Virtual Institute – High Productivity Supercomputing

15th VI-HPS Tuning Workshop

Maison de la Simulation, Saclay

7-10 April 2014

15th VI-HPS Tuning Workshop (MdS)

- Presenters
  - Marc-André Hermanns (German Research School for Sim. Sci.)
  - Emmanuel Oseret & Andres Charif-Rubial (UVSQ)
  - Sameer Shende (University of Oregon PRL)
  - Alexandre Strube (Jülich Supercomputing Centre)
  - Ronny Tschüter & Matthias Weber (TU Dresden)
- Thanks
  - Local arrangements & facilities (MdS)
    - Michel Kern, Aurélie Monteiro
    - Julien Derouillat, Pierre Kestener
    - Systems: MdS, IDRIS
  - Sponsors: CEA, GENCI, PRACE

Outline

Monday 7 April

- 09:00 (early registration & set-up, individual preparation)
- 12:00-13:00 (lunch)
- 13:00-13:30 (setup)
- Welcome & introduction to VI-HPS
- Introduction to parallel performance engineering
- 15:00-15:30 (break)
- Lab setup: computer systems & software environment
- Building & running NPB-MZ-MPI/BT-MZ example code
- 17:30 (adjourn)

Outline of rest of week

Tuesday 8 April

- 09:00-10:30 **Score-P** & **CUBE**
- 11:00-12:30 **Score-P** & **ParaProf/PerfExplorer**

Wednesday 9 April

- 09:00-10:30 **Scalasca** & **Vampir**
- 11:00-12:30 **MAQAO**

Thursday 10 April

- 09:00-10:30 **TAU**
- 11:00-12:30 Conclusion:
  Using accelerators
  Engineering workflow

- Hands-on exercises part of each tool presentation every morning session
- Hands-on coaching to apply tools to analyse & tune your own codes each afternoon to 17:30

Participant survey

We'd like to know a little about you, your application(s), and your expectations and desires from this tutorial

- What programming paradigms do you use in your app(s)?
  - only MPI, only OpenMP, mixed-mode/hybrid OpenMP/MPI, ...
  - Fortran, C, C++, multi-language, ...
- What platforms/systems *must* your app(s) run well on?
  - Cray XT/XE/XK, IBM BlueGene, SGI Altix, Linux cluster™, ...
- Who's already familiar with *serial* performance analysis?
  - Which tools have you used?
    - time, print/printf, prof/gprof, VTune, ...
- Who's already familiar with *parallel* performance analysis?
  - Which tools have you used?
    - time, print/printf, prof/gprof, Periscope, Scalasca, TAU, Vampir, ...

Prepare to analyse your own application codes

- Ensure your application codes build and run to completion with appropriate datasets
  - initial configuration should ideally run in less than 15 minutes with 1-4 compute nodes (up to 64 processes/threads)
    - to facilitate rapid turnaround and quick experimentation
  - larger/longer scalability configurations are also interesting
    - turnaround may be limited due to busyness of batch queues
- Compare your application performance on other systems
  - VI-HPS tools already installed on a number of HPC systems
    - if not, ask your system administrator to install them
      (or install a personal copy yourself)

Disclaimer

Tools will ***not*** automatically make you, your applications or computer systems more *productive*.

However, they can help you understand ***how*** your parallel code executes and ***when / where*** it's necessary to work on *correctness* and *performance* issues.

In case of doubt, confusion or alarm...

DON'T PANIC!

The workshop presenters are here to assist you.

NB: On the assumption that nothing terrible is going to happen and everything's suddenly going to be alright really, all advice may be safely ignored.

Workshop system (hardware)

System	***poincare***	***curie***	
Domain	mds.cea.fr	ccc.cea.fr	
Vendor	Intel	Bull	
Network		Infiniband	
		(fat nodes)	(thin nodes)
Processors	Intel E5-2670	Intel X7560	Intel E5-2680
Frequency	2.6 GHz	2.26 GHz	2.7 GHz
Compute nodes	92	360	5040
Chips per node	2	4	2
Cores per chip	8	8	8
Threads per core	2	2	2
Memory per node	32 GB	128 GB	64 GB

Workshop system (software environment)

System	***poincare***	***curie***
domain	mds.cea.fr	ccc.cea.fr
Filesystem	*GPFS*	*Lustre*
Parallel filesys		$WORKDIR
Compiler	*Intel*	*Intel*
OpenMP flag	-openmp	-openmp
MPI	*Intel*	*Bullx*
C compiler	mpiicc	mpicc
C++ compiler	mpiicpc	mpicxx
F77 compiler	mpiifort	mpif77
F90 compiler	mpiifort	mpif90
Queue	*LoadLeveler*	*SLURM*
job submit	llsubmit job	ccc_msub job
list jobs	llq	qstat