allinea

Leaders in parallel software development tools

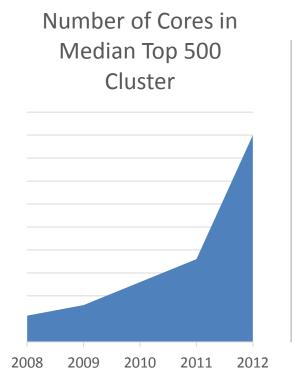
From Thousands to Millions:

Visual and System Scalability for Debugging and Profiling

Mark O'Connor VP Product Management

www.allinea.com

Introduction: Bandwidth and Complexity

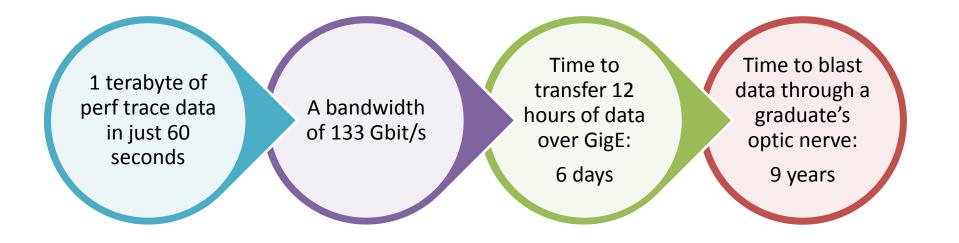


- Parallelism is increasing exponentially in HPC clusters ($R^2 \sim 0.95$)
- Performance data size and bandwidth requirements are increasing exponentially too
- More parallel execution contexts than lines of code
- Storage, networking and human visual acuity can no longer keep up



Introduction: Exploding Bandwidth

Trivial 16,000 process wave equation code





Performance Analysis Approaches

Record Everything

- Collect as much as possible and data mine it afterwards
- Use the cluster to analyse and mine large data files in parallel during analysis
- Implemented by tracebased tools such as Vampir

Statistical Analysis

- Only record data that provides:
 - Actionable information
 - Context for the above
- Example: duration of 16k MPI_Sends
 - Record the distribution shape
 - Record ranks of min / max
- Use the cluster to create small report files



Strengths and Weaknesses

Record Everything

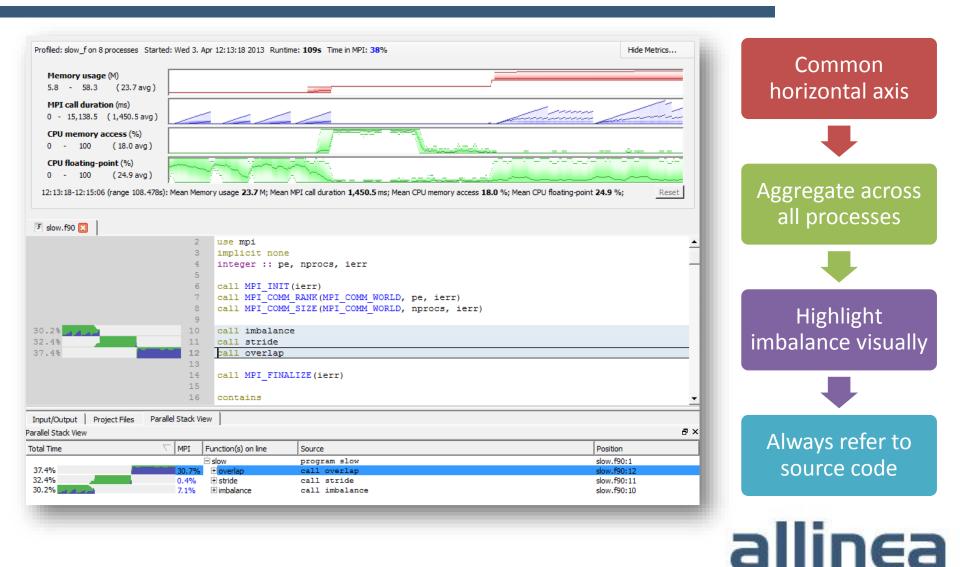
- Can sift through and analyse in extreme detail after one recording
- Shows the inner workings of communication protocols
- Extremely large trace files
- Analysis may require cluster time
- Care must be taken not to accidentally add 1000x overhead

Statistical Analysis

- Reliable performance overview with low (< 5%) overhead
- Small trace files (~20Mb)
- Simple to configure, run and interpret
- Hides the inner workings of communication protocols
- May not contain enough data to explain *why* a line or loop is slow

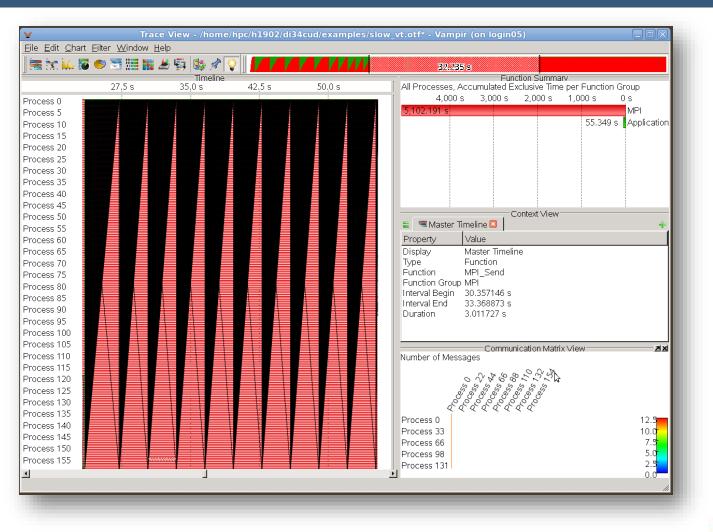
www.allinea.com

Attacking Visual Scalability



www.allinea.com

Record Everything Example



Pick one MPI call and view its specific data

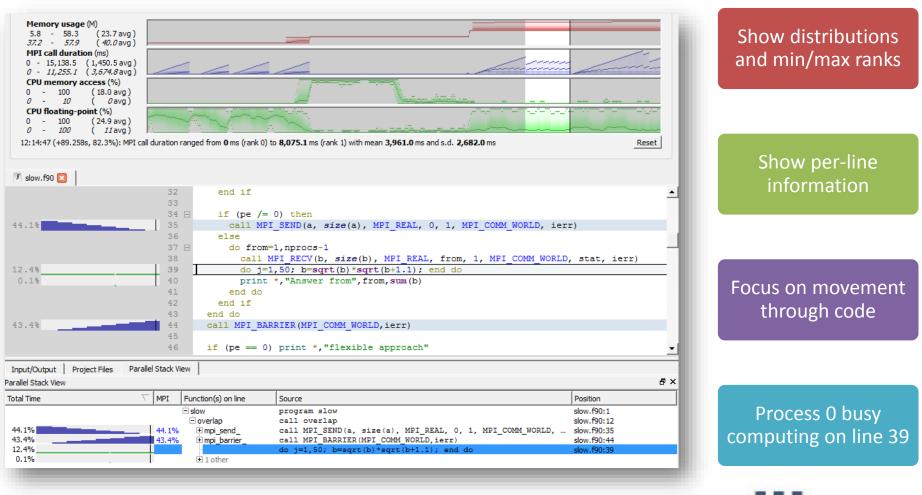
Recorded data first, source code second

Can see underlying Send + Barrier traffic

All waiting for process 0?



Statistical Analysis Example



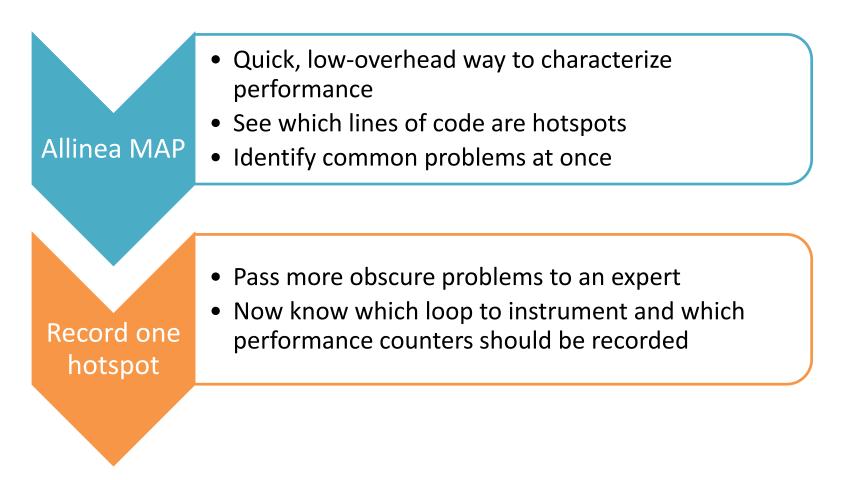
Statistical CPU Analysis

View Search Window Help			
filed: slow_f on 4 processes Start	ed: Tue Apr 9 17:12:08 2013	untime: 50s Time in MPI: 32%	Hide Metrics
CPU memory access (%)			
) - 100 (27.4 avg)			
- 100 (68 avg) CPU floating-point (%)			
- 100 (25.0 avg)			
- 50 (5 avg)	<u></u>	to the second	
PU integer (%)		the second s	estrum.
- 100 (7.5 avg) - 100 (18 avg)			
PU floating point vector (%)	and the second s		
- 100 (24.9 avg)	Frank Kann		
- 50 (5 avg)		n i serre e e e e e e e e e e e e e e e e e	
PU integer vector (%) - 100 (7.2 avg)		and the second	Philippe
- 100 (7.2 avg) - 100 (18 avg)			
PU branch (%)			
- 40 (3.0 avg)	₹		
- 40 (3 avg)			
12.20 (+18.7405, 57.0%). CF0 1	oralicit ranged from 0 % (rank t) to 10 % (rank 3) with mean 5 % and s.d. 5.0 %	Rese
ow.f90 🗶			
	12 m de i-1 2000		
10	do i=1,2000 do i=1,2000		
10 10 10	04 ⊟ do j=1,2000 x=i		
10 18 10 10	04 ⊟ do j=1,2000 x=i y=j		
10 10 10 10	04		
	04 E do j=1,2000 x=i y=j 77 a (i,j)=x*j end do end do		
	de j=1,2000 5 x=i 6 y=j 77 a(i,j)=x*j end do end do 09 end do 00 end do		
	14 do j=1,2000 15 x=i 16 y=j 17 a(i,j)=x*j 1 end do 10 end do		
	M = do j=1,2000 x=i y=j joint y=j a(1,j)=x*j end do end do end do ill ! efficient ill do l=1,500		
	14 do j=1,2000 x=i y=j 1 a(1,j)=x*j end do end do 10 end do 11 : efficient 13 : efficient 14 do j=1,2000 :		
	14 do j=1,2000 x=i x=i y=j a(1,j)=x^1j end do end do 12 ! efficient 13 do 1=1,500 44 do j=1,2000		
	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		
	$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
	$\begin{array}{ccccccc} & do & j=1,2000 \\ x \rightarrow i \\ y \rightarrow j \\ z \rightarrow i $		
	$\begin{array}{ccccc} do & j=1,2000 \\ x = i \\ y = j \\ x = i \\ y = j \\ y = j \\ end \ do \\ end \ do \\ end \ do \\ end \ do \\ i & j \\ i & j \\ i \\$		
	$\begin{array}{ccccc} do & j=1,2000 \\ x = i \\ y = j \\ x = i \\ y = j \\ y = j \\ end \ do \\ end \ do \\ end \ do \\ end \ do \\ i & j \\ i & j \\ i \\$		
1 1 1 1 1 1 1 1 1 1 1 1 1 1	14 do j=1,2000 x=i y=j y=j end do end do end do 12 ! efficient 13 do 1=1,500 44 do 1=1,500 15 do 1=1,2000 16 =1,2000 17 y=1,2000 18 =1,2000 19 end do 19 end do 19 end do 19 end do 10 end do 10 end do 10 end do 10 end do 11 end do		
t/Output Project Files Paral	do j=1,2000 x=i y=j j=1,2000 i=1,100 i=1,100 i=1,100 i=1,1,200 i=1,100 i=1,1,200 i=1,200 i=1,1,200 i=1,200 i=1,200 i=	Source	Position
t/Output Project Files Paral	do j=1,2000 x=i y=j i i=d i i=d i i=d i i=d i i=d i i=d i i=ficient i do i i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=d i end do end do end do end do end do <	program slow	slow.f90:1
t/Output Project Files Paral el Stack View Time ^ M	do j=1,2000 x=1 y=j y=j end do end do end do end do end do i efficient do i=1,500 do i=1,200 do i=1,500 do i=1,200 i end do	program slow call stride	slow.f90:1 slow.f90:11
1 1 1 1 1 1 1 1 1 1 1 1 1 1	do j=1,2000 x=i y=j i i=d i i=d i i=d i i=d i i=d i i=d i i=ficient i do i i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=d i end do end do end do end do end do <	program slow call stride a(1,j)=x*j	slow.f90:1 slow.f90:11 slow.f90:107
VOutput Project Files Paral Stack View Time ^ M 56	do j=1,2000 x=i y=j i i=d i i=d i i=d i i=d i i=d i i=d i i=ficient i do i i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=d i end do end do end do end do end do <	program alow coll atride a(1, 2)xx3 subroutine stride	slow,f90:1 slow,f90:11 slow,f90:107 slow,f90:94
VOutput Project Files Paral Stack View Fine A M 56 examples and a manual statements 56 examples and a manual statements 57 examples and a manual statements 58 examples and a manual statements 59 examples and a manual statements 59 examples and a manual statements 50 examples and a manual statement	do j=1,2000 x=i y=j i i=d i i=d i i=d i i=d i i=d i i=d i i=ficient i do i i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=j.2000 do i=d i end do end do end do end do end do <	program slow coll stride a(is)broti subrotine stride a(i,j)**j source file not found: /home/david/.allinea/wtapper/libmap-sampler-pmpi-david-sony-27811.c	slow (90:1 slow, 60:0:10 slow, 60:0:10 slow, 60:0:4 slow, 60:118 44a/uk sony-27811.c:1
1 1 1 1 1 1 1 1 1 1 1 1 1 1	do j=1,2000 x=1 y=3 y=3 end do end do end do end do j=1,2000 do i=1,500 end do end do end do end do end do end do Elsow Estride.mpi.barrier_	program slow call stride a(1,5)=x ²) subroutine stride a(1,5)=x ²)	slow.f90:1 slow.f90:11 slow.f90:107 slow.f90:94

- Able to see cache performance, floating-point or integer operations
- ... and other MPI key data

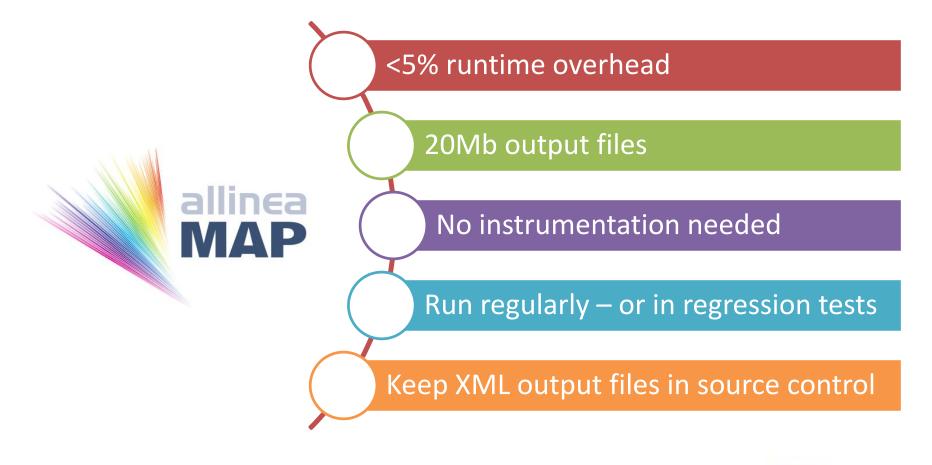


Complimentary Approaches





Surprising uses for Statistical Analysis





Success with Allinea MAP





Integrated with Allinea DDT

				-	
Input/Output	Project Files	Parallel	Stack View		
Parallel Stack V	iew				0 🕱
Total Time	*	MPI	Function(s) on line	Source	Position
38.8% 27.1% 18.0% 15.3% 0.8%			 malloc ■ _IO_str_pbackfail ■ automatic_vs_alloc 	Call allocates Allocate(a(xm,ym,levs)) Deallocate(a) Call automatic_vs_allocated Call array_notation	benchmarks.F90:1 benchmarks.F90:109 benchmarks.F90:596 benchmarks.F90:597 benchmarks.F90:110 benchmarks.F90:108
					Allinea MAP map-dev

- World-class scalability
 - Shares Allinea DDT tree architecture proven beyond Petascale
 - Data is merged on the cluster: no huge files.



Allinea DDT at Scale



Full scale on Blue Waters

- Full interactive GUI at 700,000+ processes, 30x faster than required
- "We can ramp up and down and not only pay for the largest possible case"

Full Cray Support

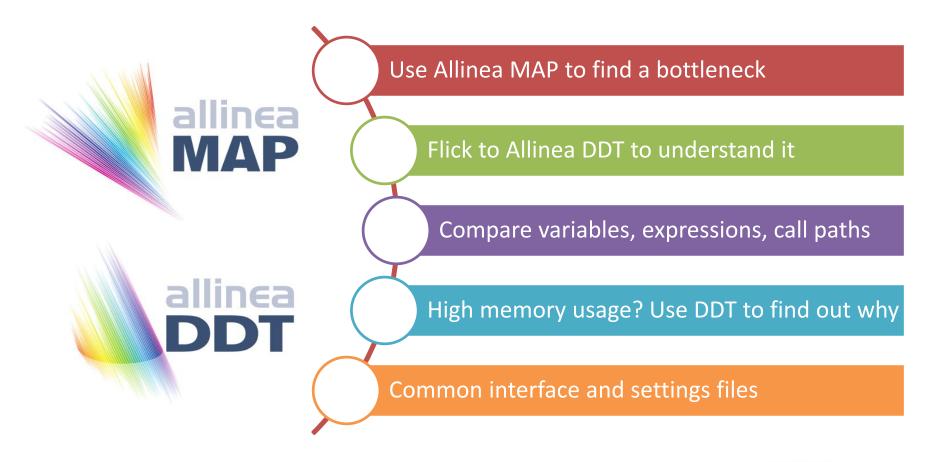
- "Allinea has proven a great partner on multiple installations"
- "Known for its scalable performance and interface"

Full scale on Titan

- "The transition has been smoother than previously thought possible"
- "DDT is tightlyintegrated into the Cray programming environment"



Unified Products: DDT + MAP





Thank-you! Any Questions?

Try Allinea MAP and Allinea DDT for yourself: www.allinea.com



	ducts		Partners	Knowledge center	News and events	Blog	About us	Contact us
Products								0
Allinea DDT	>				Allinea MAP			
Allinea MAP	>	The second second			An MPI profiler that just works,			
Overview	>	-	in a		without slo			
Features	>		Nor.					
Platforms	>	And Address of the owner	The second secon					
Free Trial	>		Bar I Ba.		Download	d a tria		MAP
Licensing	>				Domilioa	a a ma	1	MAP
Purchase	>		profile	er built on si	implicity b	autu	and tr	uet
Downloads	>							
Get support	>			ent your code or remen you would mplexec:	iber arcane compilatio	n settings	. Just compile	your code with
Demo video	>							
Trials, downloads, support	>	Simple. Allinea	a MAP shares	C datafile.in s the same batch queue data when they are dor	0	DDT, whi	ch means it w	ill happily submit
lext Steps			1					
Contact sales	>			ilts that mak	e sense wit	hout	a two-	day
Allinea DDT free tria	<	training	g cours	se				
Buy Allinea DDT	>			, Allinea MAP shows yo				
Allinea MAP free tri	al >	spent computi the run.	ng in green a	and communicating in bl	ue, it's the clearest wa	y to see w	that actually h	appened during
Buy Allinea MAP	>	ule rult.						

Built on Allinea DDT's industry-leading GUI, many of the Allinea MAP views will be familiar - the source cod



