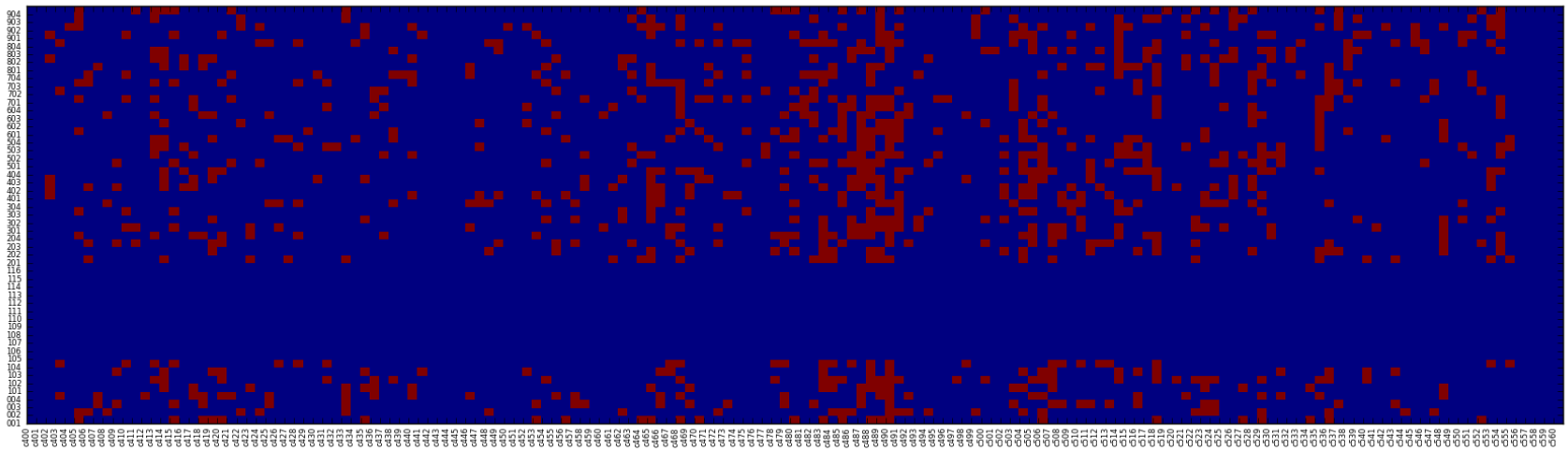
TACC Stats: A Comprehensive and Transparent Resource Usage Monitoring Tool for HPC Systems



Texas Advanced Computing Center

R. Todd Evans, Bill Barth, James Browne

TACC Stats is NOT a Line-level Performance Analysis Tool

TACC Stats . . .

- Is NOT like: TAU, IPM, gprof, Vampir, Perfexpert, Intel VTune etc.
- Does NOT provide profiling at the level of function calls, control structures, or statements
- Does NOT require instrumentation
- Does NOT need to be invoked by the user or system managers

TACC Stats IS a Transparent Resource Usage Monitoring Tool

TACC Stats . . .

- Monitors all jobs on all nodes automatically
- Collects hardware counters: Flops, MBW, cache hit rates . . .
- Linux/IB/Lustre stats: I/O, memory usage, CPU usage . . .
- Tests jobs automatically for inefficient resource utilization
- Uses a SQL database w/ Web Interface
- Is Open Source: `https://github.com/TACC/tacc_stats`
- TACC Stats + XALT provides
  libraries, modules, threads, executables . . .

TACC Stats Operation

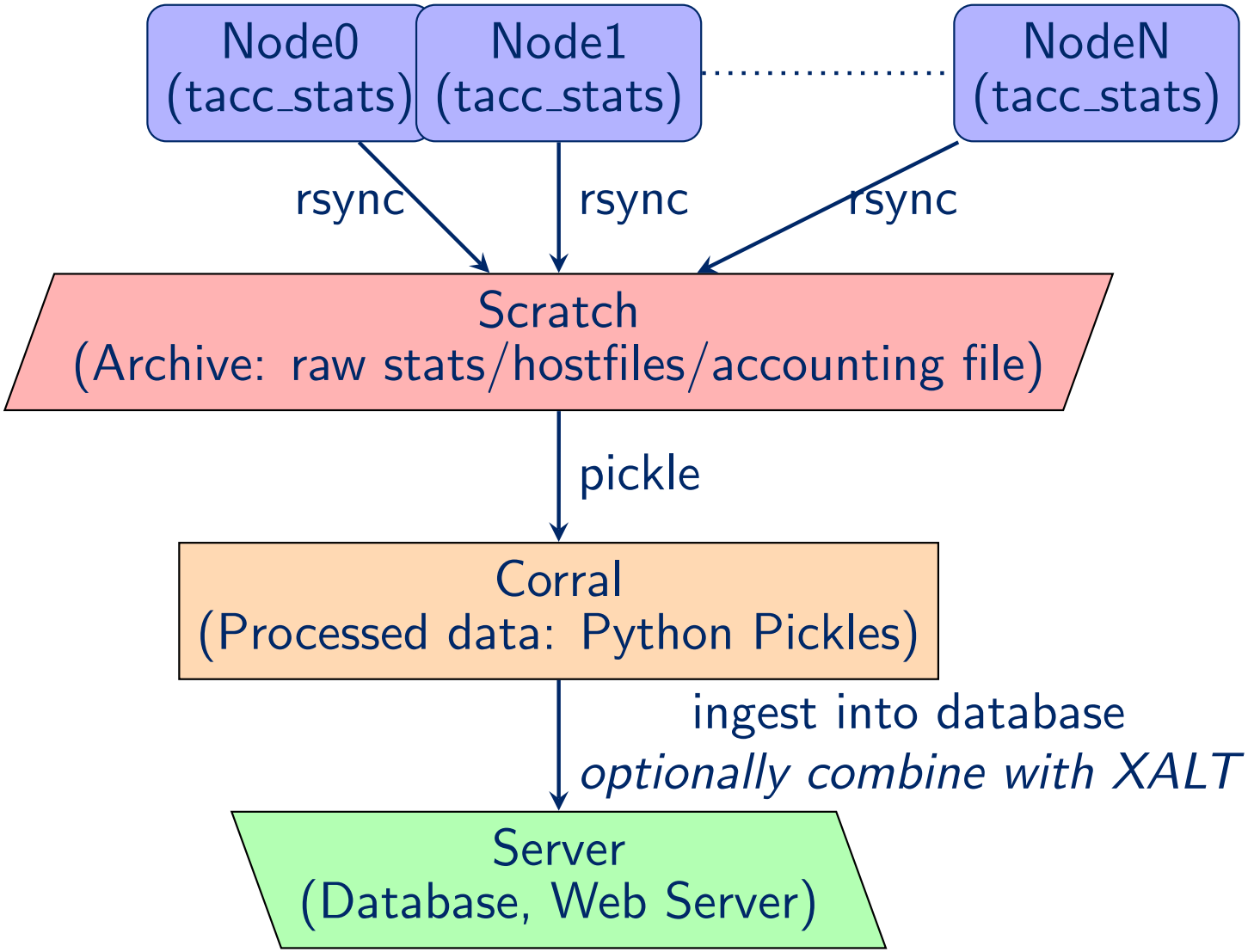
Deployment

- Distributed as a Python package
- Light-weight component (raw C) runs on compute nodes
- Starts automatically after install

Automated Operation

- Measurements for all jobs & nodes in prolog/epilog + every 10 minutes $\rightarrow$ at least two measurements per job
- `rsync`'d once a day at random times between 2 and 4 am.
- pickled once per day
- ingested into database once per day
- tests run once per day
- Every Production Job is subjected to Daily Tests

TACC Stats Operation



Data Collected

Summary: This data is collected for every job

- I/O: Block/Lustre/NFS/GigE
- CPU utilization
- IB stats
- Chip stats - "on-core" and "off-core" - instructions, cache hit rates, flops (SSE scalar/SSE vector/256), memory reads/writes ...
- Memory, Virtual Memory, NUMA data
- 150KB per host per day - Stampede 0.8GB/day

Data Computed

Metrics computed - averaged over time and hosts/cores

- CPU Usage
- Cycles per Instruction (CPI)
- Cycles per L1D replacement (CPLD)
- Load L1Hits
- Load L2 Hits
- Load LLC Hits
- All Load Operations
- Memory Bandwidth
- Memory Usage High Water Mark
- Infiniband Packet Rate
- Infiniband Packet Size
- FLOPS
- Vectorization Fraction
- Ethernet Bandwidth

Tests

We flag jobs for the following every day

- High CPI
- High Memory Usage on normal nodes
- Low memory usage on 1 TB nodes (mean is $\sim$120 GB)
- Low vectorization percentage VecPercent - often indicates user is not using AVX
- MPI over GigE network instead of IB
- Will soon test Lustre usage and MDS load
- Inefficiently using their reserved nodes (to be continued) ...

Case Studies - Many SUs & CPI

SU = Service Unit

Node hours

MILC code

- One project using MILC found to be running higher than expected CPI (1.1 vs 0.7)
- Members were not using available vectorized intrinsics
- 11% reduction in runtime

DNS Turbulence Code

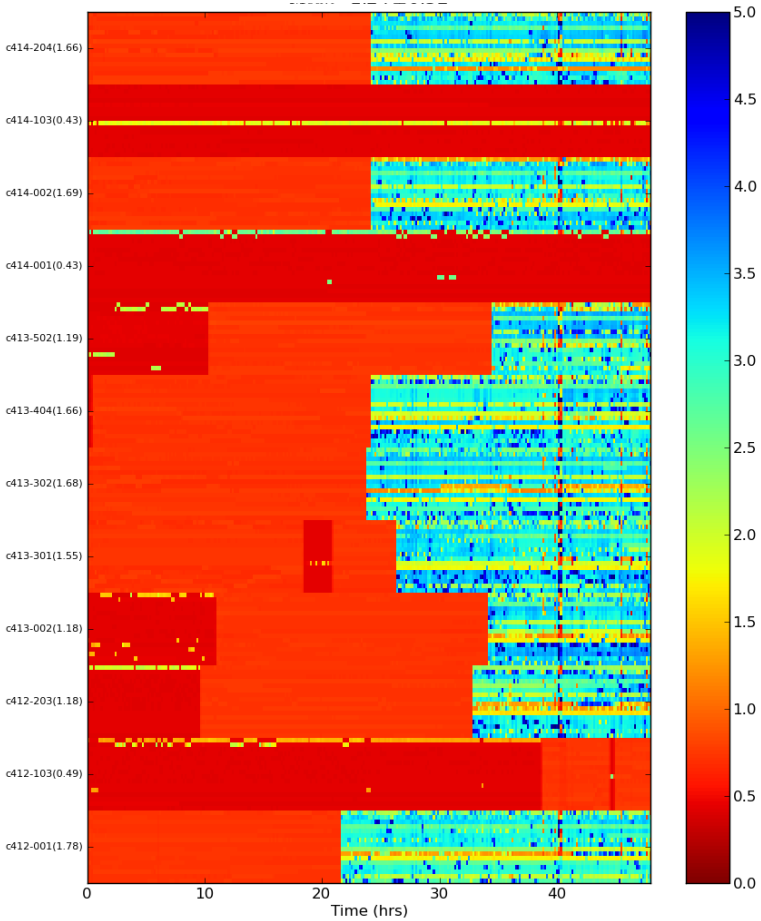
- CPI of 1.1 w/ lots of SUs
- Line-level profiling revealed MPI/OpenMP hot spots
- Converted OpenMP workshare block to parallel do block
- Improvement: 7% overall, 10% in main loop, 76% in code block

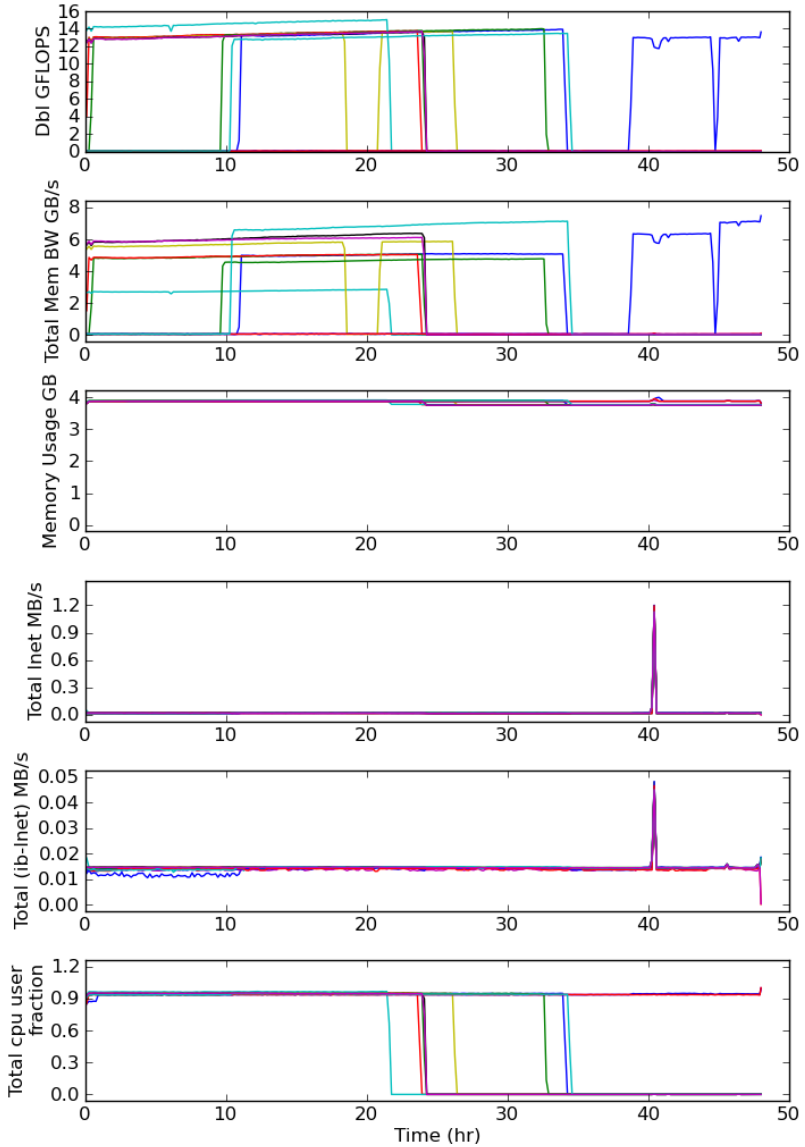
Case Studies - CPI

GR application - Singularity

- CPI of 1.15 w/ lots of SUs: 239,631 for 1st quarter 2014
- Code was making many extraneous calls to `cat` and `rm`
- Code was not using any optimization flags (-O3 or -xhost)
- 26% decrease in run time after simple changes made

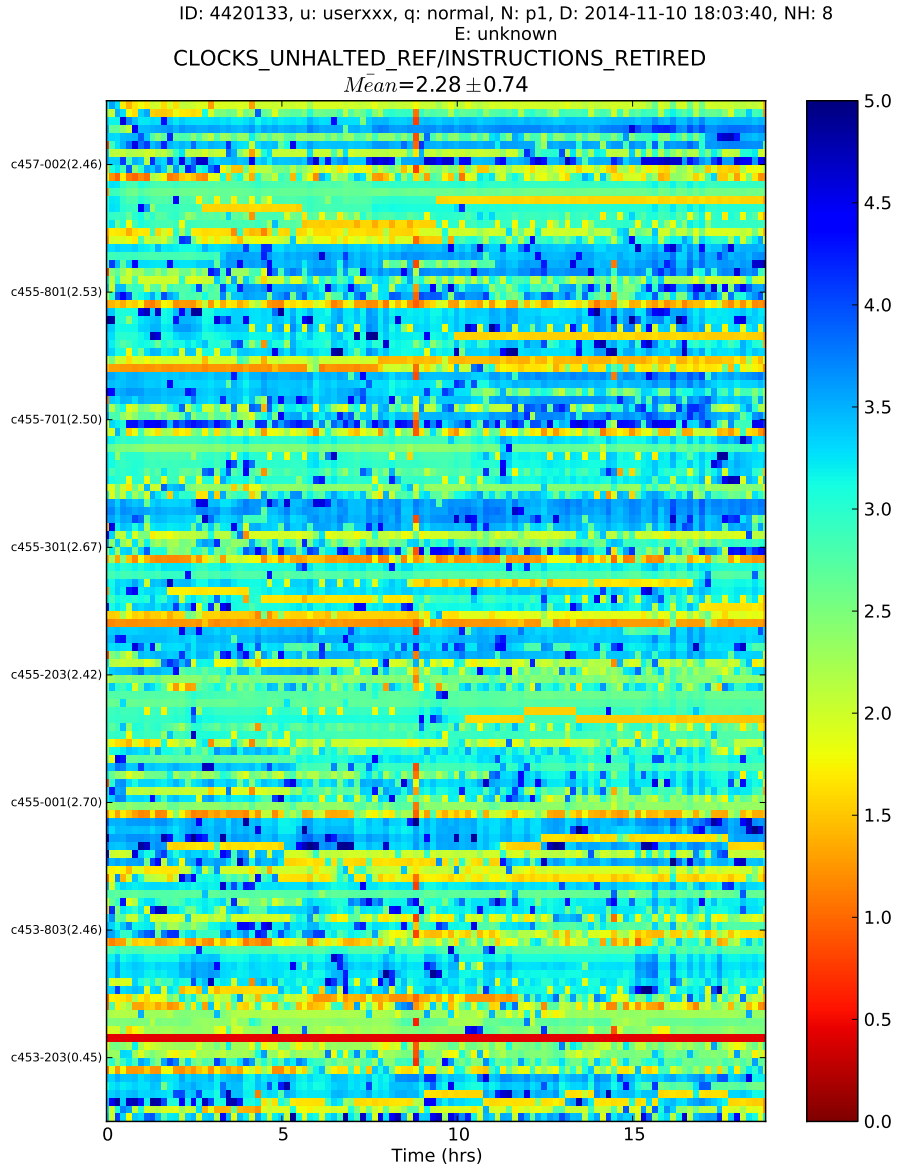
Case Studies - Performance Drop





Case Studies - Idle nodes

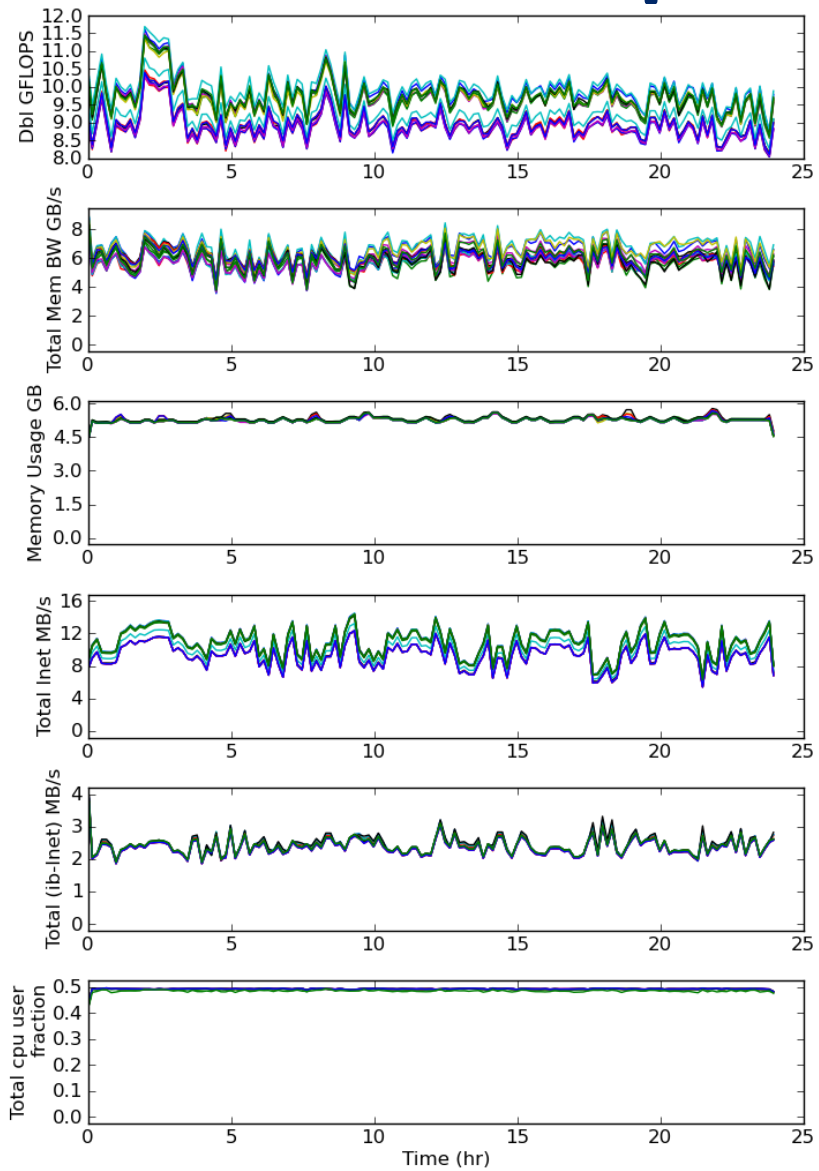
- Roughly $\sim$ 1% of SUs spend on production jobs w/ idle hosts



Case Studies - Undersubscription

Test if MPI/OpenMP

- Alerts us to users using less tasks per node than they should
- CPU Usage is 50%, memory usage is <20%
- Halve nodes for job, double MPI wayness



Web Interface

Backed by SQL Database

- Combined queries are supported in search fields
- Search by jobid, date, user, executable, project, nodes, run time, metrics ...
- Data is ingested from accounting file, XALT, processed stats data
- Raw time-series data is served and plots are generated on the fly
- Linked into Splunk logs - link takes user to logs filtered by host and time range for a given job.
- Used for day to day consultant activities typically for diagnosing issues experienced by our users

Web Interface

11/18/2014 TACC Stats Stampede

TACC Stats Stampede

Search Using JOBID:

Search

Search Using Combined Fields:

Start Date mm/dd/yyyy End Date mm/dd/yyyy

UID
User Name
Project
Executable
Search Field Value

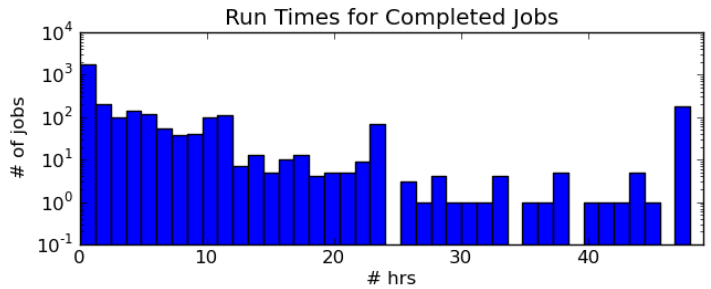
Search

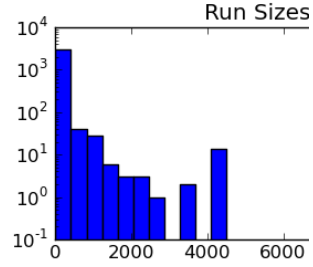
2013 / 09	17	18	19	20	21	22	23	24	25	26	27	28	29	30															
2013 / 10	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2013 / 11	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
2013 / 12	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2014	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

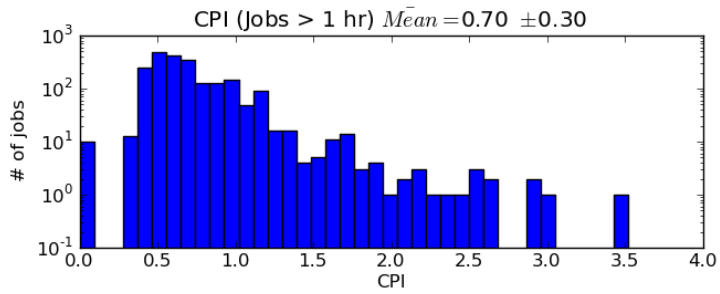
Web Interface

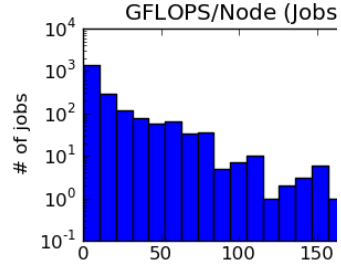
11/18/2014 TACC Stats

[date=2014-11-17]-search









Jobs over 1 mn in run time = 3626

Job Listing

Jobs flagged for:

- GigE BW > 1.049e+06 B

lagarder	4457699(1.612e+06)			
iplant	4457846(5.406e+06)	4456558(2.384e+06)		
unknown	4456785(2.247e+06)	4453596(1.806e+06)	4453592(1.768e+06)	4453591(1.612e+06)

Breakdown of MPI/OpenMP Usage

Q1 2015

Type	% SUs
Serial	22
Hybrid	19
Pure OpenMP	0.2
Pure MPI	59

Hybrid Jobs (MPI&OpenMP) Tasks/Threads Breakdown ($>$ 500,000 SUs March 2014 - March 2015)

Tasks/node	Threads/task	CPI	CPU Usage	MemUsage (GB)	SUs
16	1	0.54(6)	1580(17)	7(5)	14,000,000
2	8	0.76(9)	1130(450)	13(7)	3,400,000
1	16	0.78(13)	532(264)	17(6)	1,200,000
4	4	0.69(10)	1150(335)	9(6)	580,000
4	1	0.65(18)	437(305)	9(3)	820,000
8	1	0.65(21)	731(133)	11(6)	850,000
1	1	0.59(4)	198(297)	7(1)	540,000

Top Applications by SUs: March 2014-March2015

Production runs: Jobs/SUs = 983,675/548,368,901

Top 10 Executables

Application	Executable	SUs	Memory Usage (GB)
NAMD	namd2	59,423,911	6(3)
GROMACS	mdrun_mpi	18,668,824	4.0(2)
LAMMPS	lmp_stampede	13,119,414	4.2(9)
VASP	vasp_std	11,639,474	10(5)
FLASH	flash4	9,313,134	14(7)
WRF	wrf.exe	8,973,222	10(7)
CHARMM	charmm	8,507,615	6(1)
VASP	vasp	8,452,074	9(5)
KORAL	ko	7,862,293	6(3)
ENZO	enzo.exe	7,691,763	21(6)

MPI comparisons of Applications

NAMD - MPI comparison

36,812 Jobs from March 2014 - March 2015

37,000,000 SUs mvapich2 74,000 SUs impi (0.2%)

metric	mvapich2	impi
cpi	0.49(2)	0.49(8)
cpld	25(12)	31(3)
mem bw	0.12(3)	0.09(1)
L1 reuse	0.88(5)	0.93(1)
GFLOPs	17(6)	16(6)
Vec Fraction	0.05(1)	0.062(5)

Scaling of Applications

NAMD Jobs from March 2014 - March 2015

NAMD Jobs over 1 hour

Nodes	GLOPS/node
2	14(7)
4	23(5)
8	21(5)
16	20(8)
32	17(3)
64	15(5)
128	14(6)
256	19(5)

Work in Progress

- Build a basis of metrics on which applications can be mapped.
- User/project performance can be compared to expected performance
- Validating/interpreting raw data
- Real-time, continual stats updates via the network: syslog or RabbitMQ (available but not deployed)
- Deployment at additional Centers - TACC (Lonestar, Stampede, Wrangler)/SDSC(Gordon, Comet)/LSU(SuperMIC)/Purdue(Conte,others???)
- Adding support for additional architectures and accelerators: MIC, GPU
- Data is being incorporated into XDMoD (XD Metrics on Demand Portal, a comprehensive HPC management tool)

Acknowledgements

- NSF-funded STCI project ACI-1203604
- Robert McLay (TACC), Mark Fahey (NICS) for XALT
- University at Buffalo, SUNY Robert Deleon, Thomas Furlani, Steven Gallo, Matthew Jones, Abani Patra