Debugging Memory Problems on Cray XT3 Supercomputers with TotalView Debugger

Chris Gottbrath, <u>Ariel Burton</u> TotalView Technologies

Robert Moench, Luiz DeRose





#### What is TotalView?

#### Source Code Debugger

- C, C++, Fortran 77, Fortran90, UPC
  - Complex Language Features
- Wide Compiler and Platform Support
- Multi-Threaded Debugging
- Parallel Debugging
  - MPI, PVM, Others
- Remote Debugging
- Memory Debugging capabilities
  - Integrated into the debugger
- Powerful and Easy GUI
  - Visualization
- CLI for Scripting

-	fo	rk_loopLinux			•
<u>File E</u> dit <u>View G</u> ro	oup <u>P</u> rocess	Thread Actio	on Point Too <u>l</u> s	<u>W</u> indow	<u>H</u> elp
thirds	Go Halt Ki	II Restart Ne	🔰 📢 💕 ext Step Out	Run To	
	Process 1	: fork_loopLinux pread 1 (Stonne	(Stopped) 📃		
Stack	Trace	ireau i (Stoppe	u) Stac	k Frame	
select, C++ wait_a_while, C++ snore, C++ forker, C++ fork_wrapper, C++ wain.	FP=bf FP=bf FP=bf FP=bf FP=bf FP=bf	ffeaa8 ffeaa8 ffeae8 ffeb68 ffebd8 ffec08 ffeam ffeam	nction "snore" arg: 0 bock "\$b1#\$b2": timeout: ( bock "\$b1": me: <b>N</b>	: <b>x00000000</b> struct timeva <b>x0000000 (0)</b>	1)
649     +foo = b;       650     }       651	imeval timeou hile (&timeou ose) ("Thread %lo mut)	ıt; it); d woke up in :	Snore()\n", (1	ong) (pthread_	self
		8 9 10	11 12 13	14 15 16	
<b>18</b> 19 20 21 2	2 23 <b>24</b>	25 26 27	28 29 30	31 32 33	34
35 36 37 38 3	9 40 41	<b>42</b> 43 44	45 46 47	<b>48</b> 49 50	51
52 53 <b>5</b> 4 55 5	i6 <b>57</b> 58	59 60 61	62 <b>63</b> 64	65 66 67	
CO 70 71 72 72					68
	'3 74 <b>75</b>	76 77 78	79 80 81	82 83 84	68 85
03         70         71         72         7           186         87         88         89         9           103         104         105         106         10	'3 74 75 0 91 92 07 108 109	76         77         78           93         94         95           110         111         112	79         80         81           96         97         98           113         114         115	82 83 84 99 100 101 116 117 118	68 85 102
65         70         71         72         7           86         87         88         89         9           103         104         105         106         14           120         121         122         123         15	73 74 75 00 91 92 07 108 109 24 125 126	76         77         78           93         94         95           110         111         112           127         128         129	79         80         81           96         97         98           113         114         115           130         131         132	62         63         84           99         100         101           116         117         116           133         134         135	68 85 102 119 136
85         70         71         72         7           86         87         86         89         9           103         104         105         106         11           120         121         122         123         13           137         136         139         140         14	73         74         75           00         91         92           07         108         109           24         125         126           41         142         143	76         77         78           93         94         95           110         111         112           127         128         129           144         145         146	79         80         81           96         97         98           113         114         115           130         131         132           147         148         149	82         83         84           99         100         101           116         117         116           133         134         135           150         151         152	68 85 102 119 136 136
05         70         71         72         7           86         87         86         89         9           103         104         105         106         11           120         121         122         123         12           137         138         139         140         14           154         155         156         157         15	'3         74         75           00         91         92           07         108         109           24         125         126           41         142         143           58         159         160	76         77         78           93         94         95           110         111         112           127         128         129           144         145         146           161         162         163	79         80         81           96         97         98           113         114         115           130         131         132           147         148         149           164         165         166	82         83         84           99         100         101           116         117         116           133         134         135           150         151         152           167         168         165	68 85 102 119 136 136 153 170
05         70         71         72         7           86         87         86         89         9           103         104         105         106         11           120         121         122         123         1           137         138         139         140         14           154         155         156         157         1           171         172         173         174         1	'3         74         75           0         91         92           07         108         109           24         125         126           41         142         143           58         159         160           70         106         177	76         77         78           93         94         95           110         111         112           127         128         129           144         145         146           161         162         163           178         779         180	79         80         81           96         97         98           113         114         115           130         131         132           147         148         149           164         165         166           181         182         183	82         83         84           99         100         101           116         117         118           133         134         135           150         151         152           167         168         168           184         855         168	68 85 102 119 136 136 153 153 170 187
03         70         71         72         7           86         87         86         89         9           103         104         105         106         11           120         121         122         123         11           137         136         139         140         14           154         155         156         157         13           171         172         173         174         1           168         189         90         191         14	3         74         75           00         91         92           07         106         109           24         125         126           41         142         143           56         159         160           75         176         177           92         133         193	76         77         78           93         94         95           110         111         112           127         128         129           144         145         146           161         162         163           177         179         180           195         196         197	79         80         81           96         97         96           113         114         115           130         131         132           147         146         149           164         165         166           181         162         183           198         199         204           215         216         216	82         83         84           99         100         101           116         117         116           133         134         135           150         151         152           167         168         165           164         185         186           201         202         219         221	68 85 102 119 136 136 153 170 170 187 204
05         70         71         72         7           86         87         86         89         9           103         104         105         106         11           120         121         122         123         11           137         138         139         140         14           155         156         157         13           171         172         173         174         1           188         189         90         91         14           205         206         207         208         2           222         223         224         225         2	3         7.4.         75           00         91.         92.           07         106         109           24         125.         126.           41         142.         143.           58         159.         160.           75         176.         177.           92         193.         194.           92         210.         211.	76         77         78           93         94         95           110         111         112           127         128         129           144         145         146           161         162         163           178         179         180           195         196         197           212         213         214	79         80         81           96         97         96           113         114         115           130         131         132           147         146         149           164         165         166           181         162         183           198         216         217           252         233         234	82         83         84           99         100         100           116         117         116           133         134         132           150         151         152           167         168         165           184         185         186           201         202         202           216         235         236         235	68 85 102 119 136 153 170 170 187 204 221
03         70         71         72         7           86         67         86         69         9           103         104         105         106         1           120         121         122         123         1           137         138         139         140         1           155         156         57         17         1           171         172         173         174         1           186         189         90         191         1           205         206         207         206         2           222         223         224         225         2           238         240         241         242         2	3         7.4         75           10         91         92           107         108         109           24         125         126           141         142         143           150         160         16           161         1.59         160           17         17.6         177           192         193         194           192         193         194           192         193         194           192         193         194           193         194         194	76         77         78           93         94         95           110         111         112           127         128         129           144         145         146           161         162         163           178         179         180           178         179         180           178         196         197           180         196         197           192         196         197           212         230         231           246         247         246	79         80         81           96         97         96           113         114         115           130         131         132           147         148         149           164         165         166           181         182         183           198         99         200           215         216         217           252         233         234           249         250         251	82         83         84           99         100         101           116         117         118           133         134         133           150         151         152           167         168         186           201         202         203           216         219         202           235         236         237           252         253         254	<ul> <li>66</li> <li>85</li> <li>102</li> <li>119</li> <li>136</li> <li>153</li> <li>170</li> <li>187</li> <li>204</li> <li>221</li> <li>238</li> <li>255</li> </ul>



# Supported Compilers, Distributions and Architectures

#### Platform Support

- Linux x86, x86-64, ia64, Power
- Mac Power and Intel
- Solaris Sparc and AMD64
- AIX, Tru64, IRIX
- Cray X1, XT3, IBM BGL
- Languages / Compilers
  - C/C++, Fortran, UPC, Assembly
  - Many Commercial & Open Source Compilers
- Parallel Environments
  - MPI (MPICH1 & 2, LAM, Open MPI, poe, MPT, Quadrics, MVAPICH, & many others )
  - UPC



# Message Queue Debugging

- Message Queue Graph
- Message Inspection
- Cycle detection
  - Find deadlocks

Layout	Cycle Detection	Filter	Save As
Detect Cycles	3		
Next Cycle			
Reset Cycle	Search		
Apply	Hide		Help





# TotalView Parallel Debugger Architecture for Cluster Debugging

#### Cluster Architecture

- Single Front End (TotalView)
  - GUI and debug engine
- Debugger Agents (tvdsvr)
  - Low overhead, 1 per node
  - Traces multiple rank processes
- TotalView communicates directly with tvdsvrs
  - Not using MPI
  - Optimized Protocol







#### Subset Attach

#### TotalView does not need to be attached to the entire job

- You can be attached to different subsets at different times through the run
  - You can attach to a subset, run till you see trouble and then 'fan out' to look at more processes if necessary.
- This greatly reduces overhead
  - There is a danger of missing things

[	-	Atta	ach Subset -	AttachSubsetAl	.pha.O				
	Processes to	o Attac	h To:	(2 showing,	2 filtered, 4 total)				
l	Attached	Rank	Communicator Rank	Host					
l		2	2	ralfie.etnus.com	/nfs/fs/u3/home/t				
I		3	3	ralfie.etnus.com	/nfs/fs/u3/home/t				
	4		۵II (	None					
l	-Filters		<u></u>						
l	Commu	unicato	r: All		V				
l	Talking	to Rani	c All		V				
l	Messa	зө Турс	). 🗖 Send 🗖	Secelve 📕 Une	espected				
l	📕 Array o	f Ran <u>k</u> s	: faddle						
	☐ Hall control group								
	ОК	Ī	Car	ncel	Help				







## Memory Debugging with TotalView

- Application runs with a component called the Heap Interposition Agent (HIA)
- NO source code modification
- Usually engaged automatically by TotalView
  - simple as starting the application under TotalView and enabling Memory Debugging in the GUI
  - sometimes more explicit steps are required
- Monitors the application's interactions with the Heap Manager
- Integrated with the Debugger
  - data displays annotated with information from the HIA
  - error and event notification
  - view the current state of the heap, compare with earlier state
- Low overhead



#### Enabling TotalView Memory Debugging on the Cray XT3

- Cray XT3 Compute Node executables are statically linked
  - executable must be linked with the HIA:
    - cc -g app.c
    - cc -o app app.o -Lpath -ltvheap\_xt3 -lgmalloc
- Normally a parallel job is started using the yod launcher:
  - yod -sz=256 app
- Instead, start TotalView on yod:
  - totalview yod -a -sz=256 app



# Integration with TotalView ----Pointer Annotation

# **Based on information from the HIA**

Ω×

- Shows
  - Allocated
  - Allocated Interior
  - Deallocated
  - Deallocated Interior
  - Corrupted Guard Block(s)

			addr	- main	- 1.1					
ile <u>E</u>	dit <u>∨</u> ie	w To	oo <u>l</u> s <u>V</u>	/indow				<u>H</u> elp	5 🗍	
.1	$\overline{\nabla}$								4	
xpressio	on: addr			Ad	dress:	0×bfffd1f4				
IV	pe:   int r			Valua				Details		
000400	dQ (Dam	ulin a)	. 0.400	value	(D)					
000496	du (Dani	giing) -	-> 0X00	000000	(0)					
	-			misa	addr -	main - 1	l. <b>1</b>			• □
	<u> </u>	Edit	View	Tools	Wind	w				<u>H</u> elp
	<u>[</u> 1.1		4							
	Expres	ssion:	misadd	r		Address:	0×bfffd	1f0	Data	
	· ·	Type: [	int "		v	مايام			Deta	
	0~080/	19664	(Donalir	a Intoria	v 0 _ ~ 0		1 701			—fil
	070004	13064	(Danyin	iy interio	n) - 2 0.	~00000000	, (0)			
	_									



## Memory Debugging with TotalView

- Heap Manager API Errors
- Read-before-Write --- reading uninitialized data
- Use-after-free --- dangling pointers
- Bounds Errors
- Leaks



#### Heap Manager API Errors

- HIA monitors calls to the Heap Manager
- Checks arguments and return values
- Updates its tables
- Checks for errors, e.g.:
  - Double free()
  - free() interior
  - free() unknown
  - realloc() errors
  - Invalid alignment
  - Checks guards (more later)
- Notifies TotalView

Block Ir	nformation ——		171 hutos	0-000460621			
10×00	040000	_	171 bytes	0X00040004			
Statue	Deallocated						
Flage:	none						
Flays.	none						
Block Ba	acktrace Informa	tion ——					
Select t	the desired tab	below to s	see the block allocation or d	eallocation			
Select t	the desired tab	below to s information	see the block allocation or d n may not always he availat	eallocation			
Select t backtra	the desired tab ice. Backtrace s Window to see	below to s information a the noin	see the block allocation or d n may not always be availat t at which the application str	eallocation ble. Examine the			
Select 1 backtra Process	the desired tab ice. Backtrace s Window to sei	below to s information e the poin	see the block allocation or d n may not always be availat t at which the application sto	eallocation ble. Examine the opped due to the			
Select 1 backtra Proces: event.	the desired tab ice. Backtrace s Window to see	below to s information e the poin	see the block allocation or d n may not always be availat t at which the application sto	eallocation ble. Examine the opped due to the			
Select f backtra Process event. Backtrac	the desired tab ice. Backtrace s Window to ser e	below to s information e the poin	see the block allocation or d n may not always be availal t at which the application sto	eallocation ble. Examine the opped due to the			
Select f backtra Proces: event. Backtrac	the desired tab ace. Backtrace s Window to ser e	below to s information e the poin	see the block allocation or d n may not always be availat t at which the application sto Source Information	leallocation ble. Examine the opped due to the			
Select f backtra Proces: event. Backtrac	the desired tab ace. Backtrace s Window to ser e Function	below to s information e the poin Line #	see the block allocation or d n may not always be availat t at which the application sto Source Information	leallocation ble. Examine the opped due to the			
Select f backtra Process event. Backtrac	the desired tab ace. Backtrace s Window to ser e Function	below to s information e the poin	see the block allocation or d n may not always be availat t at which the application sto Source Information	leallocation ble. Examine the opped due to the			
Select 1 backtra Process event. Backtrac	the desired tab ace. Backtrace s Window to ser e Function ree	below to s information e the poin	see the block allocation or d n may not always be availat t at which the application sto Source Information /opt/toolworks/totalview.6X.	eallocation ble. Examine the opped due to the 			
Select 1 backtra Process event. Backtrac	the desired tab ace. Backtrace s Window to ser e Function ree <b>louble_free</b>	below to s information e the poin Line # 50	see the block allocation or d n may not always be availat t at which the application sto Source Information /opt/toolworks/totalview.6X. /home/demouser/memory	eallocation ble. Examine the opped due to the .7.0-5/linux-x86 <b>/main.cxx</b>			
Select 1 backtra Proces event. Backtrac	the desired tab ace. Backtrace s Window to ser e Function function funble_free nain	below to s information e the poin Line # 50 76	see the block allocation or d n may not always be availat t at which the application sto Source Information /opt/toolworks/totalview.6X. /home/demouser/memory/m	leallocation ble. Examine the opped due to the .7.0-5/linux-x86 / <b>main.cxx</b> iain.cxx			
Select 1 backtra Proces event. Backtrac ID I T-2 fil G-2	the desired tab ace. Backtrace s Window to ser e Function function ree fouble_free nain libc_start_mai	below to s information e the poin Line # 50 76 n	see the block allocation or d n may not always be availab t at which the application sto Source Information /opt/toolworks/totalview.6X. /home/demouser/memory/m /lib/tls/libc.so.6	leallocation ble. Examine the opped due to the 			
Select 1 backtra Proces event. Backtrac	the desired tab ace. Backtrace s Window to ser e Function function <u>fouble_free</u> nain _libc_start_mai _start	below to s information e the poin Line # 50 76 n	see the block allocation or d n may not always be availab t at which the application sto Source Information /opt/toolworks/totalview.6X. /home/demouser/memory/m /lib/tls/libc.so.6 /mnt/alt3/home/demouser/me	leallocation ble. Examine the opped due to the .7.0-5/linux-x86 /main.cxx lain.cxx			



#### **Event Filtering**

#### Notification can be restricted to a set of events of interest

-	Memor	y Event Notification
<u> </u>	⊴otify me when these events trigger:	
	Event	Description
	Allocation Failed	An allocation call failed or the address returned is
	Double allocation	Allocator returned a block already in use: heap m
	Double free	Program attempted to free an already freed block
	Free interior pointer	Program attempted to free a block incorrectly, via
	Free notification	A block for which notification was requested is be
	Free unknown block	Program attempted to free an address not in the h
	Guard corruption	The guard areas around a block have been overw
	Invalid aligned allocation reques	tProgram supplied an invalid alignment argument t
	Misaligned allocation	Allocator returned a misaligned block: heap may
	Realloc notification	A block for which notification was requested is be
	Realloc unknown block	Program attempted to reallocate an address not i
	Termination notification	The target is terminating, memory analysis can be
	🗹 Unknown error	Some unknown error has occurred
	Help	OK Cancel



#### Read-before-Write --- Reading Uninitialized Data

- Program reads from a newly allocated area before initializing its contents
- Can be difficult to find because a program may have worked in the past, or appears to fail non-deterministically
- Trivial example:

```
snooker_ball_t *red = malloc ( sizeof ( *red ) );
int value = red->value;
```

```
current_score += value;
```



#### Painting

## The HIA can paint blocks on

- allocation
- deallocation

# – Paint Pattern

- defaults are unlikely values
- can be customized
- Look for pattern
- Trigger fault on dereference

-	a_red	_ball - main - 1.1	•
<u>F</u> ile <u>E</u> dit	<u>V</u> iew Too <u>l</u> s	Window	<u>H</u> elp
1.1	$\mathbf{\Sigma}$	M	
Expression:	*(a_red_ball)	Address: 0x08049b10	
<u>T</u> ype:	snooker_ball_t		Details
Field	Туре	Value	
j value	int	0xdea110cf (-559869745)	
- ×	double	-6.81916094707535e+147	
- y	double	-6.81916624944375e+147	
- spare	int	0xdea110cf (-559869745)	
colour	<string> *</string>	0xdea110cf -> <bad 0xd<="" address:="" td=""><td>ea110cf&gt;</td></bad>	ea110cf>
-			

- Intended to provoke noticeable and consistent numerical errors in arithmetic, or trigger exception
- Temporarily fix problem



#### Use-after-Free --- Dangling Pointers

- Application continues to use a block after it has been released back to the Memory Manager
- Confusion over block ownership in complex codes with many libraries
- Can be difficult to find because point failure may depend on when block is reused
- TotalView can help:
  - annotations on data displays
  - painting
  - tagging
  - hoarding



# Tagging and Hoarding

# Tagging

- tag an allocation so that when it is passed to the Heap Manager for reuse, an event is raised
- use when you know which block is being used-after-free, but don't know where the block is being freed

# Hoarding

- released blocks are not immediately passed to the Heap Manager for reuse, but retained by the HIA
- allows the application to run safely for a while after the premature deallocation



#### **Bounds Errors**

- TotalView can help find certain bounds errors by adding guard regions to allocations
  - optionally 'pre' and/or 'post' guards
  - sizes and patterns can be specified
  - alignment constraints are preserved
- Guards checked by the HIA when a block is deallocated
  - if a guard is found to be have been corrupted, an error is raised
- Full guard check can be initiated at any time from TotalView
- Choice of patterns may trigger errors earlier (ala painting)



#### Bounds errors/...

The guard areas around a block have been overwritten, suggesting a bounds error										
Block I	nformation ——									
0×08	04c068		64 bytes	0x0804c0a7						
88										
Status:	Status: Allocated									
Flags:	Flags: Operation in Progress									
	] - [	- 3								
Block Ba	acktrace Informati	on ———								
Select	the desired tab b	elow to s	ee the block allocation or o	leallocation						
backtra	ace. Backtrace in	nformation	n may not always be availa	ble.						
stoppe	d due to the ever	indow to ht	see the point at which the :	application						
Backtrar										
Dackarda										
	Function	Line #	Source Information	Ĥ						
<b>⊡</b> ⊢ <mark>4</mark>	malloc	1/19	malloc wranners dionen c							
	corrupt data	77	main.cxx							
l Hi	main	126	main.cxx							
	libc_start_main		libc.so.6							
	_start		niterapp	M						
Source			/home/demouser/mei	mory/main.cxx						
73	size = 16;									
74										
75	// Allocate some	arrays.								
76	p0 = (int *) malloc	; size * s	izeof( int ) );							
77	p1 = (int *) malloc	; c(size * s	izeof( int ) );							
78	p2 = (int *) malloc	cí size * s	izeof( int ) ):	<u>v</u>						
Point o	f Allocation P	oint of De	allocation							
<u>C</u> los	e Vie	w in Bloc	k Properties window	<u>H</u> elp						



#### Leaks

- Application deletes, or overwrites the last reference to a block before releasing the block
- Memory can no longer be accessed by the program, and cannot be reused by the Heap Manager
- Confusion over block ownership in complex codes with many libraries
- Performance loss, increase in resource usage
- TotalView can help:
  - find leaks
  - heap reports and analysis
  - heap state comparisons



#### Leak Detection

- Performed by TotalView at the request of the user
- Performs analysis similar to the first phases of a 'Mark-and-Sweep' Garbage Collector
- Conservative --- will not report anything active as a leak
- Results presented in TotalView's Heap Views:
  - Heap Graphical View
  - Heap Source View



## Heap Graphical View

-	Memory Debugging										
Ē	ile <u>E</u> dit <u>V</u> iew <u>A</u> ctions <u>T</u> ools <u>W</u> indow <u>H</u> elp										
-		_									
	Configuration Leak Detection Heap Status Memory Usage Memory Compare										
	Options										
	📕 Leaks 🔟 Guard Blocks 🔟 Baseline 🥂 🔨 🔜 🔊										
	filterapp										
	0x0804b934 - 0x08087000 (237.70KB)										
	Heap Information Backtrace/Source										
	Overall Table										
	Vverail Totais										
	Category Bytes Cou Property Value Category Bytes Count A										
	Heap 301.55KB 192 - Start A 0x08051910 Backtrace ID 13 128.00KB 256										
	PHI Allocated 79.98KB 27 HEnd Ad UXU80515001 PHI Allocated 64.00KB 128										
	Unfiltered 79 98KB 27 L L Type Leaked Laked										
	E-Deallocated 68.61KB 56 E-Pre-gu Uncorrupted E-Deallocated 0 0										
	📴 📴 Guard Blocks 0 1 💼 Post-g Uncorrupted 📴 📴 Guard Blocks 0 0										
	É E Hoarded 0 1 Filtered No D										
	Backtr 13										
		//									



# Heap Graphical View/...

	Configuration       Leak Detection       Heap Status       Memory Usage       Memory Compare         Options
	filterapp(11574)
MAAMA WWW	Memory block:         Type       Allocated         Filtered       No         Size       5.10KB         Start Address       0x0804c968         End Address       0x0804ddcf         Backtrace ID       9         Point of allocation:       File         File       sti_alloc.h         Method      default_alloc_template::_S_chunk_alloc         Line       490         Guard Blocks:       None         Backtrace       13         Backtrace       13



## Heap Source View

	40	Configu	ration   Leak Dete	ction	He	ap Stati	us Mem	ory L	Jsage   Memory C	ompare
- All		Data	Source Allocations 💸 Deal	ons	Hoar	d Optio	ons- Jeaks	s 🔲 Baseline		
inux		Proce	ss		Byte	s 🛆	Count	В	egin Address	End Address 🔺
Li		₫- <mark>filt</mark>	erapp		164	4.32KB	79	93		
X		🖻	-myClassB.cxx		15	1.50KB	77	71		
$\langle \langle \rangle$			re- <mark>myClassB::init</mark>		13	1.00KB	23	59		
-			Ē- Line 36		12	8.00KB	2	56		
3			🔂-Line 33			3.00KB		3		
· `>			Block 3			1024		1	0x08071f00	0×080
- 🎽			Block 2			1024		1	UXU8U/1at8	0.0807
- <u>F</u>				Cla	2	1024 1 50K D	5	12	UXUOUDUEIU	UXUOUJ
1			EF my Gassbmy	Ud	- 21	J.JUKD	J	12		
$\rightarrow$										
- (		Back	trace						ource	
- 5 -		ID	Function	Lin	е#	Source	Informatic			/myClassB.cxx
		┏ <mark>-12</mark>						20	2	
Σ			- malloc		149	malloc_	wrappers_	29	3 }	
h			myClassB::init		33	myClas	SB.CXX	30	)	
Ì			- myClassB::myClass	6B	10	mycias main av	SB.CXX	31	void myClassB::i	nit(void) {
- 31			- main libe stort main		19	main.cx libe co.l	×	32	2	
			nuc_stant_main			filteranı	, 1	33	B b_pp = (int **)	malloc (size * si
1			_ordit			merap	,	34	4	
$\sim$								20	for/int i=0+ isciz	orius (
4								I BL		



#### Heap View Filters

#### Filter views so that only blocks with certain properties are shown

-			Add Fil	ter					
	Filter	name: lude data matching any of the following all of the following	Evaluate	ation focus entry only acktrace entries	Share filter				
	1	Property Process/Library Name	Operator contains	Value	Remove				
		Process/Library Name Source File Name Class Name							
		Function Name Line Number Size (bytes)							
	The abov	PC //e.	evaluated in 1	the order shown, usi	ng the settings				
	To improve performance, place the condition that will remove the most entries at the top of the list.								
				<u></u> K	Cancel				



#### **Filtered Heap Graphical View**





#### Heap Comparisons

# At any point, save the state of the heap, including:

- allocated and deallocated blocks
- leaks
- guard states
- full stack backtraces and source code snippets
- Read in at a later time
  - process may have terminated
- Compare different snapshots



#### Heap Comparisons/...





## Try it Yourself!

- Kick the Tires
  - Sign up for a 15 day evaluation at http://www.totalviewtech.com

#### Get more Info

- Full Documentation available on line at http://www.totalviewtech.com
- Watch a webcast at http://www.totalviewtech.com
  - Introduction to TotalView Source Code Debugger
  - Introduction to Memory Debugging
- Contact us at info@totalviewtech.com