

Installing Paraver & first steps

Germán Llort (tools@bsc.es) Barcelona Supercomputing Center



Install Paraver in your laptop



Install Paraver in your laptop

• Start Paraver:

Linux:

laptop> tar xf wxparaver-4.10.0-Linux_x86_64.tar.bz2
laptop> wxparaver-4.10.0-Linux_x86_64/bin/wxparaver

- Windows: Unzip & double-click on wxparaver-4.10.0-win/wxparaver.bat
- MAC: Unzip & double-click on wxparaver.app
- Any issue? Remotely from JUWELS (ssh -Y):

juwels> cd /p/project/training2123/tools/paraver/4.10.0
juwels> LC_ALL=C ./bin/wxparaver

Paraver
File Hints Help
Morkspaces
None
Window browser
All Traces 🔹
Files & Window Properties
 Home directory Desktop Documents Downloads Games Music News Nextcloud Pictures Private Projects Public
✓ Automatic Redraw Force Redraw

Install Paraver tutorials (Automatic download)

Download tutorial #3 – Introduction to Paraver and Dimemas methodology

		Tutorials down	Tutorials See this?
	Tutorials	Select tutorials to download and install:	
Paraver File Help Tutorials 46 Wind About	Experimental and the second seco	Paraver introduction (MPI) Dimemas introduction Introduction to Paraver and Dimema Methodology Tutorial on HydroC analysis (MPI, Dir Trace preparation	Barcelona Supercomputing Center Centro Nacional de Supercomputación
A Traces 🗘	3. In the Default directories box, change the Tuboridir root directory. 4. Save your new settings by clicking the Ok button in the Preferences Window. 5. After that, we will account cally refresh the tuborial list. 6. If rohting happens, come back here and press the index button (the first one at the submittelt) to rebuilt the tuborial int. Withe button <i>index</i> dean't seem to work (you're till reading this help!), please inty that: Every tuborial is inside its own subdirectorery. Every tuborial is inside its own subdirectorery. The button index dubring the index index in the index into the index index index in the index i	Cancel OK	1. Introduction to Analysis with Paraver - MPI 2. Introduction to the Use of Dimemas
Click on Help → Tutorials	 Lie Amorenyuserinyustorisituli. Amorenyuserinyuserinyustorisituli. Amorenyuserinyustorisituli. Amorenyustorisituli. Amorenyustori		3. Introduction to Paraver and Dimemas methodology 4. Analysis with Paraver & Dimemas - Methodology 5. HydroC Tutorial 5. HydroC Tutorial
	a = Cose		6. Paraver trace preparation

Close

Install Paraver tutorials (Manual install)

Download tutorials archive

https://tools.bsc.es

sc.es/paraver-tutorials	Home Paraver » Dimemas » Extrae Research » Documentation » Downloads Publications
	news@tools:~ > Paraver 4.7.2 avail
	Home » Documentation » Paraver tutorials
	These seven tutorials can be opened with wxParaver versions newer than 4.3.0, and you'll be able to follow the steps within the tool. To install them, download and untar the package and follow the instructions of the Help/Tutorial option on the Paraver main window. Following there is a list of available tutorials:
	Paraver introduction (MPI) Start here to familiarice with Paraver basic commands and the first steps of a performance analysis.
	Dimemas introduction The basic steps to learn how to configure and run the Dimemas simulator and to start looking at the results.
	Introduction to Paraver and Dimemas methodology This tutorial presents different ways to analyze a MPI application through well-known rules, their diagnosis and how they impact on your exploration (no traces included).
	Methodology This tutorial shows some examples of the analysis that can be done using the provided configuration files.
	Tutorial on HydroC analysis One example of performance analysis of the MPI application Hydro and further (MPI, Dimemas, CUDA) Simulations with Dimemas.
	Trace preparation Look at this tutorial to select a representative region for a large trace that cannot be loaded into memory.
	Trace alignment tutorial. If you identify some unexpected unnalignement or backwards communications, use this tutorial to learn how to correct shifts between processors.
All tutorials	wy w prefer you can download all of them togheter in a single package:
paraver-tutorials-20150526.tar.gz	.tar.gz format (127 Mb) .zip format (127 Mb)

Install Paraver tutorials (Manual install)

- Uncompress downloaded package
- Rename the folder:
 - paraver-tutorials-20150526 \rightarrow tutorials
- Open File → Preferences –

		Parave	г	-	×
File	Hints	Help			
Lo	oad Tra	ce		Ctrl+O	
Pi	revious	Traces		•	_
U	nload T	races			
Lo	oad Cor	nfiguration			_
Pi	revious	Configuration	S	•	-
Sa	ave Con	figuration			
Lo	oad Ses	sion		Ctrl+L	
Pi	revious	Sessions		•	
Sa	ave Sess	sion		Ctrl+S	
Pi	referen	ces			
Q	uit			Ctrl+Q	
Files &	Window	Properties			
	> 💼 l	ibxml2			
) 🗐 n	nemkind			
) 🗐 r	npich			
) 🗐 N	MRNet			
) 🗐 o	openmpi			
) 🔲 p	papi			
	-	Paraver			
Para	ver files	i			•
	Auto	matic Redraw		Force Redr	aw

Install Paraver tutorials (Manual method)

Trace		
Fill State gaps	with IDLE State	
View full path	in trace selector	
Maximum loadab	le trace size (MB) 500	- +
Default directori	25	
Traces	/home/gllort	Browse
CFGs	/home/gllort/Apps/Paraver/4.8.2-devel2/cfgs	Browse
Filters XML	/home/gllort/Apps/Paraver/4.8.2/share/filters-config	Browse
Tutorials root	/home/gllort/Apps/Paraver/4.8.2-devel2/tutorials	Browse
Tmp dir	/home/gllort	Browse
Behaviour		
Allow only one	e running instance	
Automatically sa	ve session every 1 – + minutes	
Show dialog fo	or crashed auto-saved sessions on startup	
Show help cor	itents on a browser	

Click on Tutorials root → Browse and point to your folder "tutorials"	
Be Paraver	Tutorials Tutorials See this? See this?
Vindo About	1. Introduction to Analysis with Paraver - MPI 2. Introduction to the Use of Dimemas

i= (~~ ~~ ~~

Close

Install CUDA workspace

Download from JUWELS & copy inside your Paraver folder:



Paraver files

Automatic Redraw

Force Redra

Cancel OK

First steps of analysis

Download sample trace from JUWELS (3 files, pcf, prv, row):

laptop> scp <USER>@juwels-booster.fz-juelich.de:/p/project/training2123/work/llort1/bschands-on/traces/lulesh2.0_booster_27p.* .



Measure the parallel efficiency

Click on "mpi_stats.cfg"

Check the Average for the column labelled "Outside MPI"

Tutorials									
The first question to answer when analyzing a parallel code is "how efficient does it	1			MPI call p	rofile @ lu	lesh2.0_bc	poster_27p	.prv	
run?". The efficiency of a parallel program can be defined based on two aspects: the parallelization efficiency and the efficiency obtained in the execution of the serial		IC ID 30 🛛 🔍 🗮	🔳 н 🙌 💵 🌶	4 [•] Σ ½ [⊾	Default 🔻	*			
regions. These two metrics would be the first checks on the proposed methodology.			Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait I	MPI_Waitall	MPI_Barrier	1PI_Reduce 📩
To measure the percelled officiency load the configuration file		THREAD 1.15.1	84.80 %	0.17 %	0.15 %	0.97 %	6.63 %	0.28 %	0.06 %
<u>cfgs/mpi/mpi_stats.cfg</u> Th s configuration pops up a table with %time that		THREAD 1.16.1	85.33 %	0.12 %	0.10 %	1.17 %	6.45 %	0.44 %	0.00 %
every thread spends in every Mi I call. Look at the global statistics at the bottom of the outside mpi column. Entry Average represents the application parallel		THREAD 1.17.1	91.74 %	0.17 %	0.16 %	0.77 %	1.50 %	0.12 %	0.27 %
efficiency, entry Avg/Max represents the global load balance and entry Maximum		THREAD 1.18.1	89.33 %	0.12 %	0.11 %	0.12 %	1.95 %	0.34 %	0.00 %
represents the communication efficiency. If any of those values are lower than		THREAD 1.19.1	92.12 %	0.08 %	0.07 %	0.09 %	1.81 %	0.01 %	0.00 %
control window to identify the phases and iterations of the code.		THREAD 1.20.1	82.52 %	0.12 %	0.10 %	0.21 %	7.21 %	0.32 %	0.00 %
		THREAD 1.21.1	80.60 %	0.09 %	0.07 %	3.20 %	3.53 %	0.38 %	0.98 %
To measure the computation time distribution load the configuration file		THREAD 1.22.1	81.81 %	0.13 %	0.10 %	0.56 %	7.10 %	0.00 %	0.00 %
histogram of the duration for the computation regions. The computation regions		THREAD 1.23.1	86.09.%	0.10 %	0.14 %	2.55.04	2.71 %	0.32 %	0.00 %
are delimited by the exit from an MPI call and the entry to the next call. If the		THREAD 1.24.1	91 99 %	0.14 %	0.09 %	4 01 %	5.07 %	0.35 %	0.00 %
histogram does not show vertical lines, it indicates the computation time may be		THREAD 1.25.1 THREAD 1.26.1	81 47 %	0.11 %	0.07 %	0.30 %	9.63 %	0.12 %	0.01 %
correlate both views.		THREAD 1.20.1	80.67 %	0.15 %	0.05 %	0.21 %	9.70 %	0.40 %	0.00 %
	Darallal officianay (Aya)		00.07 /0	0.11 /0	0.00 //	0.21 /0	5.7 6 70	0.10 //	
 To measure the computational load (instructions) distribution load the configuration file cfgs/papi/2dh_useful_instructions_cfgThis 	Parallel efficiency (Avg)	Total	2,334.46 %	3.39 %	3.06 %	31.36 %	125.25 %	8.43 %	2.89 %
configuration pops up a histogram of the instructions for the computation regions.		Averag	86.46 %	0.13 %	0.11 %	1.16 %	4.64 %	0.31 %	0.11 %
The computation regions are delimited by the exit from an MPI call and the entry	Comm efficiency (Max)	84-wimmer	95.83 %	0.24 %	0.23 %	5.09 %	9.70 %	1.02 %	0.98 %
distribution of the instructions may be not balanced. Open the control window to		Minimum	80.60 %	0.06 %	0.06 %	0.09 %	1.24 %	0.00 %	0.00 %
look at the time distribution and correlate both views.		StDev	4.22 %	0.04 %	0.04 %	1.42 %	2.63 %	0.19 %	0.25 %
• To measure the serial regions performance look at the IPC timeling looded	Load balance (Avg/Max)		0.90	0.51	0.49	0.23	0.48	0.31	0.11 ₋
		4							<u>ا</u>
E Close									

Focus on the iterative part



Focus on the iterative part



		MPI call pi	r <mark>ofile</mark> @ lu	lesh2.0_b	ooster_27p	.prv	
ie id 30 🔍 💐	🔳 н 🚧 💵 🐓	¥Σ% [►[Default 🛛 🔻	14 14			
	Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Waitall	MPI_Barrier	MPI_Reduce
THREAD 1.15.1	84.80 %	0.17 %	0.15 %	0.97 %	6.63 %	0.28 %	0.06 %
THREAD 1.16.1	85.33 %	0.12 %	0.10 %	1.17 %	6.45 %	0.44 %	0.00 %
THREAD 1.17.1	91.74 %	0.17 %	0.16 %	0.77 %	1.50 %	0.12 %	0.27 %
THREAD 1.18.1	89.33 %	0.12 %	0.11 %	0.12 %	1.95 %	0.34 %	0.00 %
THREAD 1.19.1	92.12 %	0.08 %	0.07 %	0.09 %	1.81 %	0.01 %	0.00 %
THREAD 1.20.1	82.52 %	0.12 %	0.10 %	0.21 %	7.21 %	0.32 %	0.00 %
THREAD 1.21.1	80.60 %	0.09 %	0.07 %	3.20 %	3.53 %	0.38 %	0.98 %
THREAD 1.22.1	81.81 %	0.13 %	0.10 %	0.56 %	7.10 %	0.00 %	0.00 %
THREAD 1.23.1	89.49 %	0.18 %	0.14 %	3.98 %	2.71 %	0.32 %	0.00 %
THREAD 1.24.1	86.98 %	0.14 %	0.09 %	3.55 %	3.34 %	0.35 %	0.00 %
THREAD 1.25.1	81.88 %	0.11 %	0.07 %	4.01 %	5.07 %	0.36 %	0.01 %
THREAD 1.26.1	81.47 %	0.15 %	0.09 %	0.30 %	9.63 %	0.12 %	0.01 %
THREAD 1.27.1	80.67 %	0.11 %	0.06 %	0.21 %	9.70 %	0.40 %	0.00 %
Total	2,334.46 %	3.39 %	3.06 %	31.36 %	125.25 %	8.43 %	2.89 %
Average	86.46 %	0.13 %	0.11 %	1.16 %	4.64 %	0.31 %	0.11 %
Maximum	95.83 %	0.24 %	0.23 %	5.09 %	9.70 %	1.02 %	0.98 %
Minimum	80.60 %	0.06 %	0.06 %	0.09 %	1.24 %	0.00 %	0.00 %
StDev	4.22 %	0.04 %	0.04 %	1.42 %	2.63 %	0.19 %	0.25 %
Avg/Max	0.90	0.51	0.49	0.23	0.48	0.31	0.11
4							•

Recalculate efficiency of iterative region



		MPI call p	r <mark>ofile</mark> @ lu	lesh2.0_b	ooster_27p	.prv	
IC ID 30 🔾 🔍	<mark>II</mark> H <mark>H </mark> II 🦻	¥Σ% [►[Default 🛛 🔻	4			
	Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Waitall	MPI_Barrier	MPI_Reduce
THREAD 1.15.1	84.80 %	0.17 %	0.15 %	0.97 %	6.63 %	0.28 %	0.06 %
THREAD 1.16.1	85.33 %	0.12 %	0.10 %	1.17 %	6.45 %	0.44 %	0.00 %
THREAD 1.17.1	91.74 %	0.17 %	0.16 %	0.77 %	1.50 %	0.12 %	0.27 %
THREAD 1.18.1	89.33 %	0.12 %	0.11 %	0.12 %	1.95 %	0.34 %	0.00 %
THREAD 1.19.1	92.12 %	0.08 %	0.07 %	0.09 %	1.81 %	0.01 %	0.00 %
THREAD 1.20.1	82.52 %	0.12 %	0.10 %	0.21 %	7.21 %	0.32 %	0.00 %
THREAD 1.21.1	80.60 %	0.09 %	0.07 %	3.20 %	3.53 %	0.38 %	0.98 %
THREAD 1.22.1	81.81 %	0.13 %	0.10 %	0.56 %	7.10 %	0.00 %	0.00 %
THREAD 1.23.1	89.49 %	0.18 %	0.14 %	3.98 %	2.71 %	0.32 %	0.00 %
THREAD 1.24.1	86.98 %	0.14 %	0.09 %	3.55 %	3.34 %	0.35 %	0.00 %
THREAD 1.25.1	81.88 %	0.11 %	0.07 %	4.01 %	5.07 %	0.36 %	0.01 %
THREAD 1.26.1	81.47 %	0.15 %	0.09 %	0.30 %	9.63 %	0.12 %	0.01 %
THREAD 1.27.1	80.67 %	0.11 %	0.06 %	0.21 %	9.70 %	0.40 %	0.00 %
Total	2,334.46 %	3.39 %	3.06 %	31.36 %	125.25 %	8.43 %	2.89 %
Average	86.46 %	0.13 %	0.11 %	1.16 %	4.64 %	0.31 %	0.11 %
Maximum	95.83 %	0.24 %	0.23 %	5.09 %	9.70 %	1.02 %	0.98 %
Minimum	80.60 %	0.06 %	0.06 %	0.09 %	1.24 %	0.00 %	0.00 %
StDev	4.22 %	0.04 %	0.04 %	1.42 %	2.63 %	0.19 %	0.25 %
Avg/Max	0.90	0.51	0.49	0.23	0.48	0.31	0.11

Recalculate efficiency of iterative region



Right click \rightarrow Paste \rightarrow Time

	 	ΥΣ% ΙΝΙ	Default 🔻	4			
	Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Waitall	MPI_Allreduce	MPI_Comn
THREAD 1.15.1	. 82.12 %	0.17 %	0.17 %	1.16 %	6.39 %	9.90 %	
THREAD 1.16.1	82.91 %	0.12 %	0.11 %	1.47 %	6.23 %	9.05 %	
THREAD 1.17.1	90.45 %	0.18 %	0.18 %	1.03 %	0.67 %	7.39 %	
THREAD 1.18.1	86.53 %	0.13 %	0.12 %	0.11 %	1.56 %	11.45 %	
THREAD 1.19.1	90.32 %	0.08 %	0.08 %	0.10 %	1.07 %	8.23 %	
THREAD 1.20.1	78.78 %	0.13 %	0.11 %	0.27 %	7.06 %	13.55 %	
THREAD 1.21.1	77.19 %	0.10 %	0.08 %	3.49 %	3.18 %	15.85 %	
TUDEAD 1 22 1	77.64 %	0.14 %	0.11 %	0.78 %	6.52 %	14.69 %	
THREAD 1.23.1	88.75 %	0.19 %	0.16 %	4.42 %	1.96 %	4.42 %	
THREAD 1.24.1	85.22 %	0.14 %	0.10 %	3.93 %	2.67 %	7.81 %	
THREAD 1.25.1	78.34 %	0.10 %	0.07 %	4.47 %	4.81 %	12.08 %	
THREAD 1.26.1	77.69 %	0.16 %	0.10 %	0.41 %	9.82 %	11.70 %	
THREAD 1.27.1	76.78 %	0.11 %	0.07 %	0.22 %	10.21 %	12.50 %	
Total	2 267 30 %	3 46 %	3 48 %	36.07 %	115 68 %	271.06 %	
Average	83.97 %	0.13 %	0.13 %	1.34 %	4.28 %	10.04 %	
Maximum	96.25 %	0.26 %	0.27 %	5.83 %	10.21 %	15.85 %	
Minimum	76.78 %	0.07 %	0.07 %	0.10 %	0.61 %	0.03 %	
StDev	5.30 %	0.04 %	0.05 %	1.59 %	2.86 %	3.94 %	
Avg/Max	0.87	0.50	0.47	0.23	0.42	0.63	

Efficiency of iterative region

3 numbers to quickly describe the efficiency

of your code

- Parallel efficiency \rightarrow % of time my program is computing (higher is better)
- Comm efficiency \rightarrow % of computing time of the process that communicates less (higher is bett
- Load balance \rightarrow Ratio of slow/fast processes (1 perfectly balanced)

Any value below 85% (0.85)? Pay attention!

			MPI call p	rofile @ lu	lesh2.0_b	ooster_27p	.prv	
time of the	IC ID 30 🛛 🔍 🗮	🔳 н 🙌 💵 🖄	τ Σ ½ 🕩 [Default 🛛 🔻	12			
		Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Waitall	MPI_Allreduce MP	_Comn
gner is better)	THREAD 1.15.1	82.12 %	0.17 %	0.17 %	1.16 %	6.39 %	9.90 %	
rocesses (1 is	THREAD 1.16.1	82.91 %	0.12 %	0.11 %	1.47 %	6.23 %	9.05 %	
10003303 (1 13	THREAD 1.17.1	90.45 %	0.18 %	0.18 %	1.03 %	0.67 %	7.39 %	
	THREAD 1.18.1	86.53 %	0.13 %	0.12 %	0.11 %	1.56 %	11.45 %	
	THREAD 1.19.1	90.32 %	0.08 %	0.08 %	0.10 %	1.07 %	8.23 %	
ittention!	THREAD 1.20.1	78.78 %	0.13 %	0.11 %	0.27 %	7.06 %	13.55 %	
	THREAD 1.21.1	77.19 %	0.10 %	0.08 %	3.49 %	3.18 %	15.85 %	
	THREAD 1.22.1	77.64 %	0.14 %	0.11 %	0.78 %	6.52 %	14.69 %	
	THREAD 1.23.1	88.75 %	0.19 %	0.16 %	4.42 %	1.96 %	4.42 %	
	THREAD 1.24.1	85.22 %	0.14 %	0.10 %	3.93 %	2.67 %	7.81 %	
	THREAD 1.25.1	78.34 %	0.10 %	0.07 %	4.47 %	4.81 %	12.08 %	
	THREAD 1.26.1	77.69 %	0.16 %	0.10 %	0.41 %	9.82 %	11.70 %	
	THREAD 1.27.1	76.78 %	0.11 %	0.07 %	0.22 %	10.21 %	12.50 %	
Parallel efficiency (Avg)								
	Total	2,267.30 %	3.46 %	3.48 %	36.07 %	115.68 %	271.06 %	
	Averag	83.97 %	0.13 %	0.13 %	1.34 %	4.28 %	10.04 %	
Comm efficiency (Max)	Mavinu	96.25 %	0.26 %	0.27 %	5.83 %	10.21 %	15.85 %	
	Minimum	/6./8 %	0.07 %	0.07 %	0.10 %	0.61 %	0.03 %	
	StDev	5.30 %	0.04 %	0.05 %	1.59 %	2.86 %	3.94 %	
$(\Delta v \sigma / Max)$	Acces/Mar	0.87	0.50	0.47	0.23	0.42	0.63	

Computation time distribution

Click on "2dh_usefulduration.cfg" (2nd link) → Shows time computing



2DH useful duration correlated with @ lulesh2.0_booster_27p.prv
KE ID 30 ∫ 🔍 😩 ∫ 📕 H Η III 🕂 Σ ½ ∫ ⊾ Default 🛛 🔻
Right click → Paste → Time (Focus on iterative region)

Focus on the iterative part

Click on "2dh_usefulduration.cfg" (2nd link) → Shows time computing



Computation time distribution

Click on "2dh_usefulduration.cfg" (2nd link) → Shows time computing

Duration imbalance (zigzag = some processes are taking more time than others)



Computation load distribution

■ Click on "2dh_useful_instructions.cfg" (3rd link) → Shows amount of work



Computation load distribution

■ Click on "2dh_useful_instructions.cfg" (3rd link) → Shows amount of work





Computation load distribution

• Correlate the two histograms \rightarrow Similar shapes \rightarrow Work distribution determines time computing



Computation load distribution



Where does this happen?

• Go from the table to the timeline



Where does this happen?

Go from the table to the timeline



Where does this happen?



Where does this happen?



Hints → Call stack references → Caller function

			MPI caller @	lulesh2.0_boost	er_27p.prv			
THREAD	1.1.1		3 (); {); {); {); { } } }	104 104		1111	1 1 1 1	
THREAD	1.5.1			199 6 196 1	11111		8 8:6 8	
THREAD	1.9.1				11111111	23 8 21		
THREAD	1.13.1				11111			
THREAD	1.17.1							
THREAD	1.21.1				115 MIN		1 101 10	
THBEAR	1.25.1							
THREAD	1.2/.1	9 us					2,337,47	4 us

Where does this happen?



Hints → Call stack references → Caller function



Save CFG's (method 1)



Save CFG's (method 2)





CFG's distribution

■ Paraver comes with many included CFG's → Apply any CFG to any trace!

8 🖲 🗊 Paraver File Hints Help		S Doad Configu							
Load Trace Previous Traces Unload Traces	Ctrl+O	Location:							
Load Configuration Previous Configurations Save Configuration Load Session Save Session Preferences Quit Files & Window Properties Files & Window Properties Files & Window Properties	, Ctrl+L Ctrl+S Ctrl+Q	 Places Q. Search Q. Recently Used Q gllort Desktop File System Windows Music Pictures Videos Downloads 	Name ▲ i burst_mode i i clustering i i clustering i i counters_PAPI i i CUDA i i folding iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	Size Modified 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16 18/07/16					
 Cfgs burst_mode clustering counters PAPI 		+ -	Paraver configuration	on file (*.cfg) 🛟					

CFG's distribution

■ Paraver comes with many included CFG's → Apply any CFG to any trace!

😣 🖻 🗉 🛛 Paraver		
File Hints Help		
Load Trace	Ctrl+O	
Previous Traces		
Unload Traces		
Load Configuration		
Previous Configurations	÷.	/home/gllort/Apps/wxparaver/latest/cfgs/General/views/useful_duration.cfg
Save Configuration		/home/gllort/Apps/wxparaver/4.6.2/cfgs/counters_PAPI/performance/2dh_cycles_per_us.cfg
Load Session	Ctrl+L	/home/gllort/Apps/wxparaver/4.6.2/cfgs/mpi/analysis/mpi_stats.cfg
Save Session	Ctrl+S	/home/gllort/Apps/wxparaver/latest-tutorials/3.Introduction_to_Paraver_and_Dimemas_methodology/cfgs/papi/2dh_useful_instructions.c
Df		/home/gllort/Apps/wxparaver/latest/cfgs/counters_PAPI/performance/cycles_per_us.cfg
Prererences		/home/gllort/Apps/wxparaver/4.6.2/cfgs/clustering/2dp_clusters.cfg
Quit	Ctrl+Q	$/home/gllort/Apps/wxparaver/latest-tutorials/3. Introduction_to_Paraver_and_Dimemas_methodology/cfgs/general/2dh_usefulduration.cfgs/general/2dh_usefuldurat$
		/home/gllort/Apps/wxparaver/4.6.2/cfgs/counters_PAPI/performance/2dh_usefulduration.cfg
		/home/gllort/Apps/wxparaver/4.6.2/cfgs/counters_PAPI/performance/2dh_useful_instructions.cfg
		/home/gllort/Apps/wxparaver/4.6.2/cfgs/General/sanity_checks/flushing.cfg
		/home/gllort/Apps/wxparaver/4.6.2/cfgs/counters_PAPI/performance/IPC.cfg
		/home/gllort/Apps/wxparaver/latest/cfgs/General/views/executing_cpu.cfg
		/home/gllort/Apps/wxparaver/4.6.2/cfgs/clustering/3dh_duration_cid.cfg
les & Window Properties		/home/gllort/Apps/wxparaver/latest/cfgs/clustering/clusteriD_window.cfg
3 💀		/home/gllort/Apps/wxparaver/latest-tutorials/3.introduction_to_Paraver_and_Dimemas_methodology/cfgs/mpi/mpi_stats.cfg
E tracking		/home/gllort/Apps/wxparaver/latest/cfgs/General/views/user_functions.cfg
		/home/gllort/Apps/wxparaver/cfgs/memory_location.cfg
		/home/gllort/Apps/wxparaver/cfgs/store_samples.cfg
► = 4.0.2		/home/gllort/Apps/wxparaver/cfgs/load_samples.cfg
		/home/gllort/Apps/wxparaver/cfgs/memkind_partition.cfg
E igs		
Durst_mode		
Paraver files		
raiavei files		T

Hints: a good place to start!

Suggested CFG's based on the contents of the trace





Extrae Hands-On

Germán Llort (tools@bsc.es) Barcelona Supercomputing Center



Extrae features

Platforms

- Intel, Cray, BlueGene, MIC, ARM, Android, Fujitsu Sparc, RISC-V ...
- Parallel programming models
 - MPI, OpenMP, pthreads, OmpSs, CUDA, OpenCL, Java, Python ...
- Performance Counters
 - Using PAPI interface
- Link to source code
 - Callstack at MPI routines
 - OpenMP outlined routines
 - Selected user functions
- Periodic sampling
- User events (Extrae API)



How does Extrae work?

- Symbol substitution through LD_PRELOAD
 - Specific libraries for each combination of runtimes
 - MPI
 - OpenMP
 - OpenMP+MPI
 - ...

Dynamic instrumentation

- Based on Dyninst (developed by U.Wisconsin / U.Maryland)
 - Instrumentation in memory
 - Binary rewriting
- Compiler instrumentation (-finstrument-functions)
- Static link (PMPI, Extrae API)

Recommended

BSC-Tools on JUWELS-BOOSTER

Log into JUWELS-BOOSTER:

laptop> ssh -Y <USER>@juwels-booster.fz-juelich.de

• Extrae is available under:

juwels> ls /p/project/training2123/tools/extrae/3.8.3_psmpi5.4.9-1

• ... and Paraver under:

juwels> ls /p/project/training2123/tools/paraver/4.10.0

Getting your first trace

Provided folder /p/project/training2123/work/llort1/bsc-hands-on contains:

- Sample application compiled with NVHPC + ParaStationMPI toolchain (tea_leaf)
- Jobscripts to execute and trace (run.sbatch, trace.sh)
- Configuration of the tracing tool (extrae.xml)
- Already generated tracefiles (traces/*.{pcf,prv,row})
- Copy this folder and you are ready to follow this hands-on tutorial

juwels> cp -r /p/project/training2123/work/llort1/bsc-hands-on /p/project/training2123/work/\$USER

Using Extrae in 3 steps

1. Adapt your job submission script

2. Configure what to trace

- XML configuration file
- Example configurations at \$EXTRAE_HOME/share/example

3. Run it!

• For further reference check the **Extrae User Guide**:

- <u>https://tools.bsc.es/doc/html/extrae</u>
- Also distributed with Extrae at \$EXTRAE_HOME/share/doc

Step 1: Adapt the job script to load Extrae

Example of a standard jobscript (without tracing)



Step 1: Adapt the job script to load Extrae

Jobscript modified to load Extrae

juwels> cat /p/project/training2123/work/\$USER/bsc-hands-on/extrae/run.sbatch



Step 1: Adapt the job script to load Extrae

Tracing launcher helper script

juwels> cat /p/project/training2123/work/\$USER/bsc-hands-on/extrae/trace.sh



Step 1: Which tracing library?

Choose depending on the application type

Library	Serial	MPI	OpenMP	pthread	CUDA
libseqtrace	\checkmark				
libmpitrace[f] ¹		\checkmark			
libomptrace			\checkmark		
libpttrace				\checkmark	
libcudatrace					\checkmark
libompitrace[f] ¹		\checkmark	\checkmark		
libptmpitrace[f] ¹		\checkmark		\checkmark	
libcudampitrace[f] ¹		\checkmark			\checkmark

¹ add suffix "f" if code is Fortran and default lib misses MPI activity

Step 2: Extrae XML configuration



Step 2: Extrae XML configuration (II)

juwels> cat /p/project/training2123/work/\$USER/bsc-hands-on/extrae/extrae.xml



Step 3: Run it!

Submit your job as usual

juwels> cd /p/project/training2123/work/\$USER/bsc-hands-on/extrae

juwels> sbatch run.sbatch

• Once finished (check with "squeue -u \$USER") you will have the trace in the same folder (3 files):

juwels> ls
... tealeaf_8p_4gpu_maxit6.pcf tealeaf_8p_4gpu_maxit6.prv tealeaf_8p_4gpu_maxit6.row

- Any trouble? There's a trace already generated under folder "bsc-hands-on/traces"
- Now let's look into it!

laptop> scp <USER>@juwels-booster.fz-juelich.de:/p/project/training2123/ work/\$USER/bsc-hands-on/extrae/tealeaf*.{pcf,prv,row} .



Thank you!

Germán Llort (tools@bsc.es) Barcelona Supercomputing Center



Cheatsheet: 3 main views of Paraver (I)

Timeline



Categories (e.g. MPI calls)

Cheatsheet: 3 main views of Paraver (II)

Table (Profile)

									,	
				м	PI call profi	ile @ lulesi	h2.0_27p.prv			
		I C ID 3D	Q 🔍 🔳	ны	Σ 🛠 🛯	₺ [Default 👻	4		
			Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Waitall	MPI_Barrier	MPI_Reduce	
		THREAD 1.1.1	99.04 %	0.05 %	0.06 %	0.35 %	0.30 %	0.03 %	0.00 %	
		THREAD 1.2.1	97.37 %	0.07 %	0.08 %	0.20 %	0.82 %	0.03 %	0.00 %	
		THREAD 1.3.1	93.79 %	0.05 %	0.05 %	0.22 %	0.52 %	0.03 %	0.02 %	
		THREAD 1.4.1	93.93 %	0.07 %	0.08 %	0.17 %	0.61 %	0.03 %	0.00 %	
		THREAD 1.5.1	93.75 %	0.11 %	0.11 %	0.38 %	0.19 %	0.01 %	0.00 %	
	H	THREAD 1.6.1	91.64 %	0.08 %	0.08 %	0.10 %	0.74 %	0.02 %	0.00 %	
	ĕ	THREAD 1.7.1	91.24 %	0.06 %	0.05 %	0.16 %	0.42 %	0.03 %	0.11 %	
	ă	THREAD 1.8.1	91.93 %	0.08 %	0.08 %	0.16 %	0.76 %	0.03 %	0.00 %	
-	Ľ	THREAD 1.9.1	91.20 %	0.06 %	0.05 %	0.14 %	0.59 %	0.02 %	0.50 %	
		THREAD 1.10.1	90.47 %	0.08 %	0.07 %	0.33 %	0.37 %	0.03 %	0.00 %	
	- P -	THREAD 1.11.1	89.19 %	0.12 %	0.11 %	0.35 %	0.30 %	0.01 %	0.70 %	
	σ	THREAD 1.12.1	95.80 %	0.09 %	0.07 %	0.19 %	0.83 %	0.03 %	0.00 %	
		THREAD 1.13.1	96.04 %	0.12 %	0.10 %	0.41 %	0.33 %	0.01 %	0.00 %	
	σ	THREAD 1.14.1	94.61 %	0.18 %	0.15 %	0.15 %	0.35 %	0.00 %	0.00 %	
		THREAD 1.15.1	93.28 %	0.13 %	0.10 %	0.10 %	1.21 %	0.01 %	0.00 %	
	S	THREAD 1.16.1	91.67 %	0.09 %	0.07 %	0.26 %	2.01 %	0.03 %	0.00 %	
	U	THREAD 1.17.1	93.28 %	0.13 %	0.10 %	0.11 %	1.06 %	0.01 %	0.45 %	
	S	THREAD 1.18.1	89.56 %	0.09 %	0.07 %	0.16 %	1.72 %	0.03 %	0.00 %	
	S S	THREAD 1.19.1	94.06 %	0.06 %	0.04 %	0.13 %	0.47 %	0.03 %	0.00 %	
	H	THREAD 1.20.1	89.39 %	0.10 %	0.06 %	0.25 %	1.05 %	0.03 %	0.00 %	
	ă	THREAD 1.21.1	89.62 %	0.07 %	0.04 %	0.22 %	0.30 %	0.03 %	0.90 %	
	Ľ	THREAD 1.22.1	88.08%	0.09 %	0.06 %	0.26 %	2.02 %	0.03 %	0.00 %	
	Δ	THREAD 1.23.1	98.19 %	0.14 %	0.10 %	0.16%	0.62 %	0.01 %	0.00 %	
		THREAD 1.24.1	94.10 %	0.10%	0.06 %	0.12%	1.24 %	0.02 %	0.00 %	
		THREAD 1.25.1	96.05 %	0.07%	0.04 %	0.29%	0.26 %	0.02 %	0.00%	
		THREAD 1.26.1	93.10%	0.10 %	0.06 %	0.13%	1.13 %	0.03 %	0.00 %	
	$\overline{}$	THREAD 1.27.1	94.24 %	0.08 %	0.04 %	0.18%	0.39%	0.02 %	0.00 %	
	_	Total	2,514.62 %	2.44 %	1.99 %	5.69 %	20.63 %	0.60 %	2.72 %	
		Average	93.13 %	0.09 %	0.07 %	0.21 %	0.76 %	0.02 %	0.10 %	
nary		Maximum	99.04 %	0.18 %	0.15 %	0.41 %	2.02 %	0.03 %	0.90 %	
ilai y	_	Minimum	88.08 %	0.05 %	0.04 %	0.10 %	0.19 %	0.00 %	0.00 %	
		StDev	2.79 %	0.03 %	0.03 %	0.09 %	0.51 %	0.01 %	0.24 %	
		Avg/Max	0.94	0.50	0.49	0.52	0.38	0.72	0.11	

The table can display a variety of statistics (e.g. **% of time**, # of calls, etc.) with gradient coloring showing from low values to high values

S

Cheatsheet: 3 main views of Paraver (III)

Histogram

Displays continuous metrics (e.g. **instructions executed**, duration of computations, bytes sent/received, etc.)

Gradient color represents low to high values of selected statistic (**time %**, # instances, etc.)

General tip: straight lines are good (all processes show same behavior), while variabilities usually indicate imbalances

