

Memory Intensive Computing On Stampede

Matthew Vaughn, PhD

Manager, Life Sciences Computing

Jul 15, 2013

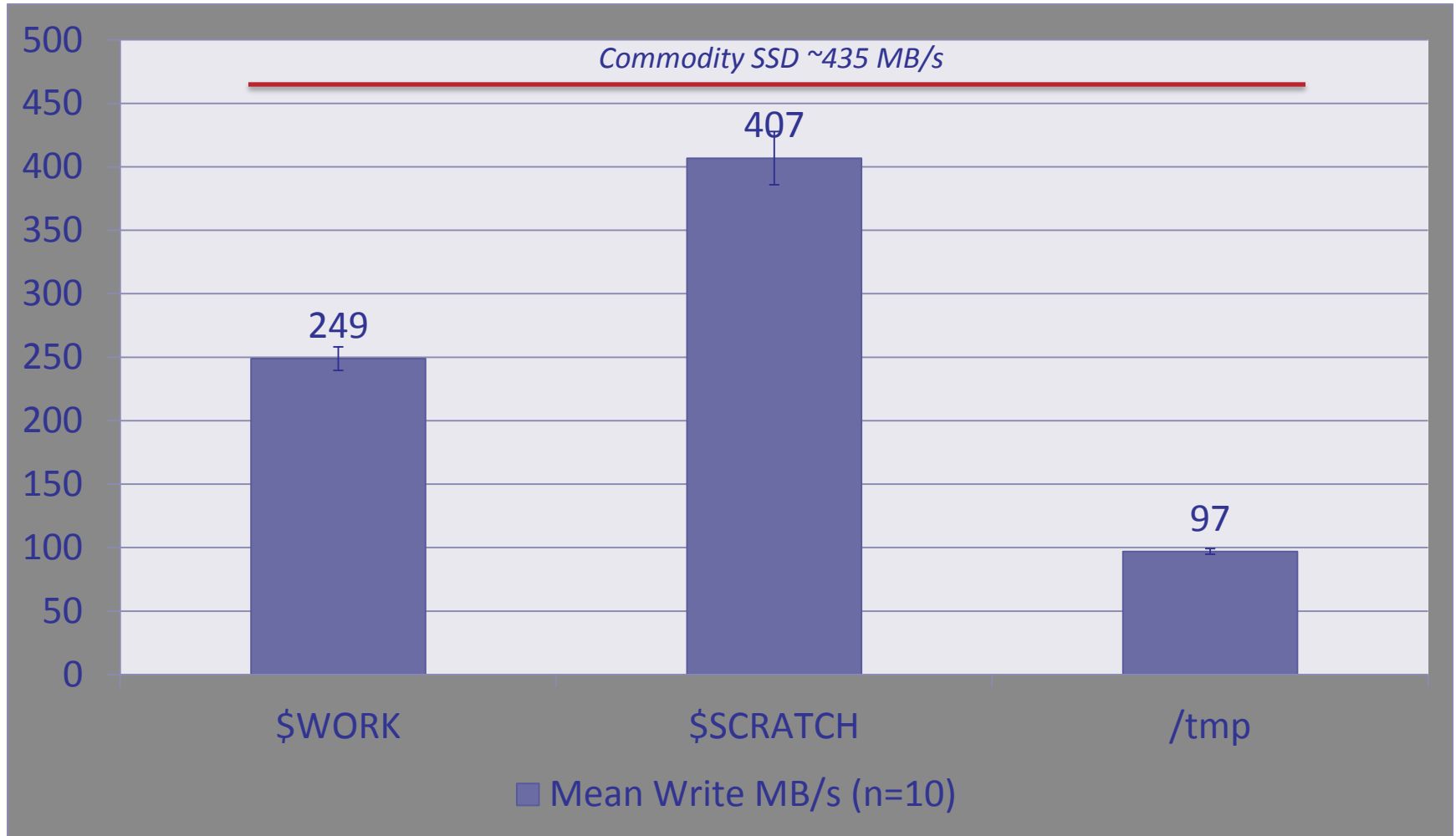
Stampede Large Memory Nodes

- 16x independent 1 TB DDR3 RAM / 4x 8-core E5 4650 Intel Sandy Bridge processors (slower clock rate than normal nodes)
- Accessible via dedicated queue: `-p largemem`
- You may request two nodes in a single job*
- You may have a total of 4 jobs active
- Charge rate: 2x SU x 2x number processors = 64 SU/wall clock hour

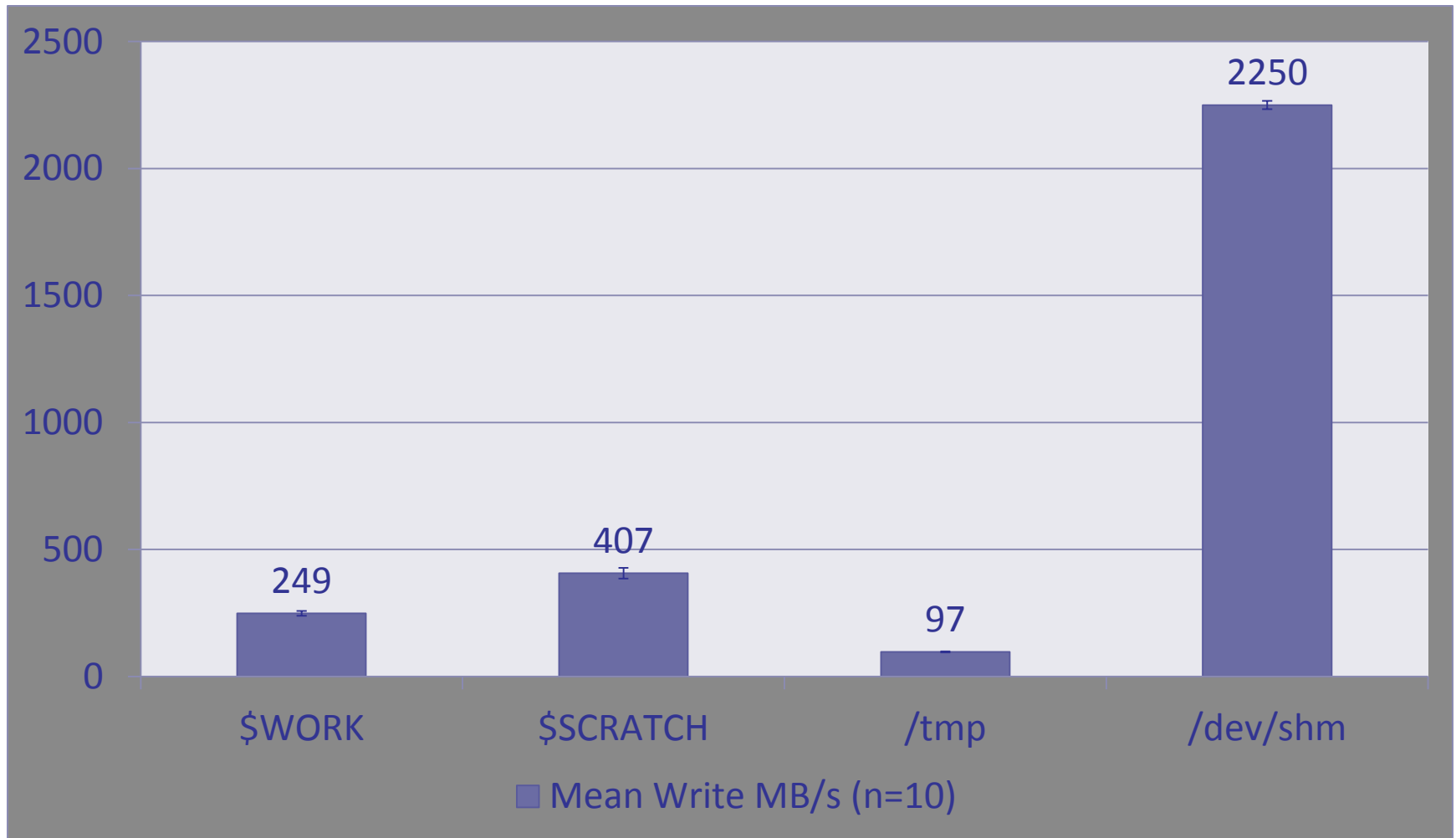
Use Cases

- Things that need a lot of RAM (obviously)
- “But what if I don’t need 1 TB shared RAM?”
- Is your code IO-bound (if it’s data-intensive it may very well be)?
- Try using Stampede’s largemem node memory as a RAM disk
- Up to 50% of available RAM is allocable as RAM disk. 500 GB is a lot of storage!

Stampede FS Write Performance

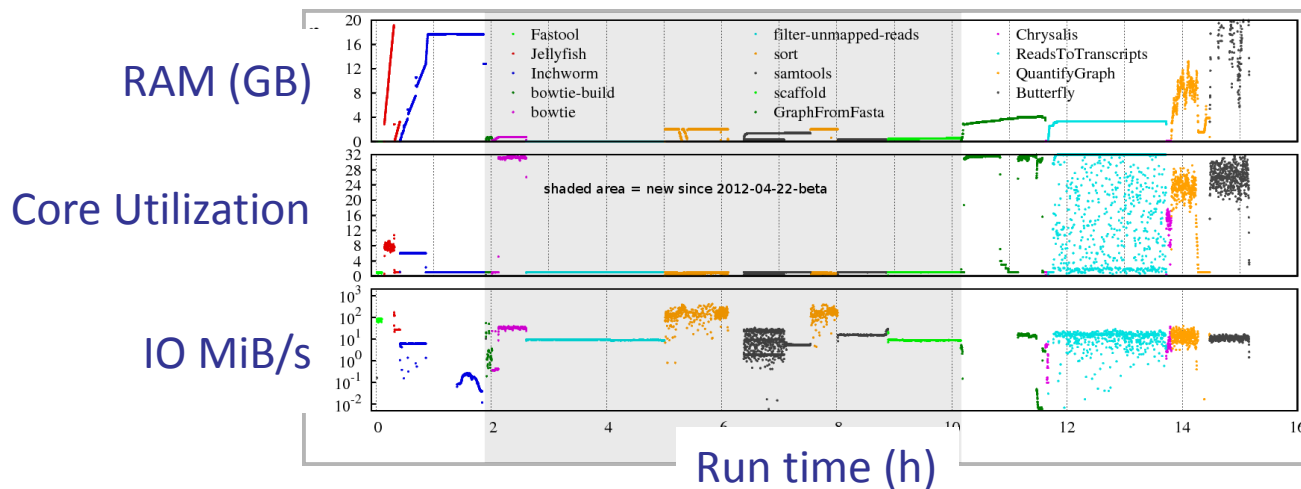


Stampede FS Write Performance



Test Case: Trinity RNAseq Assembler

- Trinity is both a large memory (500 GB) and IO-intensive application
- Profile on largemem with and without use of /dev/shm



Trinity Run Time (s) +/- dev/shm

Function	\$SCRATCH	RAM disk	Acceleration
Convert inputs	416	372	12%
Merge files	41	13	215%
Jellyfish Count	747	565	32%
Dictionary dump	42	29	45%
Kmer read in	288	284	1%
Kmer sort	103	103	0%
Kmer prune	1276	1010	26%
Bowtie forward index	240	172	40%
Bowtie mirror index	238	157	52%
GraphFromFasta	3420	2875	19%
Chrysalis	871	710	23%
ReadsToTranscripts	4628	3590	29%
QuantifyGraph	1928	1399	38%
Butterfly	2359	2110	12%
Total	16597	13389	24%

Using /dev/shm

- `cp <data> /dev/shm`
- `cd /dev/shm`
- Proceed as usual but...
 - Explicitly keep any intermediate or output files you need to persist beyond end of job
 - If you can checkpoint or snapshot, do so and send to `$SCRATCH`
 - Double check your data/intermediate file strategy to ensure that total size $< 500 \text{ GB}$ -OR- $\text{max}(\text{RAM})$

Code Optimization: PCIT

- Partial Correlation coefficient with Information Theory
 - Identifies significant correlations to define edges in a weighted network
 - Applicable to any correlation-based network
 - Scaled to ~8k x 8k on commodity hardware



Watson-Haigh NS, Kadarmideen HN, Reverter A. PCIT: an R package for weighted gene co-expression networks based on partial correlation and information theory approaches. *Bioinformatics*. 2010 Feb 1;26(3):411-3

Collaboration between James Koltes, Nathan Weeks (Iowa State), and Lars Koesterke (TACC) to find networks in a 40k element co-expression data set with the ultimate goal of going MUCH larger. Full Systems biology-level correlation analyses.

Code Optimization: PCIT



- Naïve FORTRAN implementation re-coded, optimizing for run time AND memory usage
- Along the way, doubled sensitivity by fixing a precision error
- Results accepted for XSEDE13 conference paper + lightning talk
- <https://www.xsede.org/web/xsede13/lightning-talks>

	Original $O(n^3)$	Rewrite 1 $O(n^2)$	Rewrite 2 $O(n^2)$	Xeon Phi $O(n^2)$
Time	160,000 hours	3.7 hours	20 hours	20% faster*
Memory	>256 GB	32 GB	0 GB	

***Optimization took this
from a Big Data
problem to a desktop
computing problem!***

Next steps:

- 1 million element² data set
- Open source release
- Release as a service via iPlant REST API

Take Home Messages

- There are alternative uses for large shared memory – get in there and find them. We're happy you're on the system!
- Get your code running. Take advantage of the resources. But make sure you take some time to consider efficiency:
 - If it seems like there might be improvements to be had (your code is taking 1+ TB RAM, is running > 24h, stresses on the file system, etc.) talk to TACC consulting scientists

QUESTIONS?