ALMA pipeline profiling Phase 2 report

Scientific Computing Group

Tests on Phase 2

AOC cluster

- Serial benchmarks for all datasets
- Parallelization breadth (number of MPI processes)
- Storage type
- Concurrency

Amazon Web Services (AWS)

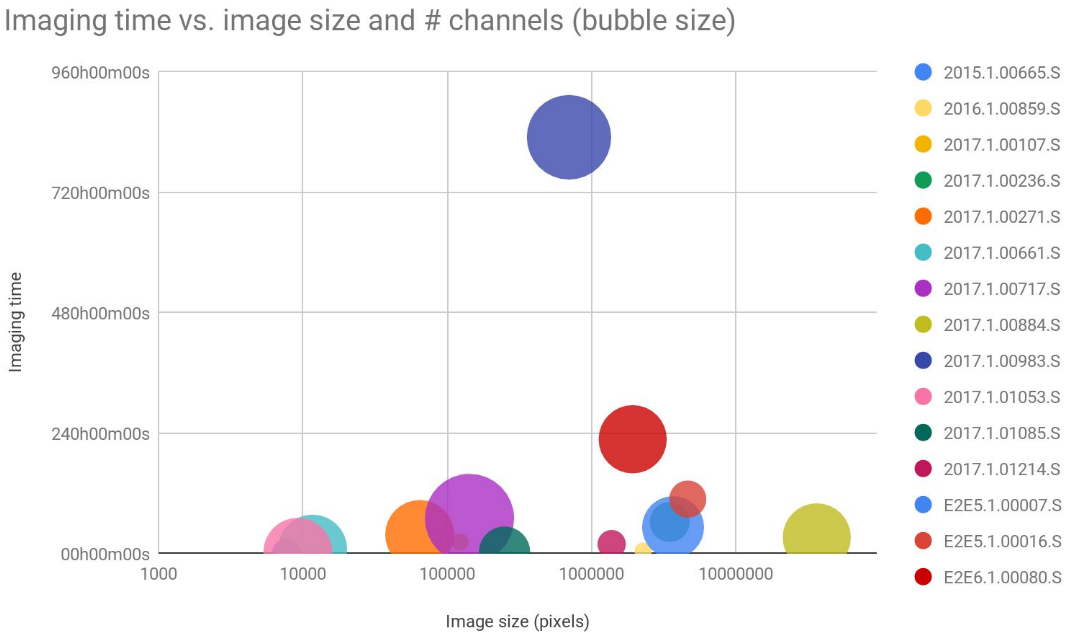
- Parallelization breadth (number of MPI processes)
- Memory limit
- Timing vs CPU type
- Number of OpenMP threads

Data selection and parameter space coverage

Improve coverage with respect to phase 1 - reviewed 4 new datasets from which 1 was selected (red data point on plot)

The following datasets were selected:

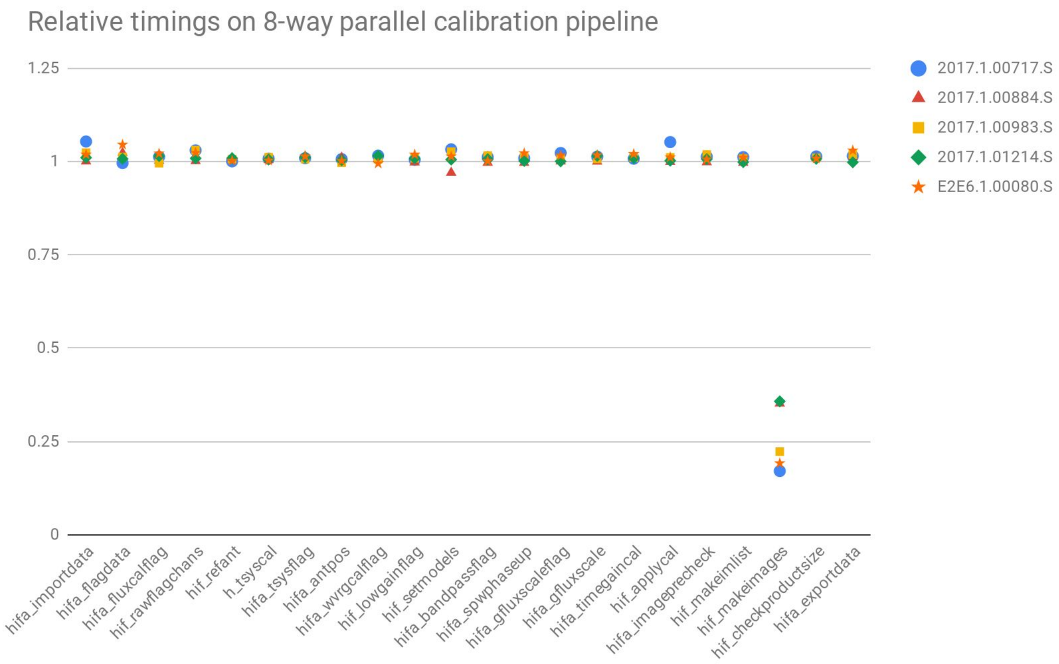
- 2017.1.00717.S
- 2017.1.00884.S
- 2017.1.00983.S
- 2017.1.01214.S
- E2E6.1.00080.S



Parallelization breadth (MPI) - calibration pipeline

- nmpost001-050 (E5-2670, 192 GB)
- Serial and 8-way parallelization
- No multi-MS
  ⇒ only tclean parallelized
- hif_makeimages() run time reduced by 63 - 85%

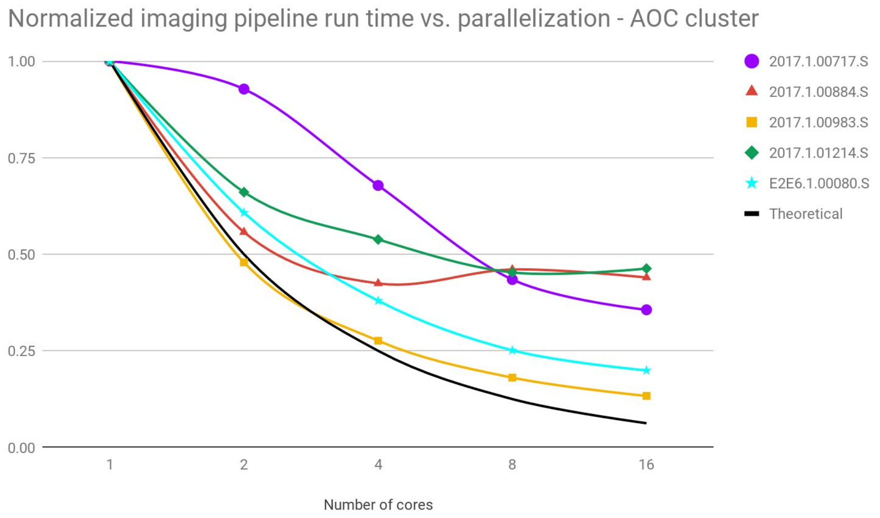
- Reduction of 1-15% total time

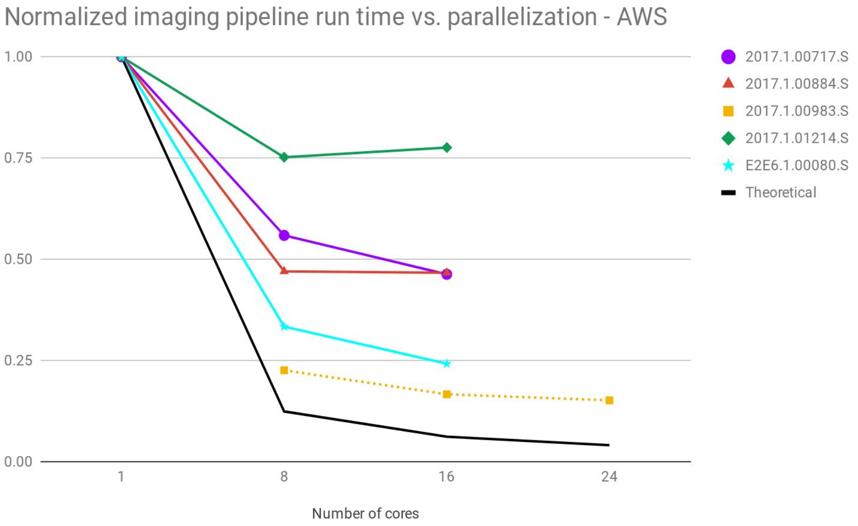


Parallelization breadth (MPI) - imaging pipeline

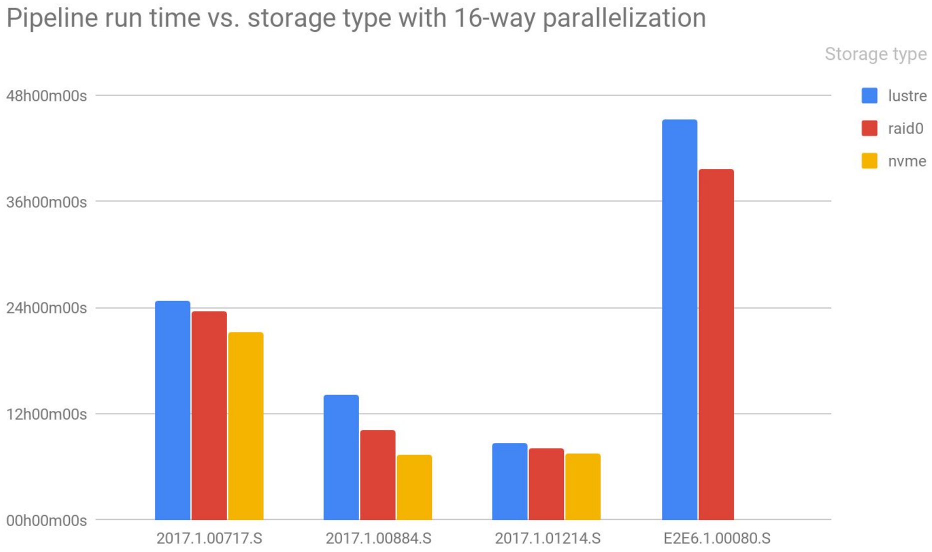
- nmpost001-050 (E5-2670, 192 GB); m5.12xlarge (Platinum 8175, 192 GB)
- Theoretical limit (black solid line): Serial run time / Number of MPI processes
- Increased run times selected for deeper investigation

- Run times decrease nearly linearly with increased MPI parallelization breadth





Storage type



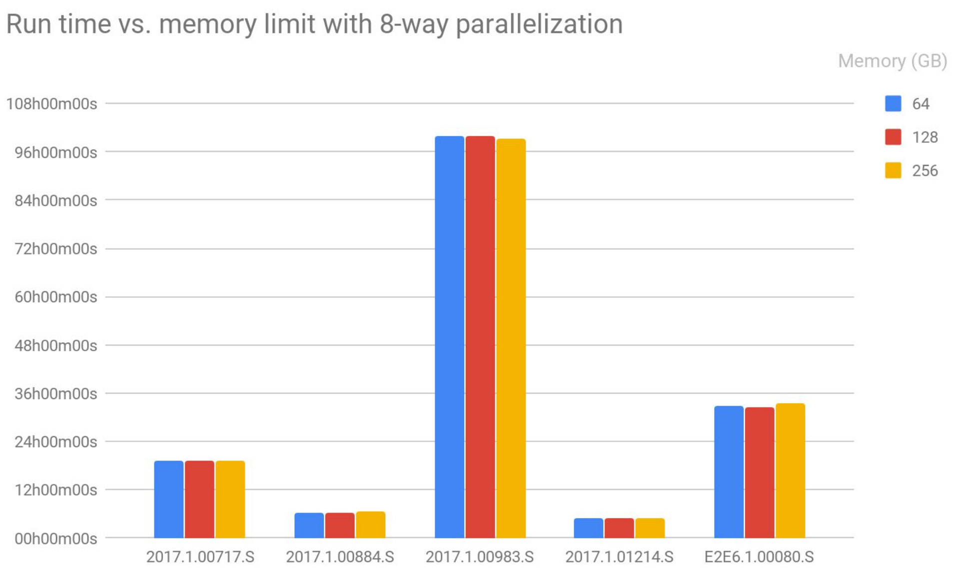
- nmpost001-050 (E5-2670, 192 GB)
- NVMe - 1.5 TB;
  RAID0 with 3 HDDs - 2.7 TB
- 2017.1.00983.S is too large for the devices

- NVMe showed less than ~15% reduced run time - more testing needed with larger devices

Memory limit

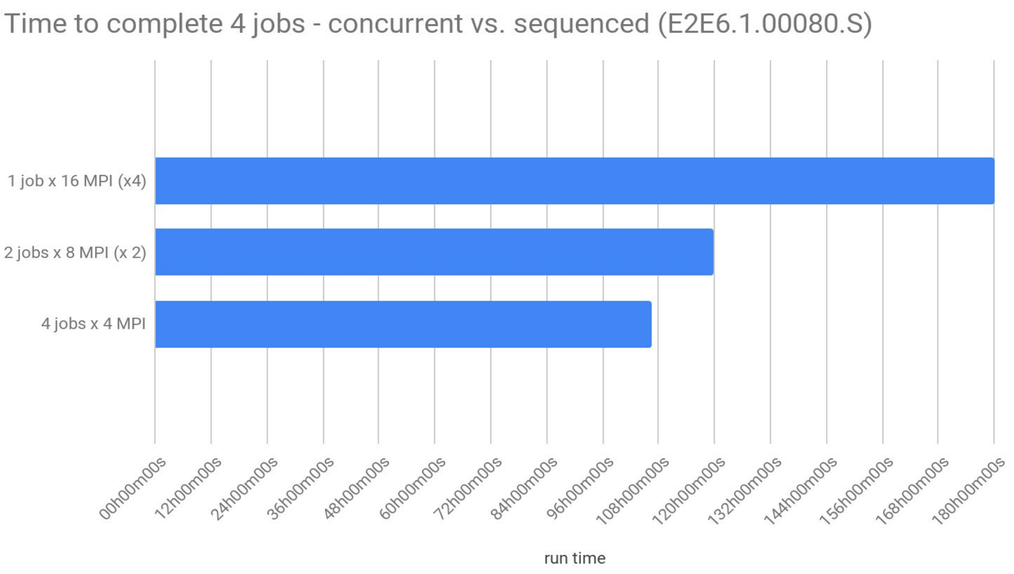
- r5.8xlarge (Platinum 8175, 256 GB)
- Limited to 64 and 128 GB via casarc system.resources.memory
- 256 GB not constrained via casarc

- No appreciable difference in imaging run time between 8, 16 and 32 GB RAM per process (8-way MPI) - more tests needed below 8 GB per process



Concurrency

- nmpost051-060 (E5-2640 v3, 256 GB)
- Clones of the same pipeline and data
- 2-way concurrency with 8-way parallelization
- 4-way concurrency with 4-way parallelization

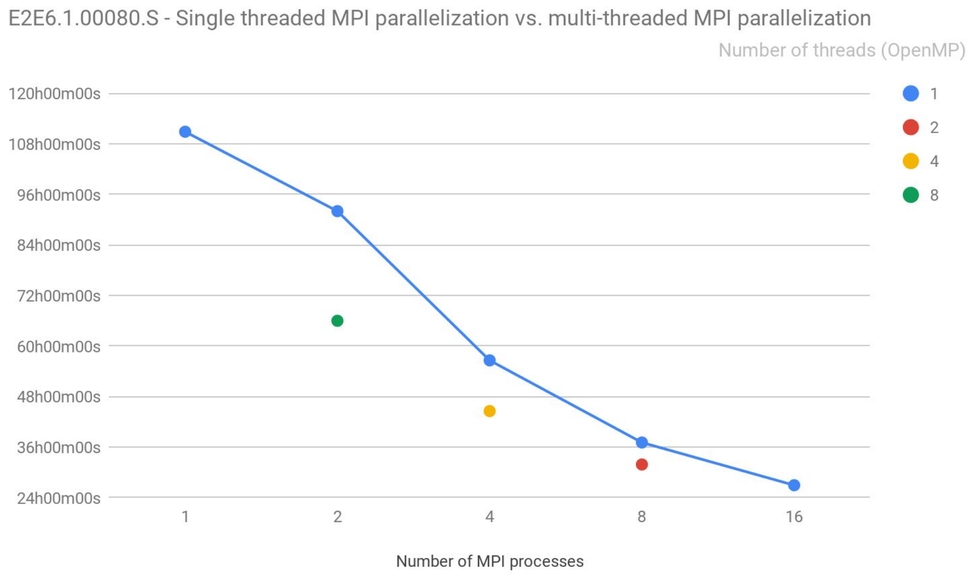


- 4-way concurrency with 4-way parallelization ⇒ most efficient timewise but swaps - more testing required
- Recommendation: Isolated or 2-way concurrency with 8-way parallelization

OpenMP

- m5.12xlarge (Platinum 8175, 192 GB)
- Test setup:
  MPI x OpenMP = 16 (# physical cores)

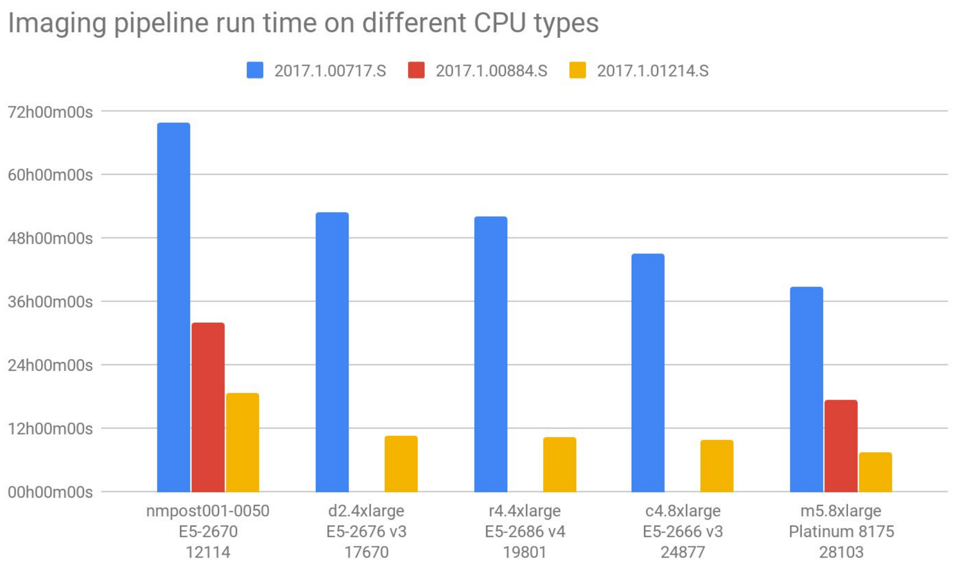
- MPI advantageous over OpenMP if there's enough memory to support more processes
  - OpenMP advantageous when memory is exhausted and there are unused cores



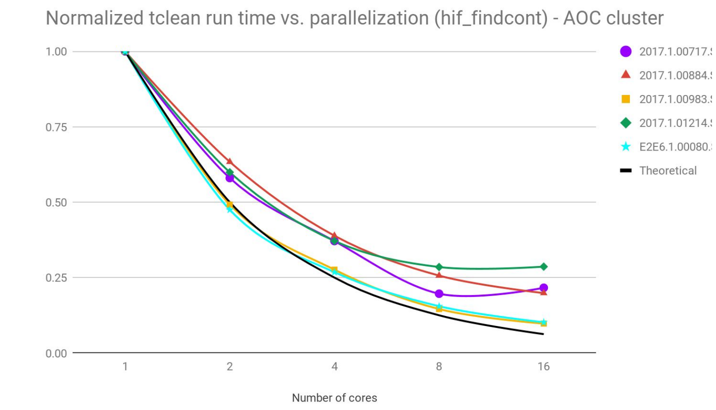
Timing vs. CPU

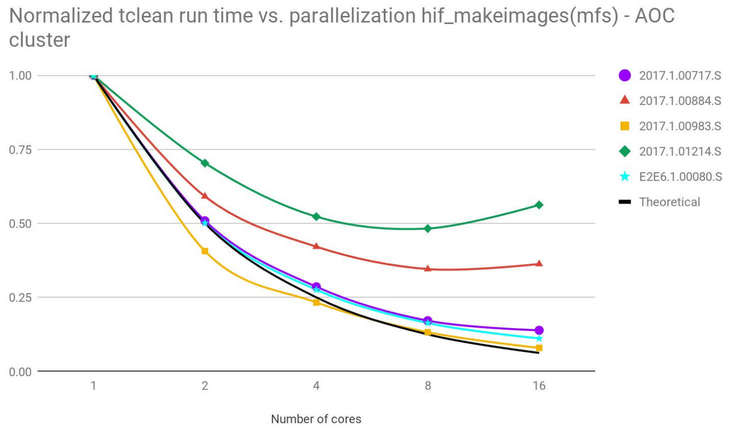
- nmpost001-050, E5-2670, 192 GB RAM
- d2.4xlarge, E5-2676 v3, 122 GB RAM
- r4.4xlarge, E5-2686 v4, 122 GB RAM
- c4.8xlarge, E5-2666 v3, 60 GB RAM
- m5.8xlarge, Platinum 8175, 128 GB RAM
- Serial runs
- Passmark measures CPU performance

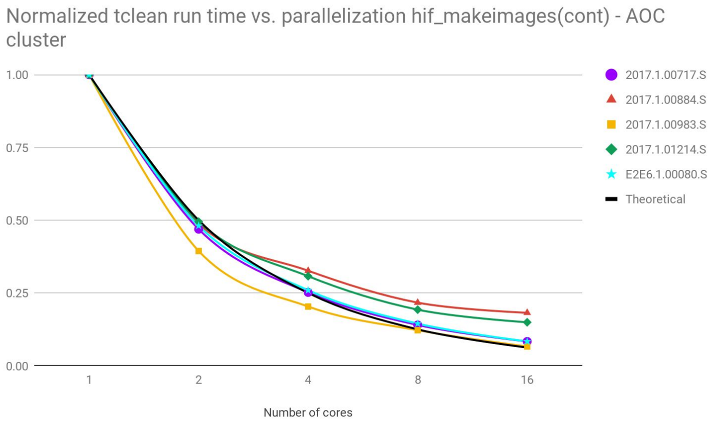
- Passmark helps comparing processors, but is not necessarily predictive of pipeline run time

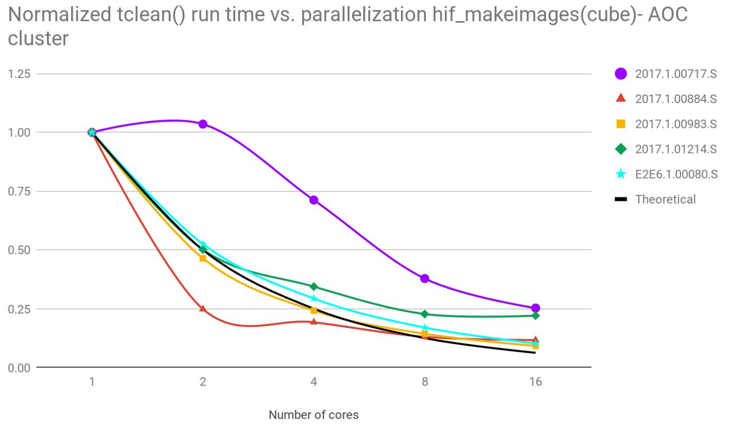


Investigation of tclean() unexpected timings





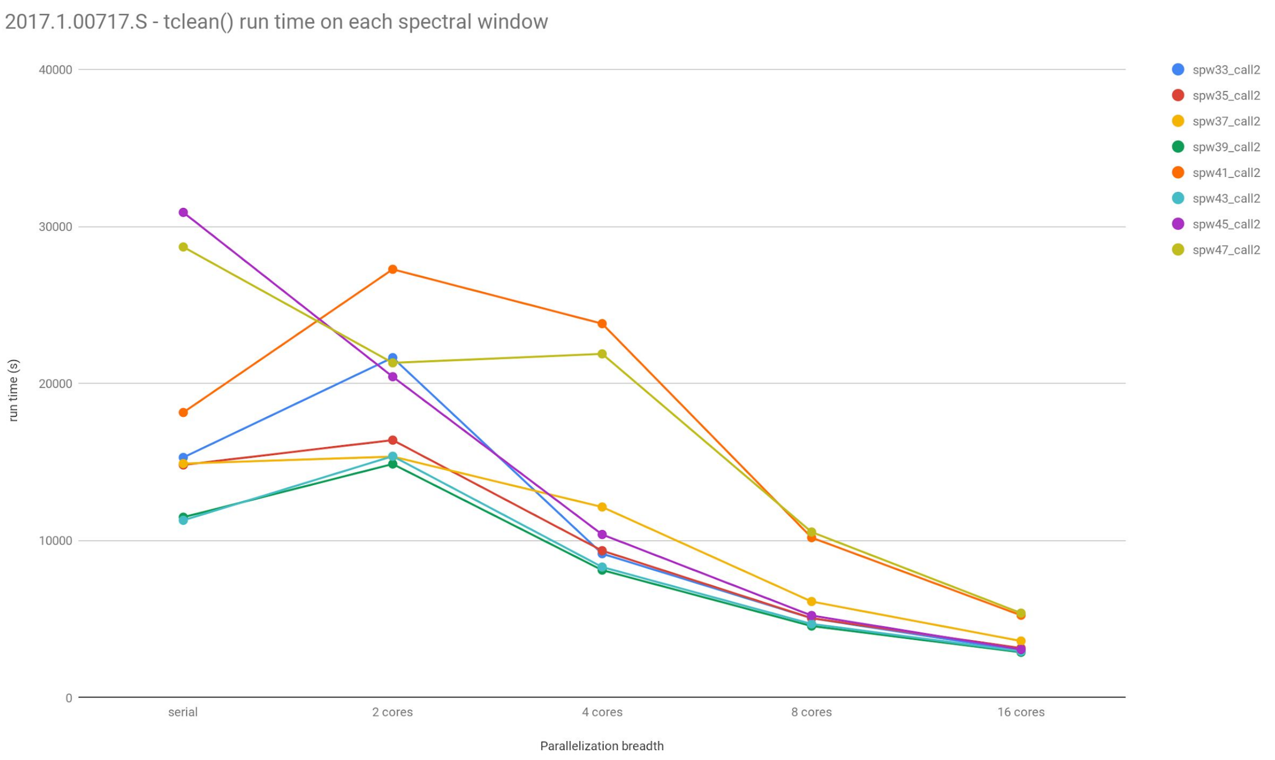




Continuum imaging - isolated spectral window

			2 cores	4 cores	8 cores	16 cores
2017.1.00717.S	start up time	709	664	772	1060
	time on major cycles	8360	4473	2295	1432
	time on minor cycles	21	25	24	37
	scatter/update model time	16	15	32	52
	TOTAL	9106	5177	3123	2581
2017.1.00884.S	start up time	268	288	359	522
	time on major cycles	992	624	398	284
	time on minor cycles	18	19	21	21
	scatter/update model time	5	7	14	27
	TOTAL	1283	938	792	854
2017.1.00983.S	start up time	2598	2741	3084	3556
	time on major cycles	95753	54381	29545	16292
	time on minor cycles	86	93	85	102
	scatter/update model time	20	54	74	128
	TOTAL	98457	57269	32788	20078
2017.1.01214.S	start up time	633	657	813	1151
	time on major cycles	1089	644	421	323
	time on minor cycles	8	8	12	14
	scatter/update model time	9	18	29	43
	TOTAL	1739	1327	1275	1531
E2E6.1.00080.S	start up time	796	860	986	1257
	time on major cycles	5107	2643	1390	741
	time on minor cycles	0	0	0	0
	scatter/update model time	8	5	13	20
	TOTAL	5911	3508	2389	2018

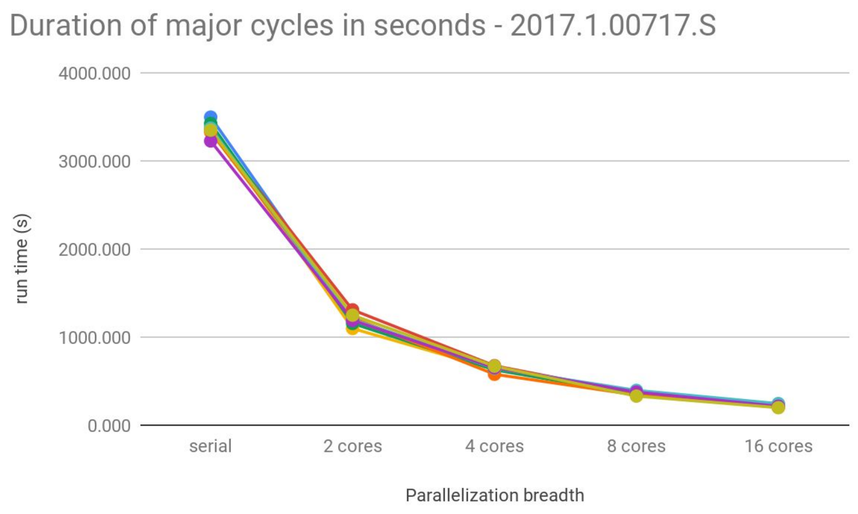
Cube imaging

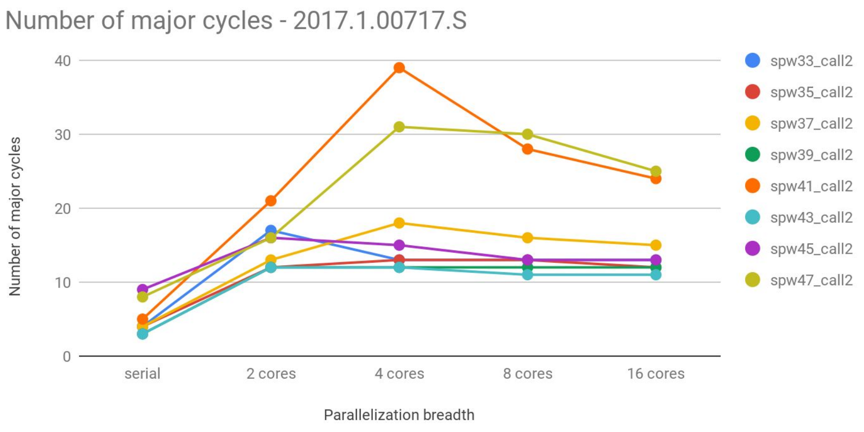


Cube imaging

- Duration of major cycles decreases as expected with parallelization breadth

- Number of major cycles varies due to CLEAN convergence being different for different subsets of channels that result from different number of processes

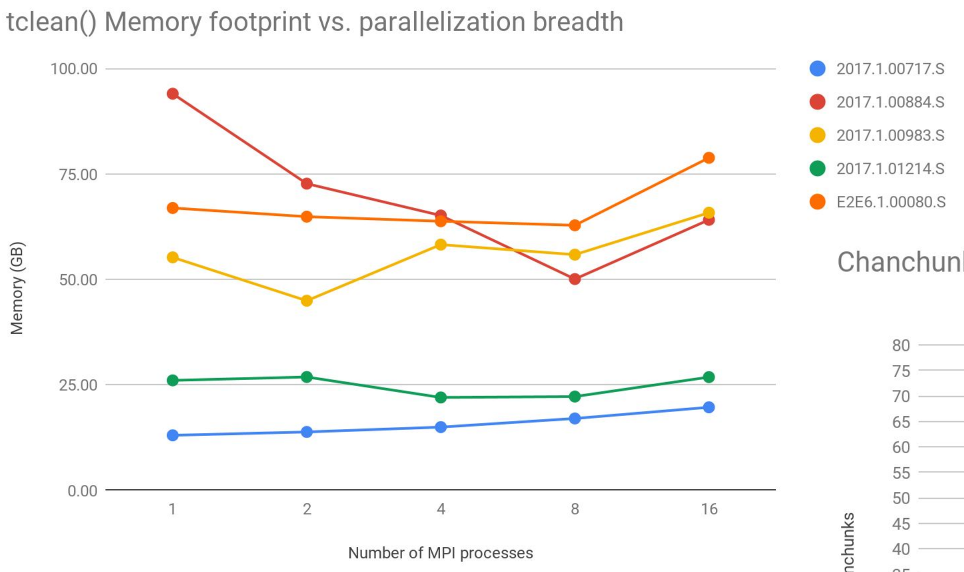


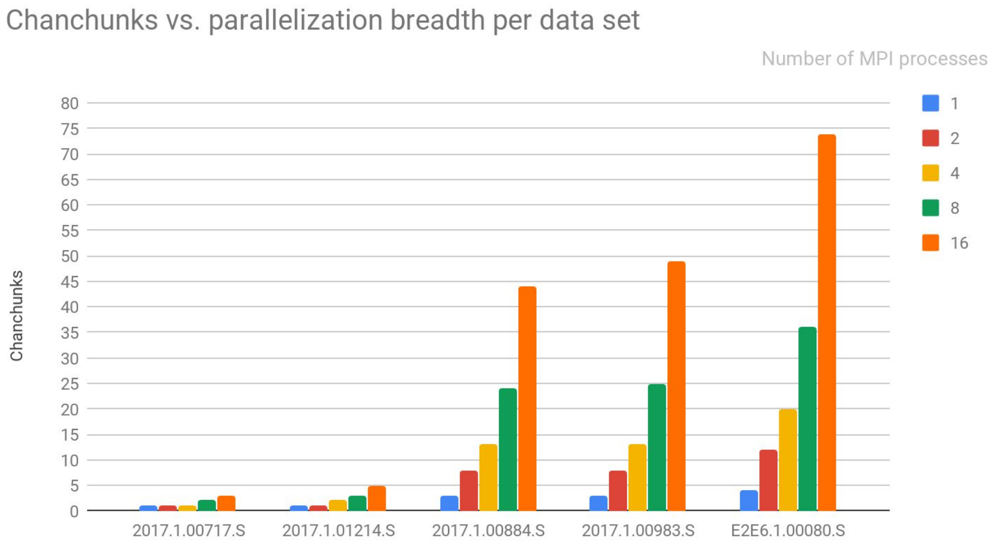


Next steps

- Rerun a subset of the tests with a newer version of CASA (most likely the pre-release that is current when new tests begin)
- Investigate the following points in this report:
  - Different number of major cycles for different parallelization breadth
  - Low memory behavior including swap memory usage for concurrent jobs
  - Performance of different storage types with larger datasets
- Enable access to CPU event counters via the PAPI (Performance Application Programming Interface - https://icl.utk.edu/papi/overview/index.html), to allow relating software performance to processor events
- Investigate locking overhead in parallel runs of self calibration when updating the model column (CAS-12612)
- As time allows, perform tests identified by operations and CASA developers.

Parallelization breadth (MPI) - imaging pipeline





Concurrency