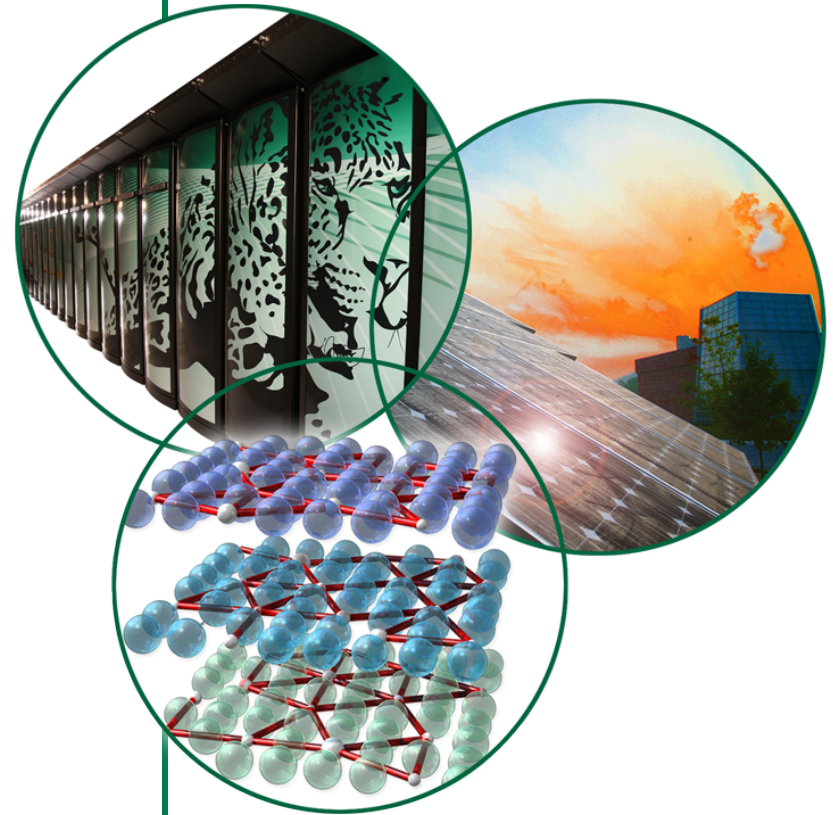


The Titan Tools Experience

Michael J. Brim, Ph.D.
Computer Science Research, CSMD/NCCS

Petascale Tools Workshop 2013

Madison, WI
July 15, 2013



Overview of Titan

- Cray XK7
 - 18,688+ compute nodes
 - 16-core AMD Opteron 6274 @ 2.2GHz
 - 32GB DDR3 RAM
 - NVIDIA Kepler K20 GPU: 14 SM with 6GB RAM
 - Gemini Interconnect: 3-D Torus



<http://www.olcf.ornl.gov/titan/>

My Roles at ORNL

- “Tools Developer” is my official job title
 - HPC debugging, performance, & system administration tools
 - Matrixed in CSMD & NCCS
 - CSMD: tools research
 - NCCS: evaluate/improve production tool offerings
- Titan (OLCF-3) Acceptance
 - responsible for testing “Programming Environment and Tools”
- OLCF-4 Tools Lead

Tools for Titan

- In production use:
 - Debugging
 - Allinea DDT, gdb, cuda-gdb, STAT, Cray ATP
 - Performance
 - Cray PAT, TAU, Vampirtrace, NVIDIA nvvp, CAPS HMPP wizard
- In testing/evaluation:
 - HPCToolkit, OpenSpeedshop, Score-P, Allinea MAP
- Allinea, CAPS, and TU-Dresden
 - prior/ongoing funding for feature improvements, mostly GPU-related
 - on-site personnel to assist users and scientific computing liaisons

Performance Tools Study

- Three goals
 1. develop familiarity with tools (as a user, not a developer)
 2. evaluate scalability and usability on Titan
 3. identify areas for improvements
- Strategy
 - follow tool use recommendations (per Titan user guide)
 - test functionality on hybrid MPI+OpenMP and MPI+GPU apps
 - evaluate usability/scalability using production science apps
- My science app friends let me down 😞
 - settled for: dummy MPI (master-worker), Sequoia IRS v1.0

Tool Configurations

- HPCToolkit 5.3.2 (svn head from June 20)
 - Profile: PAPI_L1_TCM:PAPI_TLB_TL, PAPI_TOT_CYC@50,000,000
 - Trace: PAPI_L1_TCM:PAPI_TLB_TL, Process fraction 10%
- OpenSpeedshop 2.0.2-u11
 - Profile: pcsamp
 - Trace: hwctime
- PAT 6.0.1 (perftools)
 - automatic program analysis in two phases (pat, apa)
- TAU 2.22.2-openmp
 - Profile/Trace: PAPI_L1_TCM:PAPI_TLB_TL, MPI communication tracking
- Vampirtrace 5.14.2-nogpu
 - compiler instrumentation (default on Titan), tauinst currently broken
 - 512MB trace limit per process, 128MB trace buffer
 - Profile/Trace: PAPI_L1_TCM:PAPI_TLB_TL

Tool Evaluations - Functionality

- dummy_mpi: simple master-worker MPI
 - C and C++ versions

- CUDA SDK: various GPU apps

	GNU	Intel	PGI	Cray
HPCToolkit	✓ ✓	✓ ✓	✓ ✓	✓ ✓
OpenSpeedshop	✓ □	✓ □	✓ □	✓ □
PAT	✓ ✓	✓ ✓	✓ ✓	✓ ✓
TAU	✓ □	✓ □	✓ □	x
Vampirtrace (compinst)	✓ ✓	✓ ✓	✓ ✓	✓ ✓

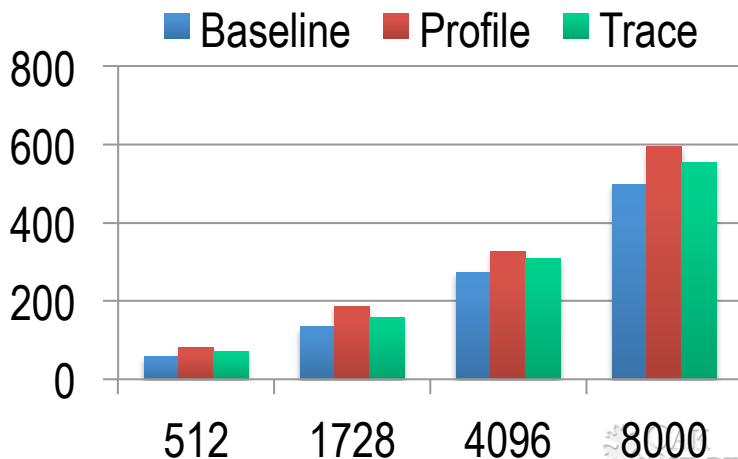
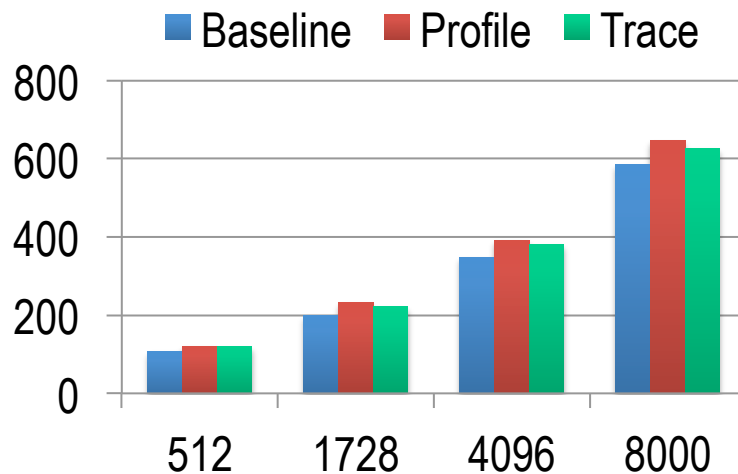
IRS Results - HPCToolkit

Storage Requirements (MiB)

PGI	512	1728	4096	8000
Profile	23	117	359	969
Trace	4	10	32	107

Cray	512	1728	4096	8000
Profile	205	1,232	4,083	11,175
Trace	13	69	240	511

Execution Overhead



IRS Results – OpenSpeedShop

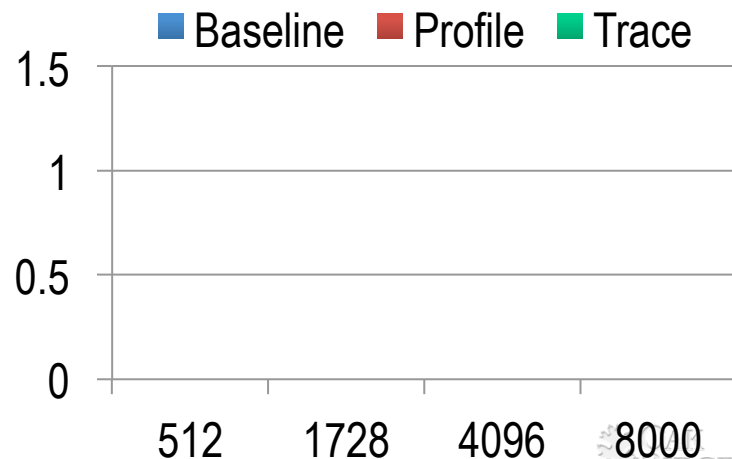
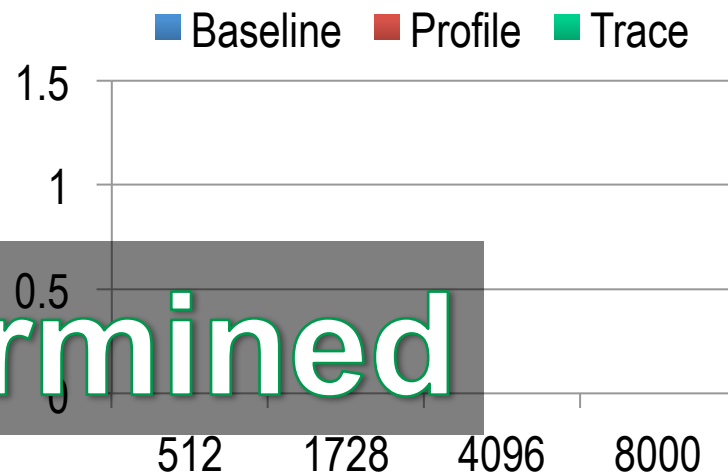
Storage Requirements (MiB)

PGI	512	1728	4096	8000
Profile				
Trace				

To Be Determined

Cray	512	1728	4096	8000
Profile				
Trace				

Execution Overhead



IRS Results – TAU

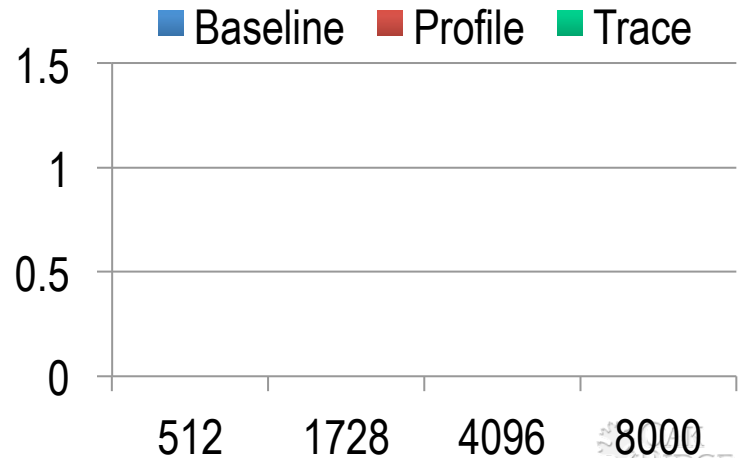
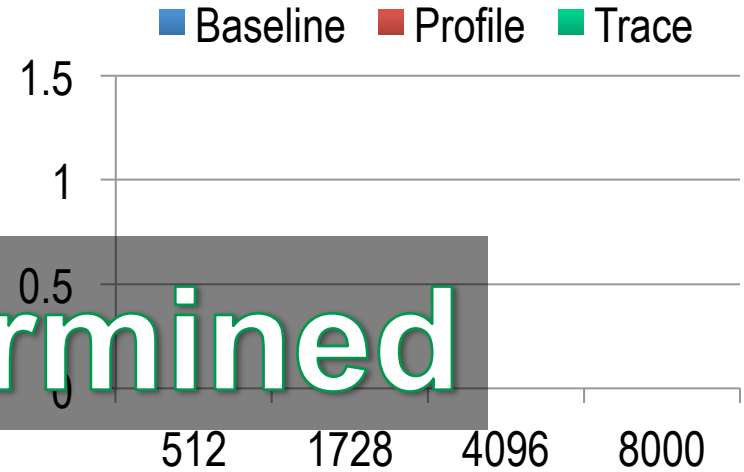
Storage Requirements (MiB)

PGI	512	1728	4096	8000
Profile				
Trace				

To Be Determined

Cray	512	1728	4096	8000
Profile				
Trace				

Execution Overhead



IRS Results – Cray PAT

Storage Requirements (MiB)

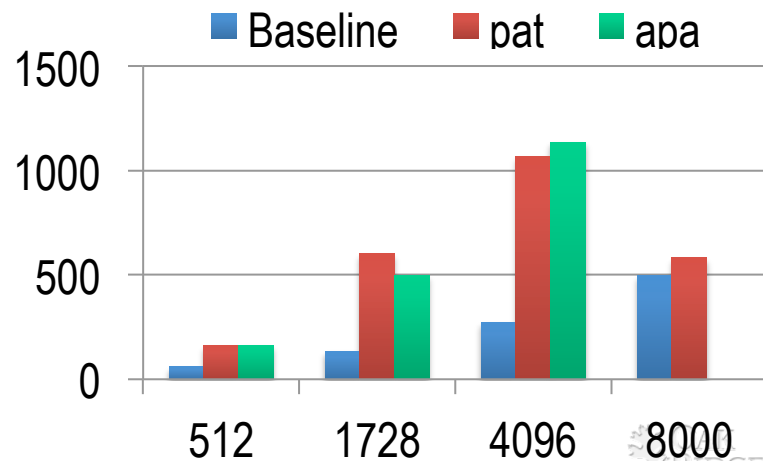
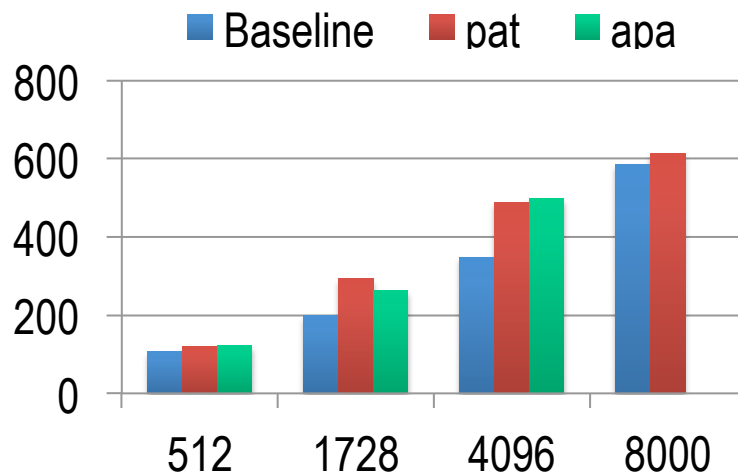
PGI	512	1728	4096	8000
pat	?	?	?	49
apa	8	27	68	TBD

script error

Cray	512	1728	4096	8000
pat	?	?	?	206
apa	46	159	377	TBD

script error

Execution Overhead



IRS Results - Vampirtrace

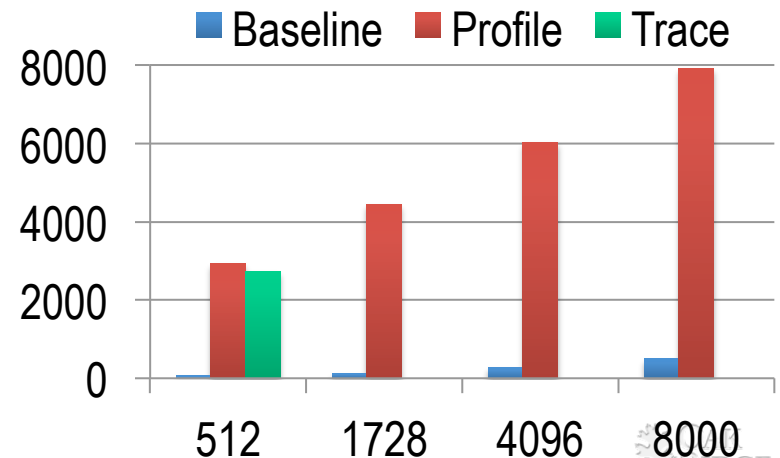
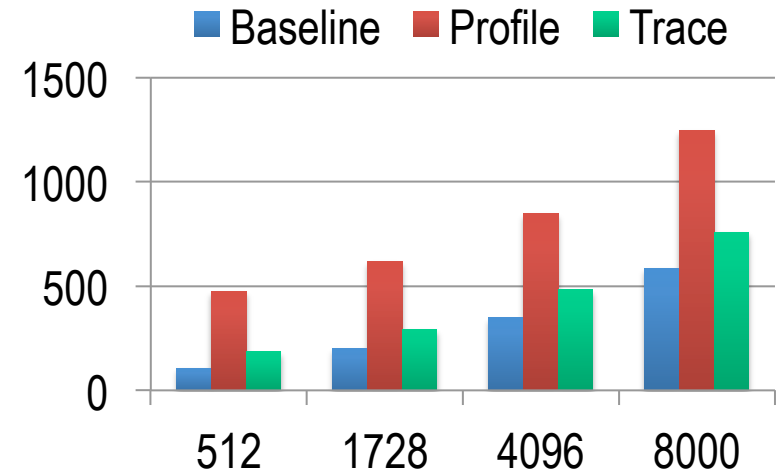
Storage Requirements (MiB)

PGI	512	1728	4096	8000
Profile	0.3	1.0	2.3	4.3
Trace	1,400	4,400	11,000	20,000

Cray	512	1728	4096	8000
Profile	2.3	7.5	18	35
Trace	1,200	TBD	TBD	TBD

> 3hr > 4hr > 5hr

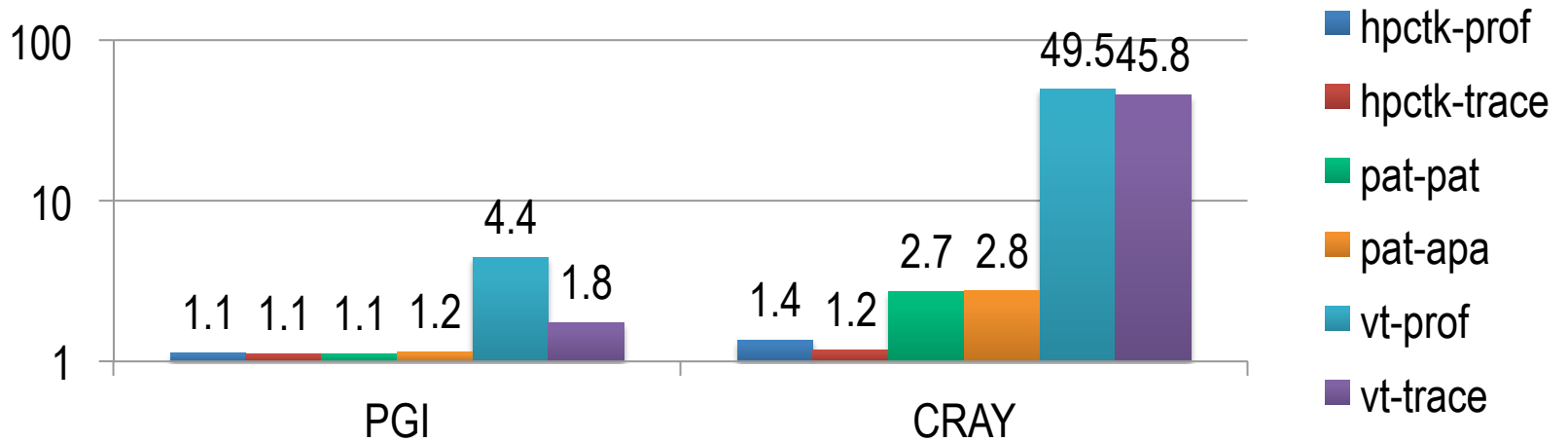
Execution Overhead



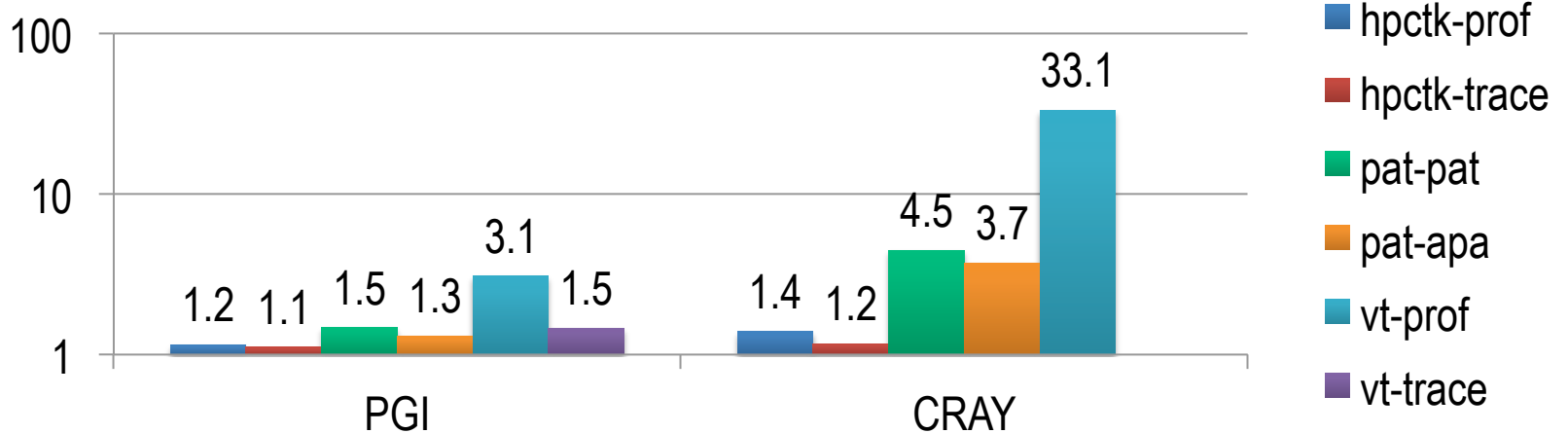
IRS Results – Comparing Tools

Normalized Execution Overhead

512



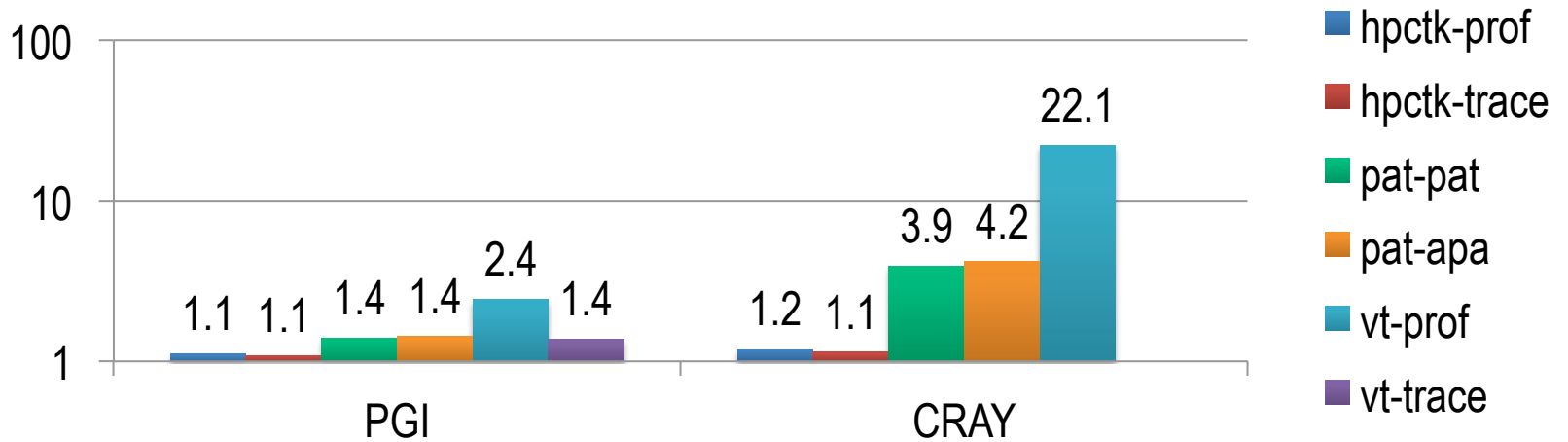
1728



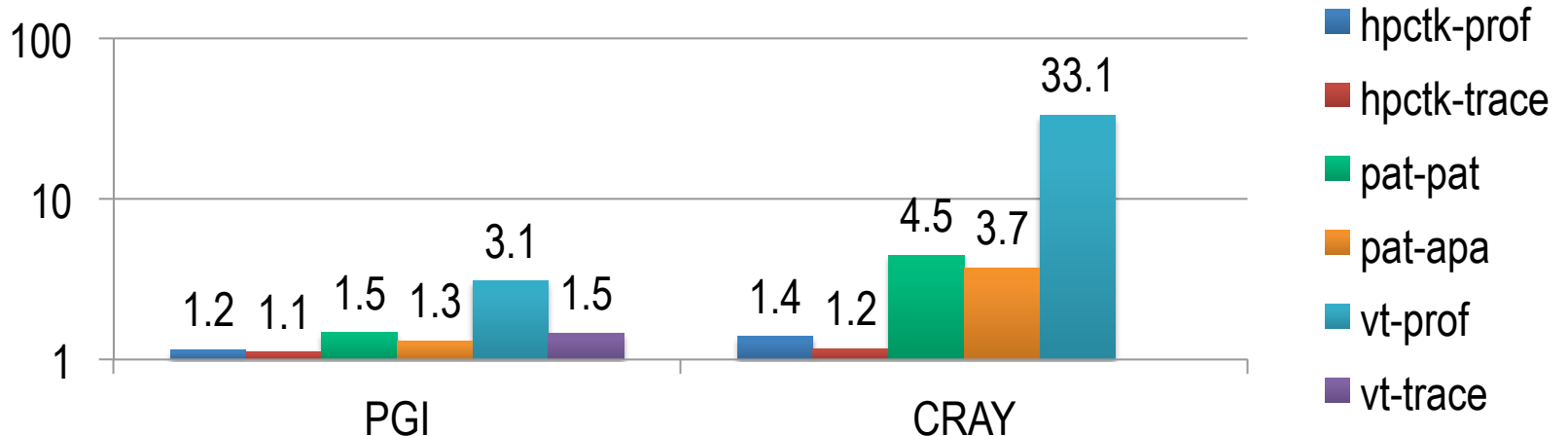
IRS Results – Comparing Tools

Normalized Execution Overhead

4096



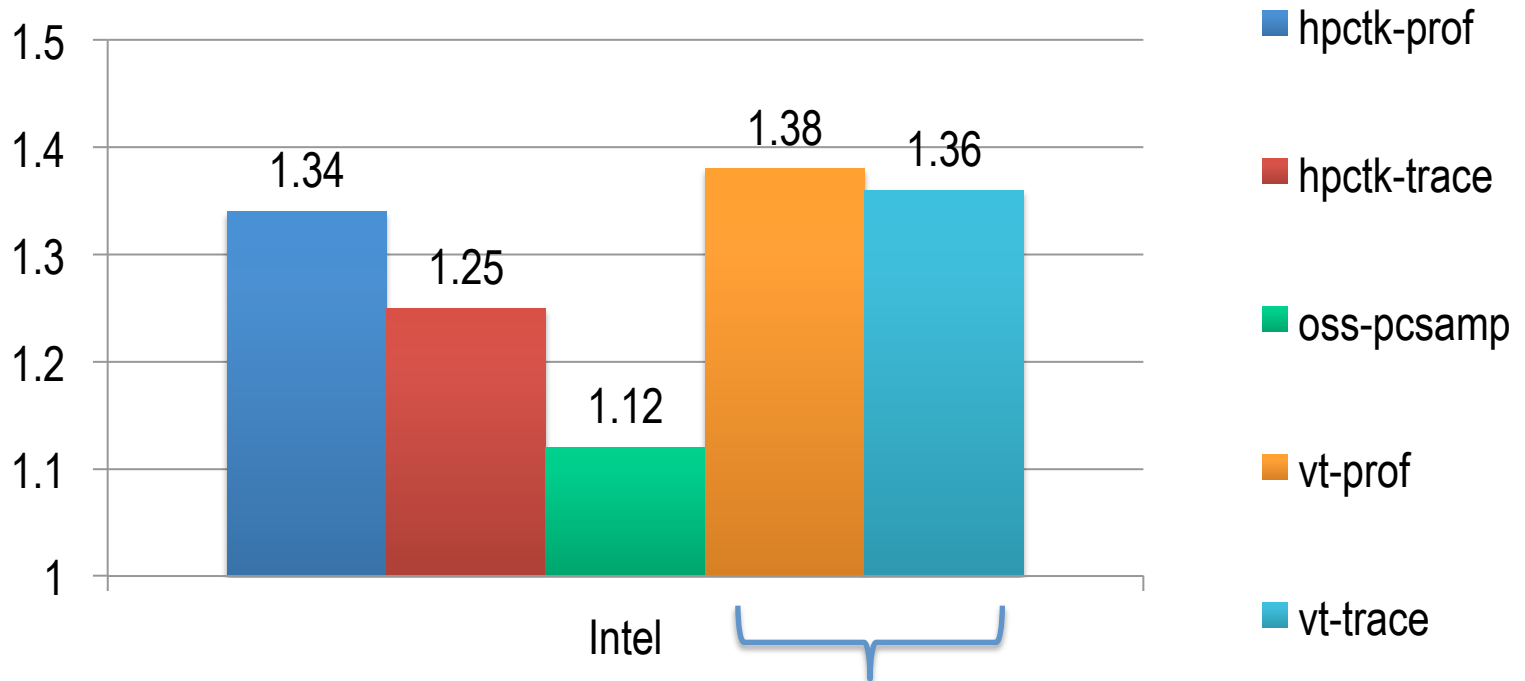
8000



dummy_mpi Results – Some Tools

Normalized Execution Overhead

Baseline time: 2,812 seconds (~3/4 hr)



Next Steps

- Work with production codes
 - LAMMPS : C++, MPI + CUDA
 - NUCCOR-J, CESM: Fortran, MPI + OpenMP
 - S3D : Fortran, MPI + OpenACC
- Compare information collected across tools
 - user/developer feedback on new insights gleaned (if any)
 - tool expert feedback
- Large-scale tests
 - at least half of Titan nodes
 - up until things break or I run out of allocation

Next Steps – Part 2

- Identify areas for tool improvements
- Work with tool developers
 - user guidance
 - scalability, new feature development

Questions & Feedback

This research used resources of the Oak Ridge Leadership Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

