lock contention
 - Busy system
 - Multiple processes contending for CPU

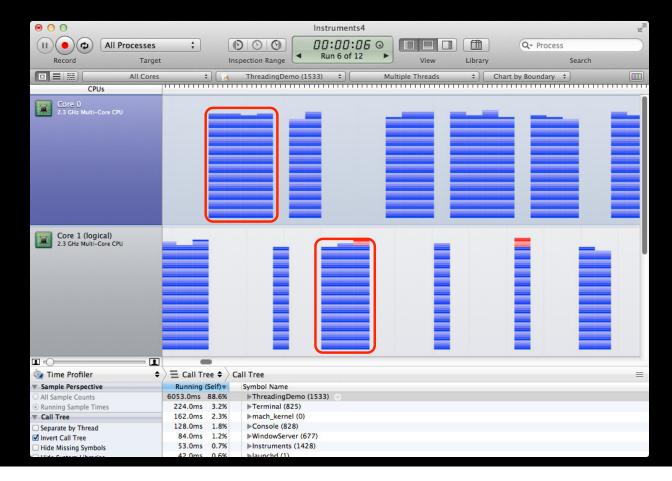
Poor Concurrency

Lock contention



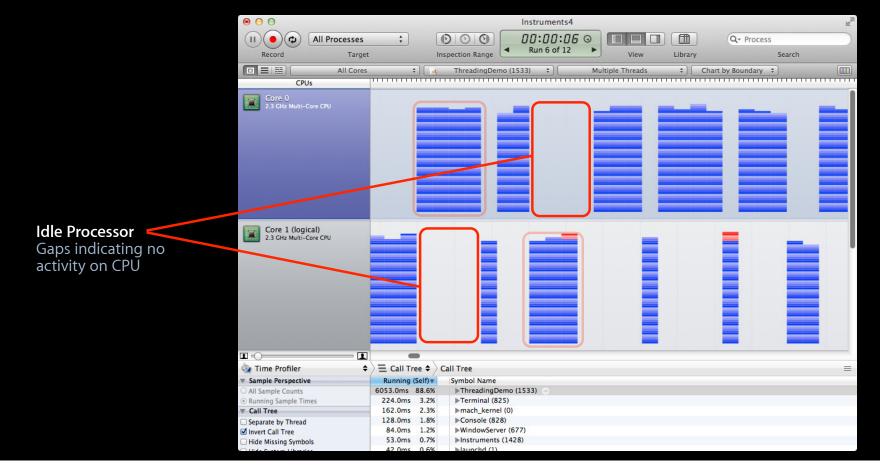
Poor Concurrency

Lock contention



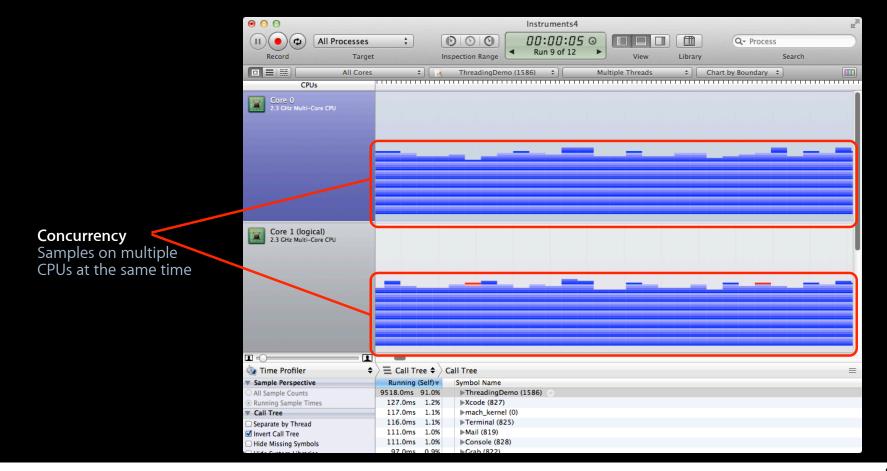
Poor Concurrency

Lock contention

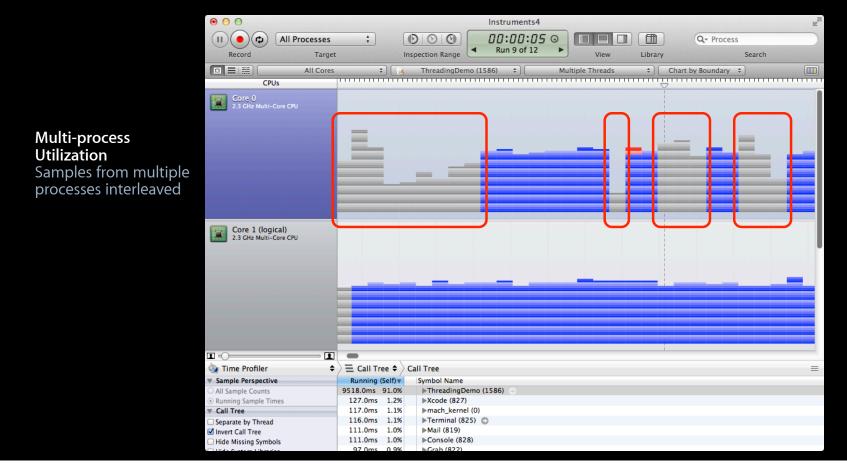


Idealized Concurrency

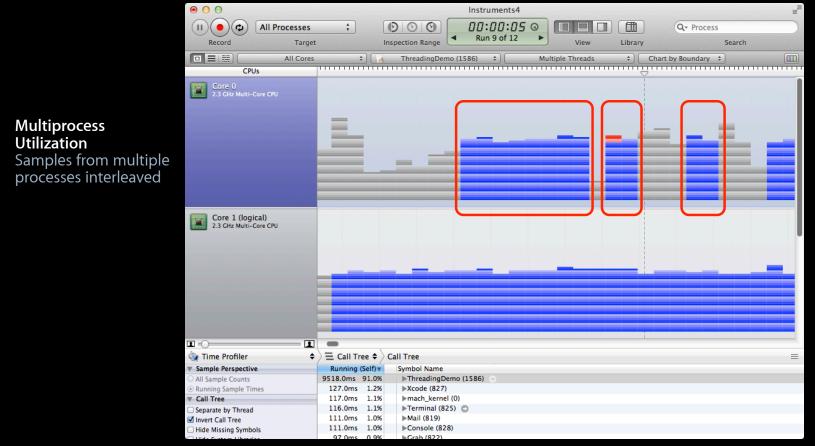
Lock contention eliminated with full CPU utilization



Concurrency in the Real World Lock contention eliminated with shared CPU utilization



Concurrency in the Real World Lock contention eliminated with shared CPU utilization

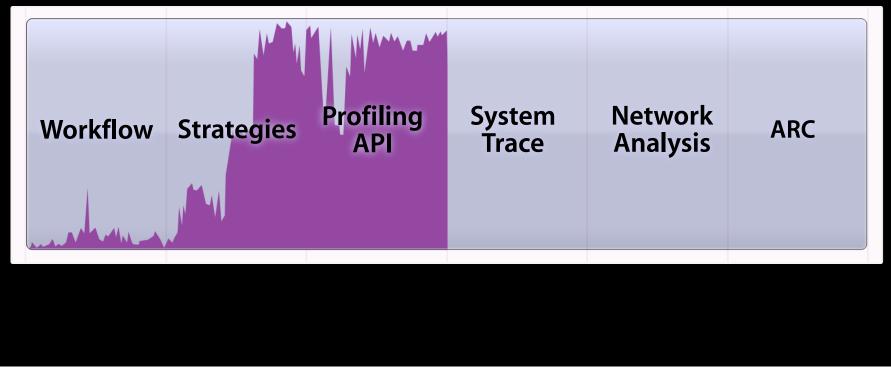


CPU Strategy Demo Detecting concurrency with time profiler



Daniel Delwood Performance Tools Engineer

Performance Analysis API Programmatically control data recording



DTPerformanceSession Framework

Only on Mac OS

- Target self or other processes (existing or launched)
- Time Profiler, System Trace, Leaks, Allocations (with zombies support), Activity Monitor, flags support
- View result in Instruments
 - 'DTPS' file has limited lifespan
 - Symbol data will be out of date once binary is rebuilt
 - Opening in Instruments converts to Trace Document and saves symbol data
- Location

/Library/Developer/4.0/Instruments/Frameworks/DTPerformanceSession.framework

Why Use DTPerformanceSession?

- Profile specific operations in your own code
- Automatically flag important events
- Profile your own performance regression tests

Using DTPerformanceSession API Setup session

- DTPerformanceSessionCreate(NULL,
 - CFStringCreateWithFormat(NULL, NULL, CFSTR("%d"), getpid()),
 - NULL, NULL);
- DTPerformanceSessionAddInstrument(session,
 - CFSTR(DTPerformanceSession_TimeProfiler), NULL, NULL, NULL);

Using DTPerformanceSession API Add calls to start/stop profilers

DTPerformanceSessionStart(session, NULL, NULL);
DTPerformanceSessionStop(session, NULL, NULL);

Using DTPerformanceSession API Insert sign posts

DTSendSignalFlag("All masterImages updated", DT_POINT_SIGNAL, true);

or...

DTSendSignalFlag("Master Images Update", DT_START_SIGNAL, true); DTSendSignalFlag("Master Images Update", DT_END_SIGNAL, true);

Using DTPerformanceSession API Save session

DTPerformanceSessionSave(session, CFSTR("/tmp/WWDC2011"), NULL); CFRelease(session);

Viewing Results Open DTPS file in Instruments

00		Instruments16	R
(II) (Choose Target	;	O O O O O O O O O O O O O O O O O O O	
Record Target	1	Aspection Range Run 1 of 1 View Library Search	
All Cores	÷) [All Processes All Threads	
Instruments	00:00	∇	
Time Profiler (i)			
🄄 Time Profiler 🔷 🖨	\ge Call Tree \Rightarrow	Call Tree	≡
Sample Perspective	Running (Self)	Symbol Name	
 ○ All Sample Counts ⊙ Running Sample Times ▼ Call Tree 	10397.0ms 83.0% 2111.0ms 16.8% 7.0ms 0.0%	 bthread_start libsystem_c.dylib bstart MIPMap b_dispatch_mgr_wakeup libdispatch.dylib 	
 Separate by Thread Invert Call Tree Hide Missing Symbols Hide System Libraries Show Obj-C Only Flatten Recursion Call Tree Constraints Specific Data Mining 	1.0ms 0.0%	▶0x7fff5fbfefe0	

iProfiler Built with DTPerformanceSession framework

iprofiler(1)

BSD General Commands Manual

iprofiler(1)

NAME

iprofiler, version 1.0

USAGE

```
iprofiler [-1] [-L] [-legacy] [-T duration] [-I sampling interval] [-d path] [-o basename]
  [-activitymonitor] [-allocations] [-leaks] [-systemtrace] [-timeprofiler] [-kernelstacks
  | -userandkernelstacks] [-allthreadstates] [-a process/pid | executable [args...]]
```

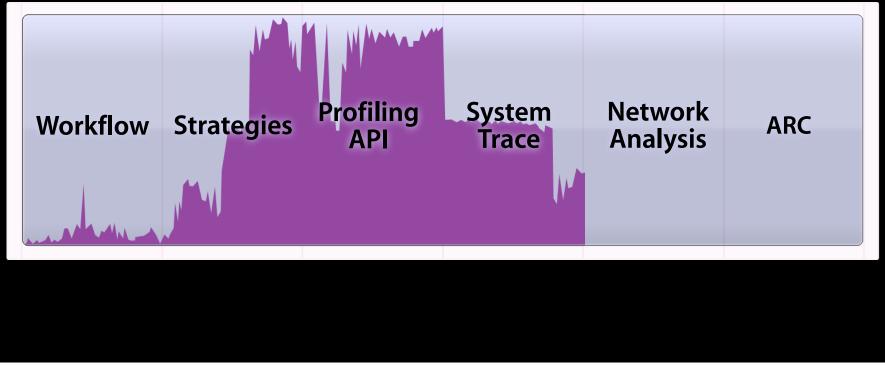
DESCRIPTION

Measure an application's performance without launching Instruments.app and then visualize the measurements at a later time in Instruments.app. The performance data gets saved in a .dtps bundle that can be opened in Instruments.app via "Open existing file...". iprofiler supports these instruments: Time Profiler, System Trace, Activity Monitor, Allocations, and Leaks. Any combination of these instruments can be run simultaneously. iprofiler supports attaching to a currently-running process, launching a process that will only run during the measurement, or profiling all currently-running processes (by not specifying process/pid or executable).

Options are :

- -1 Lists all supported instruments
- -L Lists all supported instruments, with a description

System Trace Comprehensive system analysis



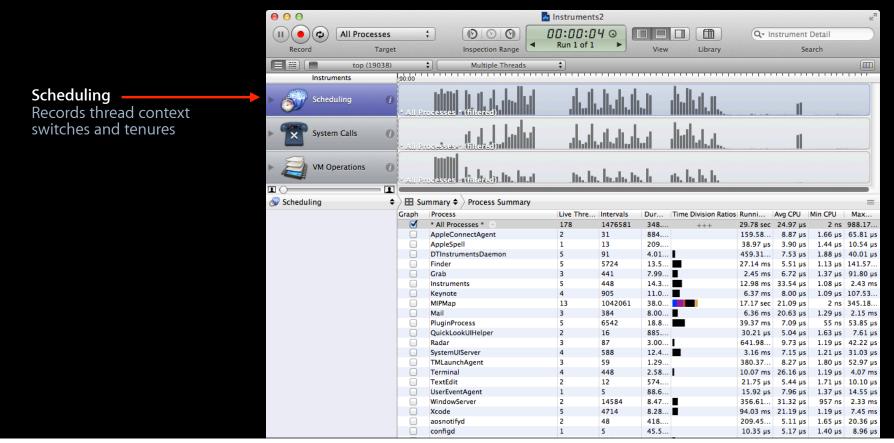
What Is System Trace?

• NEW in iOS 5!

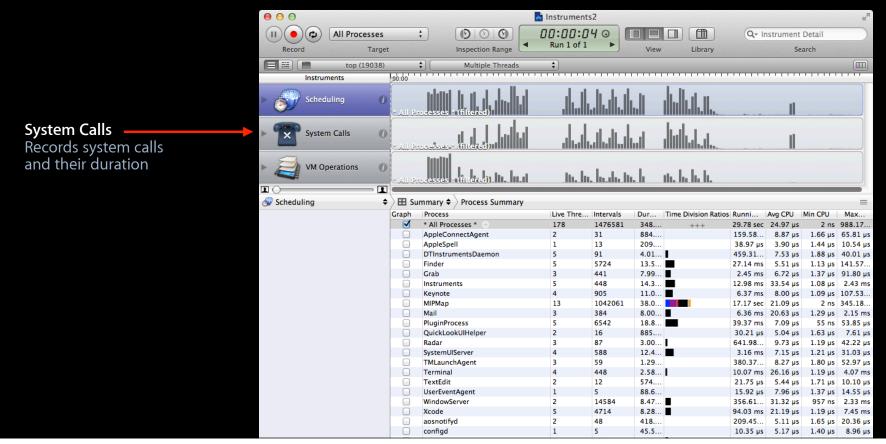
- Comprehensive analysis of system behavior
 - Thread scheduling
 - Virtual memory
 - System calls
- Available in Mac OS X since Instruments 4.0



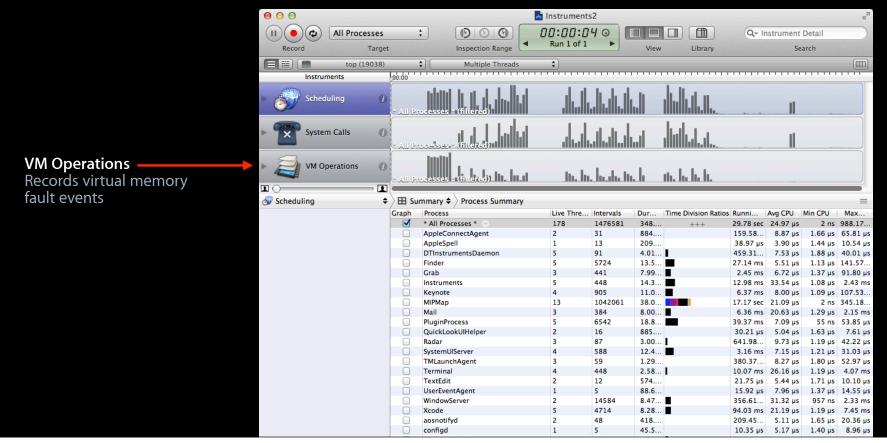
"Instruments" strategy



"Instruments" strategy



"Instruments" strategy



"Threads" strategy

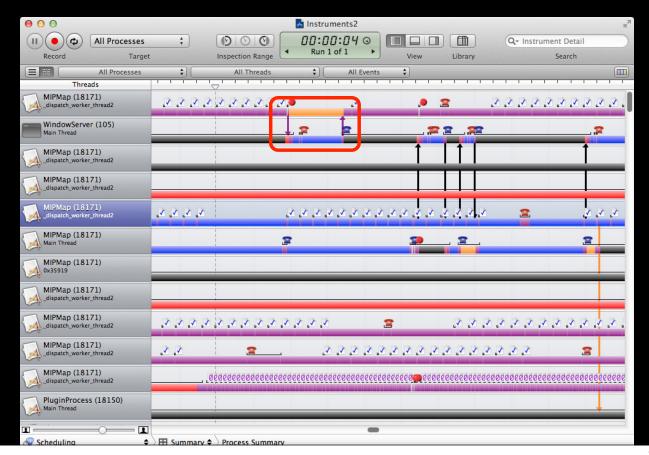
	$\Theta \Theta \Theta$		an star de la Pre ann 12 de la Carrello Man De Carlos		A	Instrument	ts2						м ^л
		All Processes		:	$\bigcirc \bigcirc \bigcirc \bigcirc$	00:00:0	140			Q-	nstrumen	t Detail	
	Record	Targe	The Columbia is a street	<u> </u>	Inspection Range	Run 1 of 1		View			Stores and the state of the	earch	
									Library		3	earch	
		All Processes		•	All Threads	÷	All Events	;]					
		Threads	20:00										
		(18171) _worker_thread2		66	<u>((,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,</u>	((1.110		(.(.(()	<u>.(((((((()</u> (Ø				•
	Window Main Three	Server (105) ^{ad}			emp pare				<u>Ø</u>	2,2022	ń	2 22	
s, VM faults,		(18171) worker_thread2			<u>0,x,((.(((.((,((,()</u>		<u>(.(1</u> Ø		<u></u>				
states	🔪 📈 _dispatch	(18171) worker_thread2					<u></u> e	<u></u> Ø	<u>(()()</u>				
			1										_
	🔊 Schedulin	g 🗧	/	ummary :	/ .								=
			Graph	Process * All Bro	ocesses * 💿	Live Thre 178	1476581	348	Time Division Ratio		24.97 µs		Max s 988.17
					onnectAgent	2	31	884					s 65.81 µs
				AppleS		1	13	209		38.97 µs			s 10.54 µs
					umentsDaemon	5	91	4.01		459.31			s 40.01 µs
			ŏ	Finder		5	5724	13.5	-		5.51 µs		s 141.57
			ŏ	Grab		3	441	7.99		2.45 ms			s 91.80 µs
			ŏ	Instrum	ients	5	448	14.3	-		33.54 µs		s 2.43 ms
				Keynote	e	4	905	11.0		6.37 ms	8.00 µs	1.09 µ	s 107.53
				MIPMap)	13	1042061	38.0		17.17 sec	21.09 µs	2 n	s 345.18
				Mail		3	384	8.00		6.36 ms	20.63 µs	1.29 µ	s 2.15 ms
				PluginP	rocess	5	6542	18.8		39.37 ms	7.09 µs	55 n	s 53.85 µs
				QuickLo	ookUIHelper	2	16	885		30.21 µs	5.04 µs	1.63 µ	s 7.61 µs
				Radar		3	87	3.00		641.98	9.73 µs	1.19 μ	s 42.22 µs
				System	UlServer	4	588	12.4		3.16 ms	7.15 µs	1.21 µ	s 31.03 µs
				TMLaur	nchAgent	3	59	1.29		380.37	8.27 µs	1.80 µ	s 52.97 µs
				Termin	al	4	448	2.58	1	10.07 ms	26.16 µs	1.19 µ	s 4.07 ms
				TextEdi	it	2	12	574		21.75 µs	5.44 µs	1.71 µ	s 10.10 µs
				UserEve	entAgent	1	5	88.6		15.92 µs	7.96 µs	1.37 µ	s 14.55 µs
				Window		2	14584	8.47			31.32 µs		s 2.33 ms
			Ō	Xcode		5	4714	8.28	-		21.19 µs		s 7.45 ms
			ŏ	aosnoti	fvd	2	48	418	_	209.45			
												1.05 u	s 20.30 us

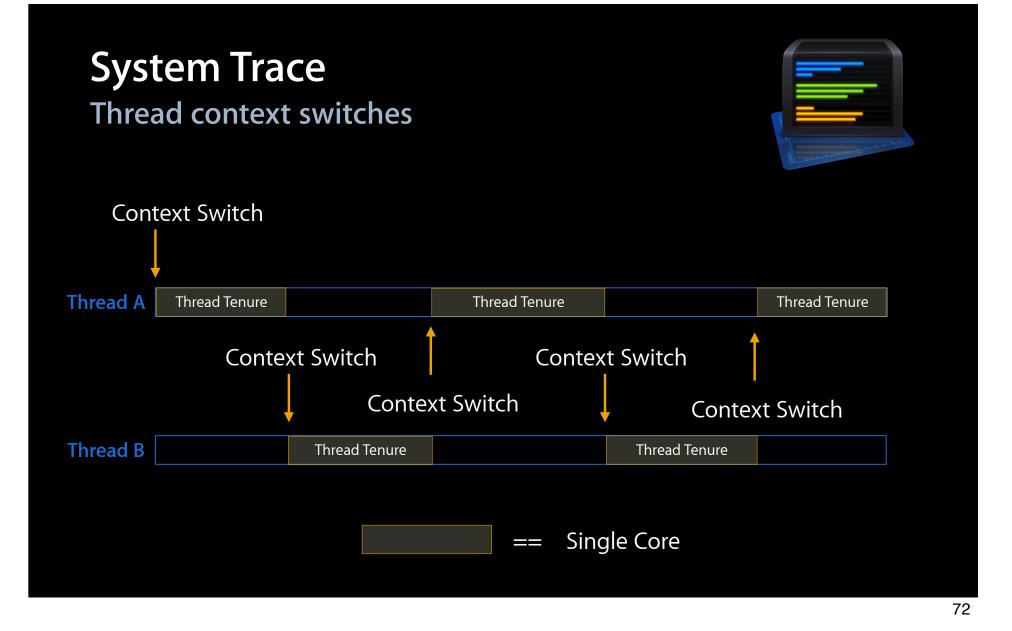
70

"Threads" strategy

Thread Context Switches

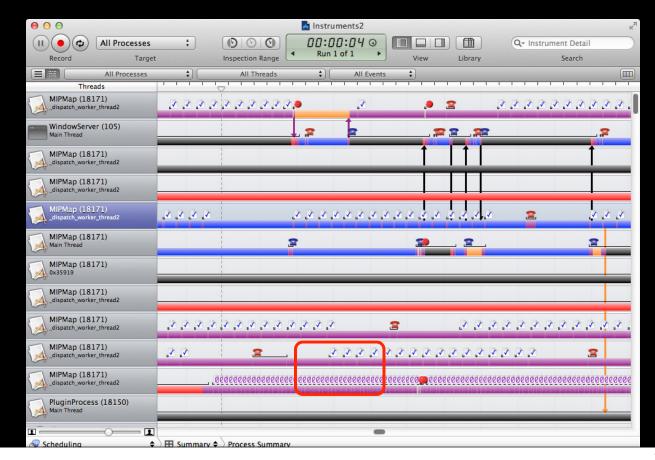
Arrows indicate threads scheduled onto cores





System Trace

"Threads" strategy



Virtual Memory Events lcons indicate cache hits,

zero fills, page-ins, etc.

System Trace Virtual memory events



- 😫 Copy On Write
- 🙆 Zero Fill
- 膏 Page-in
- Page Cache Hit
- 🛯 Non-Zero Fill

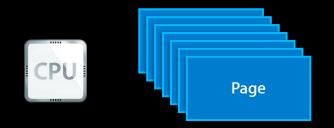
Guard Page
 File Backed Page-in
 Anonymous Memory Page-in
 Page-out

Virtual Memory Events Explained Memory is managed in segments called "Pages"



73°		Virtual Memory
		Virtual Memory

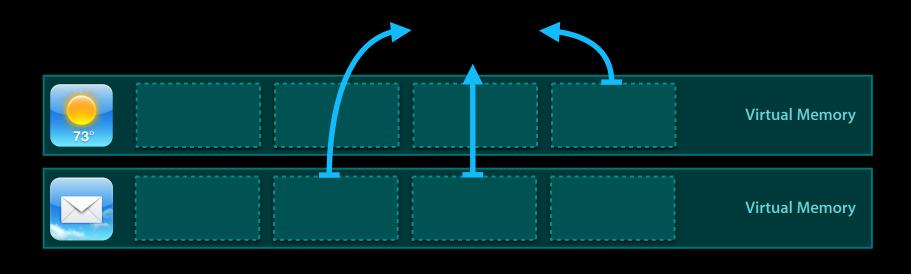
Virtual Memory Events Explained Virtual Pages become real Pages by "Page faults"



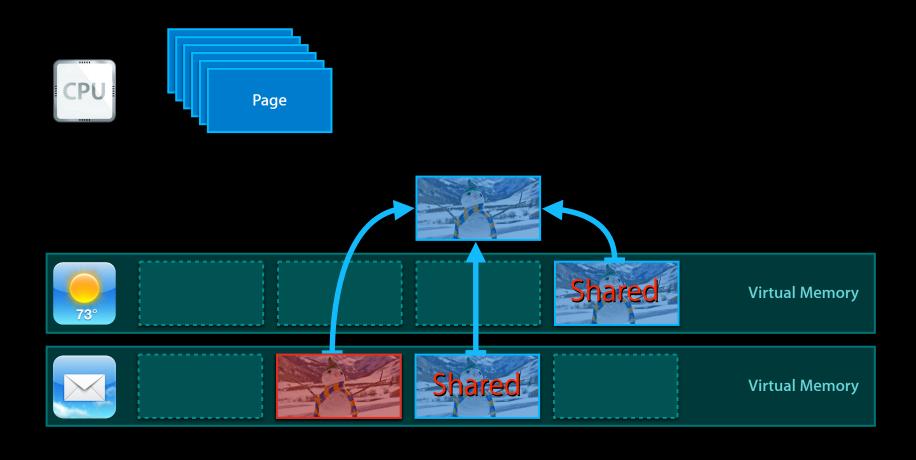
73°		Virtual Memory
		Virtual Memory

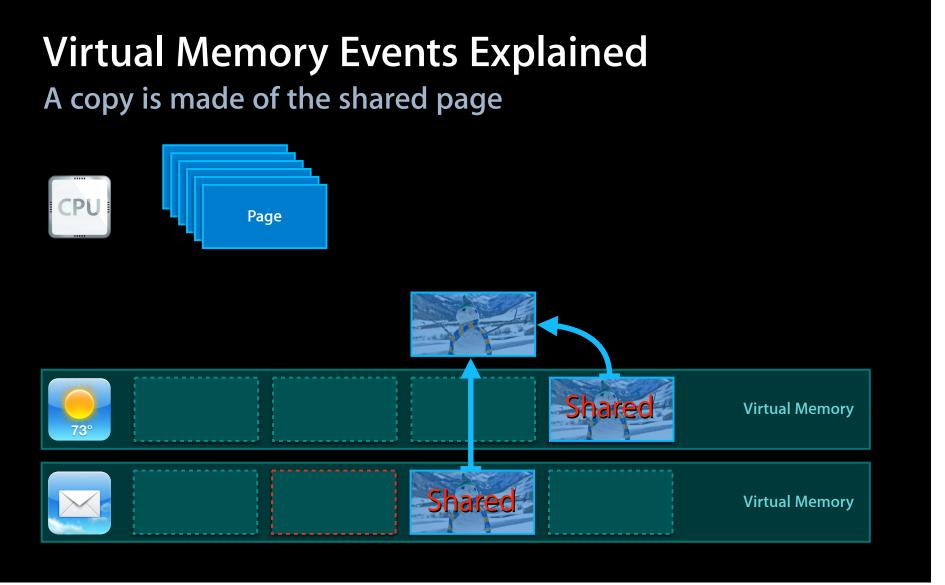
Virtual Memory Events Explained Pages can be shared





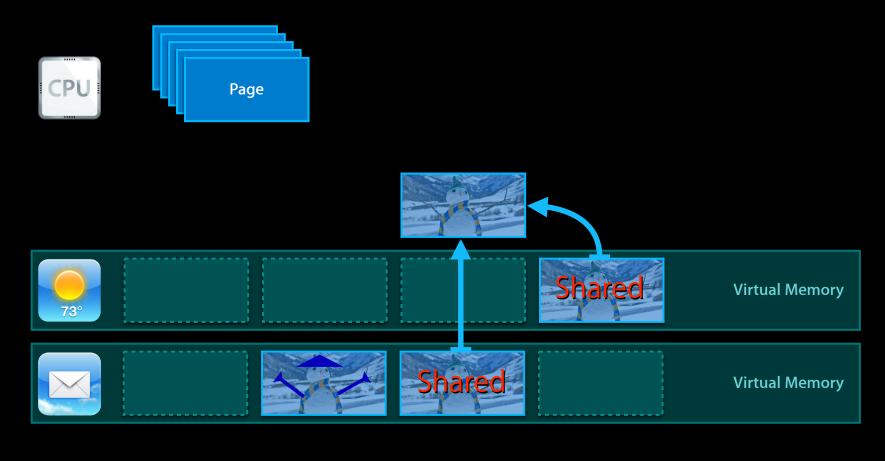
Virtual Memory Events Explained What happens when someone writes to a shared page?



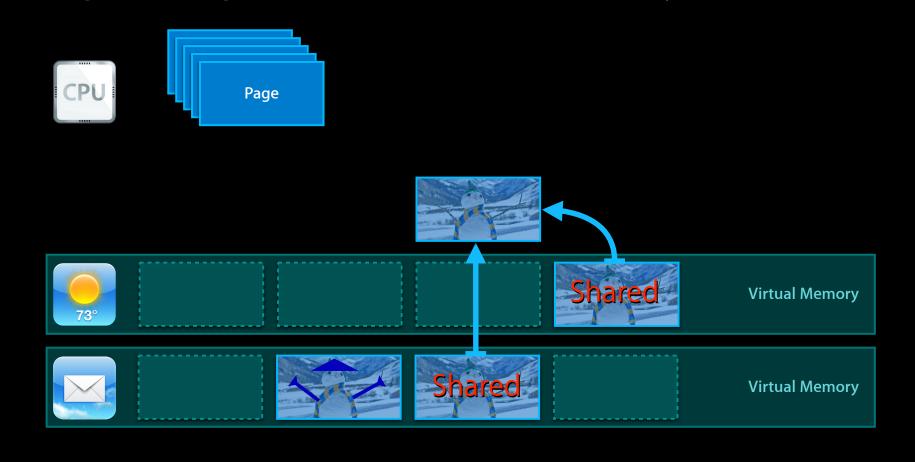


Virtual Memory Events Explained

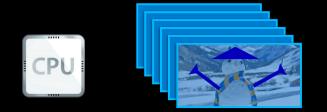
The write completes, thus a "Copy On Write" page fault



Virtual Memory Events Explained Pages no longer used are returned to the VM system



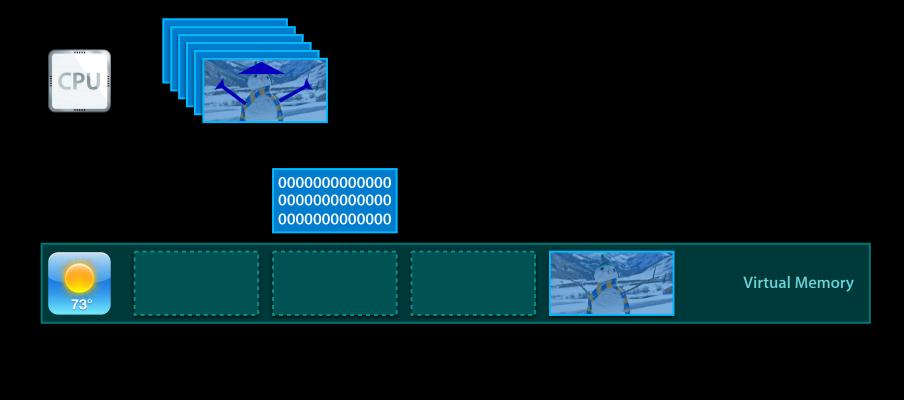
Virtual Memory Events Explained What happens when that memory is reused?





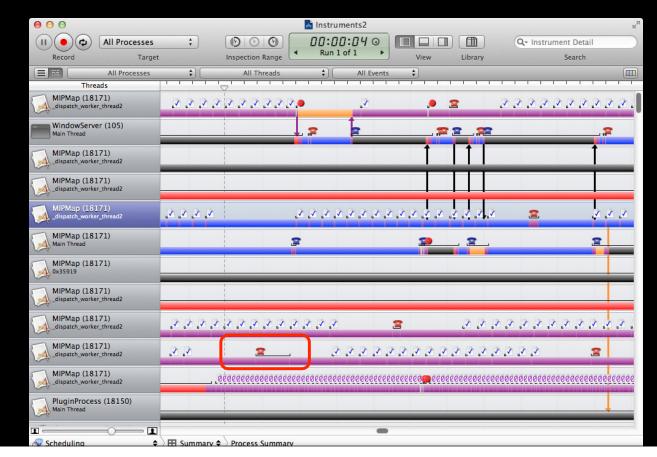
Clearly, this would be a problem!

Virtual Memory Events Explained The system performs a "Zero Fill" page fault



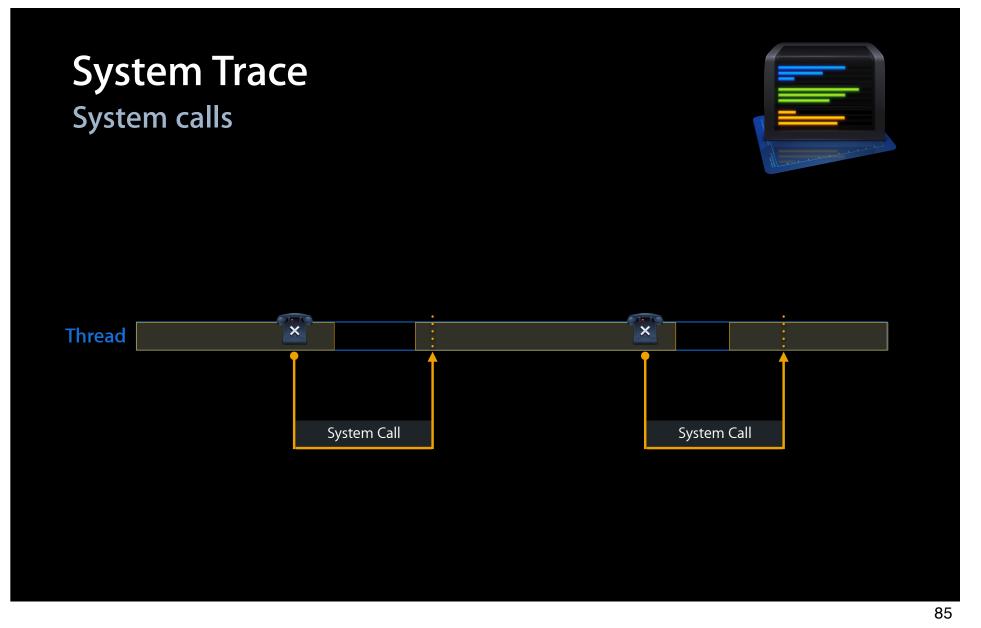
System Trace

"Threads" strategy



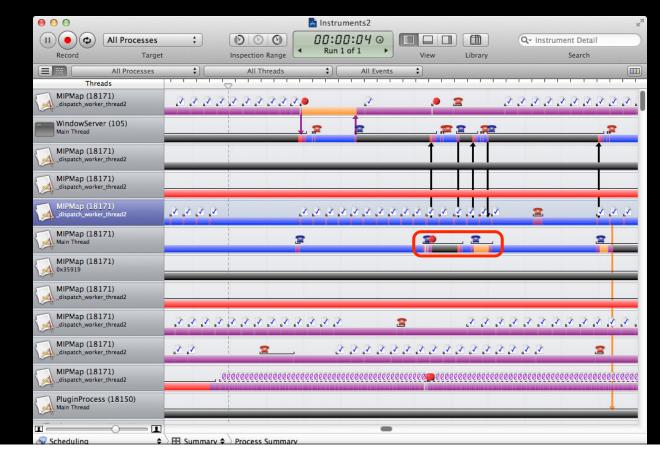
System Calls

Telephones indicate calls from user space into the kernel

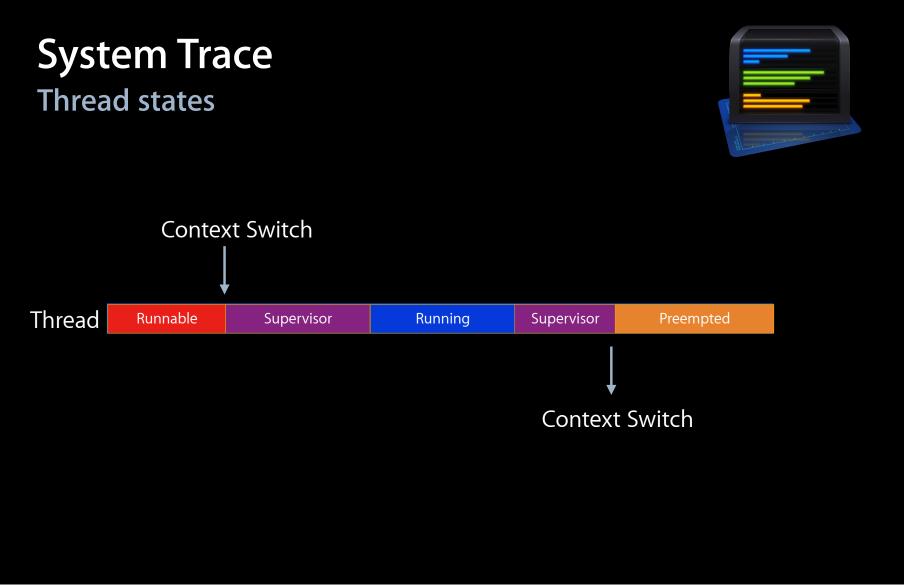


System Trace

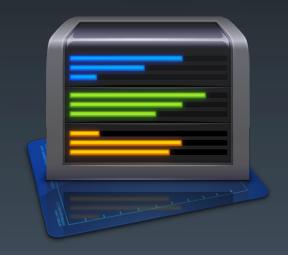
"Threads" strategy



Thread States Colored bars indicate thread state



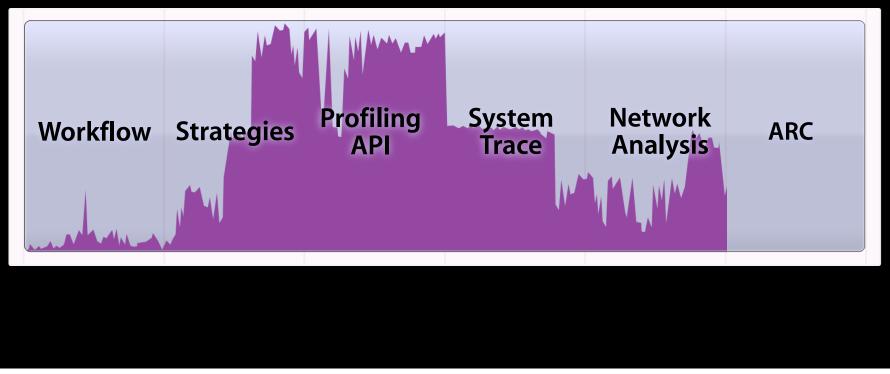
System Trace Demo



Daniel Delwood Performance Tools Engineer

Network Instrumentation

Track data flow and radio usage

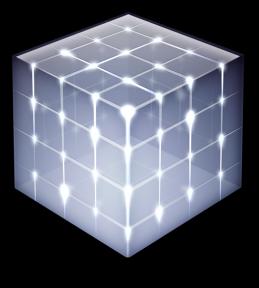


Network Connections Data flow statistics



Network Connections Instrument





- Measures data volume
 TCP/IP and UDP/IP
 - Performance metrics
- Debug latency
- Identify lost packets and re-transmission of data

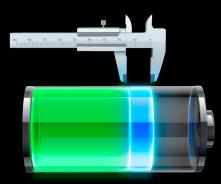
Network Connections Instrument New in iOS 5

00		In	struments4				M
II • C All Processes	‡		0:01:30 0		Q- Instrument	Detail	
Record Target		Inspection Range	Run 1 of 1 🕨 Vie	w Library	Se	arch	
Instruments	00:00		······································		01:00		
Network Connections	بالبي	II. III					
Network Connections 🔶	Connection Sur	nmary 🗢				1	=
	Process	Local	Remote	Data In 🔻 Packets I	n Data Out Packets	Duplicat Ou	it-o
	kernel(0)	tcp4 localhost:12994	tcp4 localhost:12738	105.79 KB 324	0 Bytes 0	0 Bytes 0 I	Byte
	LazyTable(9755)	tcp4 17.244.68.249:14274	tcp4 205.180.175.173:20480	21.14 KB 17	494 Bytes 2	0 Bytes 3.3	77 H
	LazyTable(9755)	tcp4 17.244.68.249:13762	tcp4 205.180.175.173:20480	20.20 KB 16	494 Bytes 2	0 Bytes 2.0	02 F
	LazyTable(9755)	tcp4 17.244.68.249:14530	tcp4 205.180.175.172:20480	18.48 KB 15	494 Bytes 2	0 Bytes 3.4	41 H
	LazyTable(9755)	tcp4 17.244.68.249:16578	tcp4 205.180.175.172:20480	14.50 KB 11	247 Bytes 1	0 Bytes 0 I	Byte
	LazyTable(9755)	tcp4 17.244.68.249:16834	tcp4 205.180.175.172:20480	11.95 KB 10	247 Bytes 1	0 Bytes 0 I	Byte
	LazyTable(9755)	tcp4 17.244.68.249:17090	tcp4 205.180.175.172:20480	8.56 KB 7	247 Bytes 1	0 Bytes 3.2	22 1
	LazyTable(9755)	tcp4 17.244.68.249:17602	tcp4 205.180.175.173:20480	8.50 KB 7	247 Bytes 1	0 Bytes 3.	16
		tcp4 17.244.68.249:17602 tcp4 17.244.68.249:14018	tcp4 205.180.175.173:20480 tcp4 205.180.175.173:20480	8.50 KB 7 7.36 KB 6	247 Bytes 1 247 Bytes 1		16 H 35 H

5.

Network Activity Instrument





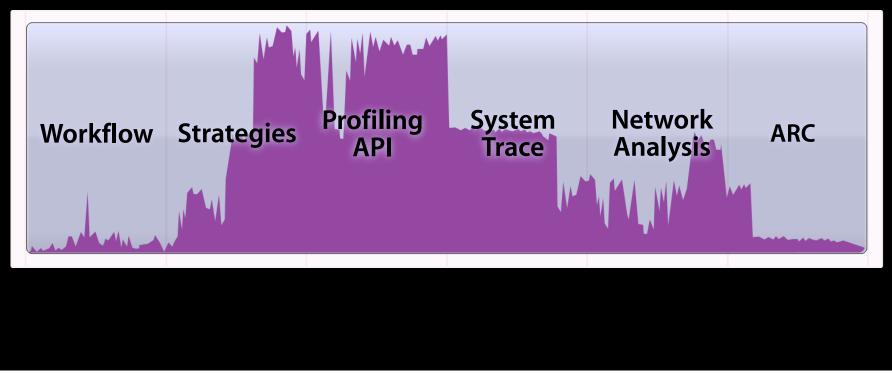
- Identify radio traffic in and out
 - Wi-Fi
 - Cellular
- Correlate power use with radio use

New in iOS 5

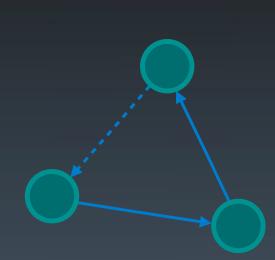


000				📥 Transf	erTrace					Ma
(II) (All Processo	es	;	$\bigcirc \bigcirc \bigcirc \bigcirc$		2:120			Q- Instrumen	t Detail	
Record Ta	rget		Inspection Rang	And the second sec		View	Library		earch	
Instruments		00:00						01:00		
Energy Usage	0	.	الألي							
Network Activity	0									
CPU Activity	0						·			
Display Brightness	0									
Sleep/Wake	0									
	1									
Network Activity	+	H Network Ad	ctivity 🗢							≡
				Cell Packets (out)			Cell Errors (in)	Cell Errors (out)		
		0	0	0	0		0	0	0	
		0	0	1	0	64	0	0		

ARC Instrumentation



ARC Instrumentation Leak cycle detection



Automatic Reference Counting (ARC) What's automatic about it?

- No more writing -retain/-release/-autorelease
- Enforces previous "convention"
- Eliminates many simple bugs
- Allows you to think about object relationships

Automatic Reference Counting (ARC) What's not automatic about it?

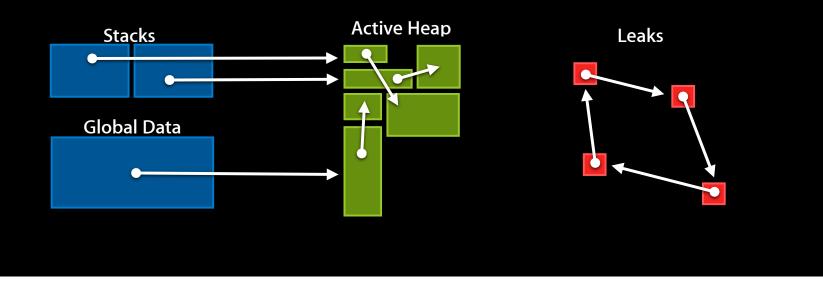
- It's not a garbage collector
- Runs alongside manual reference counting (MRC) code
- Cannot break retain cycles
- Cannot prevent leaks

Automatic Reference Counting (ARC) But Instruments can help!

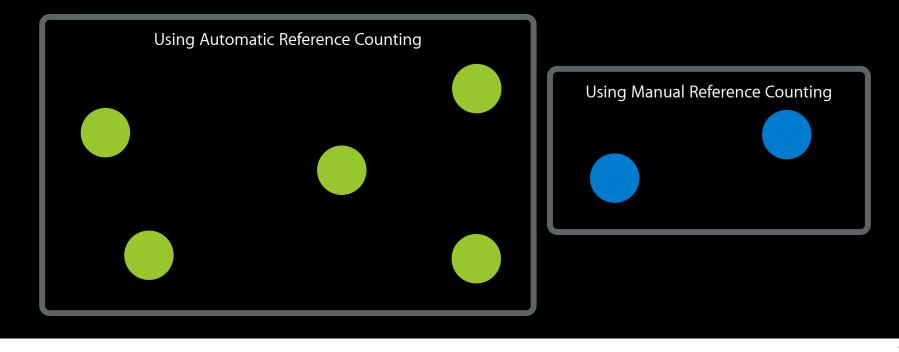
- Leaks instrument identifies
 - "Allocated memory that can no longer be reached"
 - No more pointers to it from the active heap

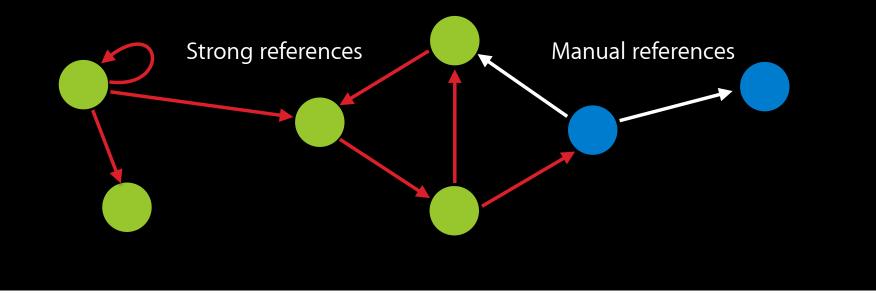
Automatic Reference Counting (ARC) But Instruments can help!

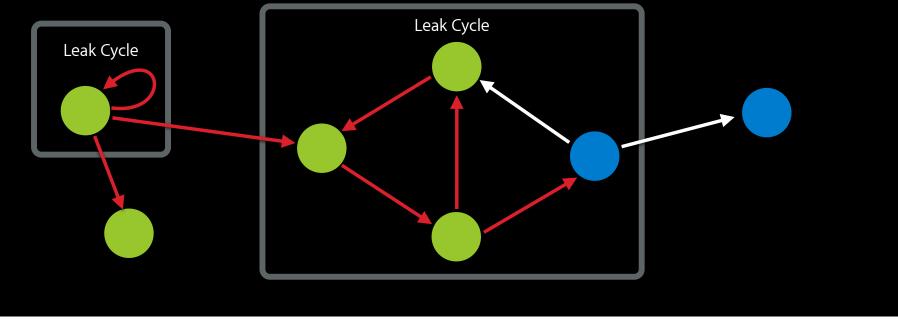
- Leaks instrument identifies
 - "Allocated memory that can no longer be reached"
 - No more pointers to it from the active heap
 - Cycle detection



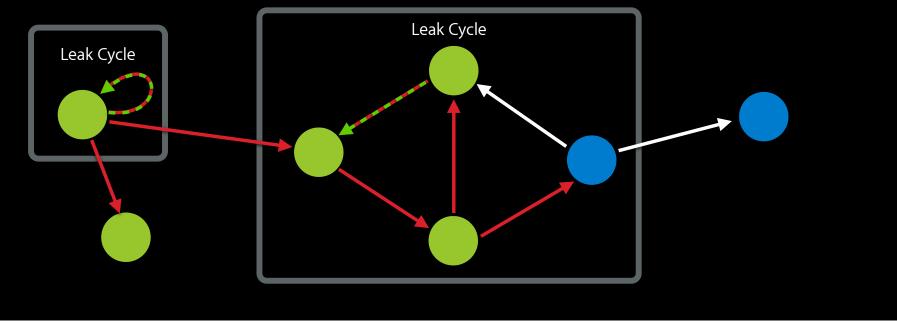








1. Fix the cycles



- 1. Fix the cycles
- 2. Fix remaining manual reference counting leaks



ARC Demo



Daniel Delwood Performance Tools Engineer

106

In Conclusion

- Tackle concurrency issues with the CPU strategy
- Examine the efficiency of your application via System Trace
- Leverage the new DTPerformanceSession framework
- Eliminate your leaked reference cycles with ARC instrumentation

More Information

Michael Jurewitz

Developer Tools & Performance Evangelist jurewitz@apple.com

Instruments Documentation

Instruments User Guide (Xcode documentation) Instruments New Features User Guide

Apple Developer Forums

http://devforums.apple.com

Related Sessions

iOS Performance and Power Optimization with Instruments	Presidio Wednesday 4:30PM
iOS Performance in Depth	Presidio Thursday 4:30PM
Objective-C Advancements In-Depth	Mission Friday 11:30AM

Labs

Mac OS X Performance Lab	Developer Tools Lab B Thursday 9:00AM
iOS App Performance Lab	Developer Tools Lab A Thursday 9:00AM
Objective-C and Automatic Reference Counting Lab	Developer Tools Lab B Thursday 2:00PM

