Radio Interferometric Imaging using a cluster of GPUs on PATh

Felipe R. H. Madsen, NRAO

National Radio Astronomy Observatory

Abstract

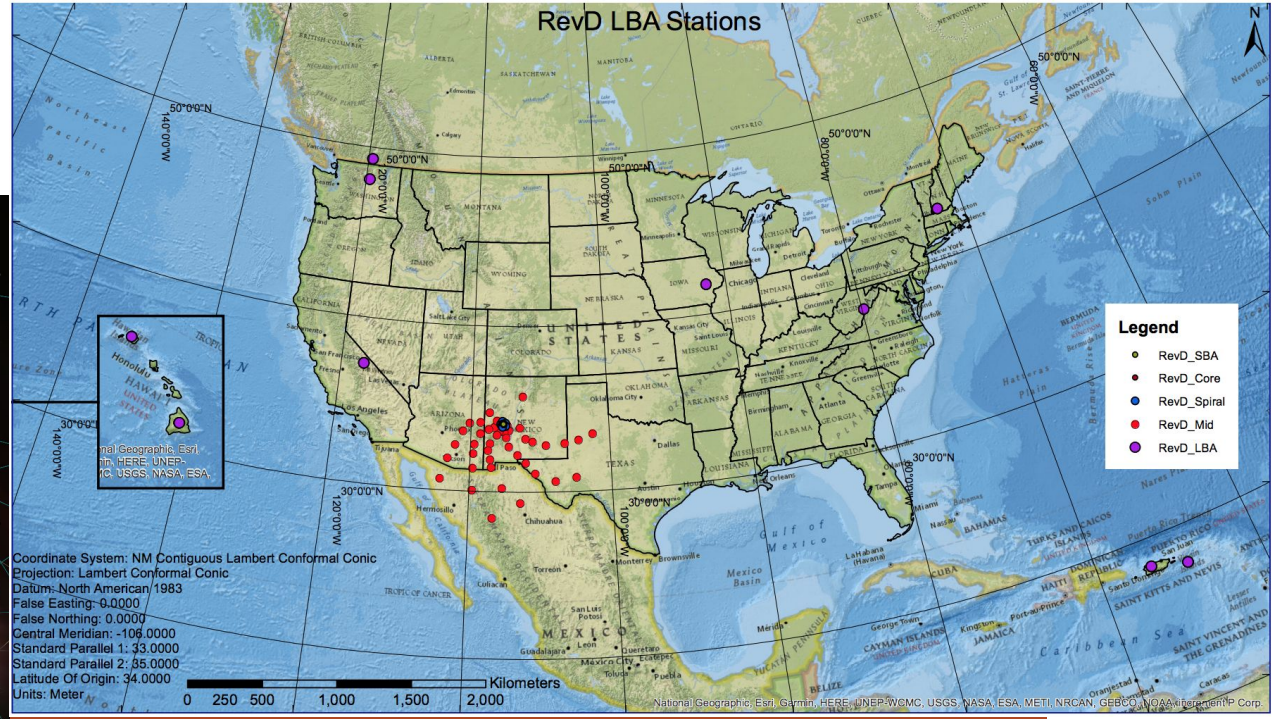
Radio Interferometers enable imaging the radio emission from celestial objects by combining signals from different antennas in a compute intensive process that may greatly benefit from GPU processing power and a high throughput computing model. We present an overview of this imaging process as well as current and future compute resource requirements. We discuss our HTCondor implementation for imaging in the PATh facility by distributing processing over multiple GPUs, as well as the development of a tool for data visualization that helps in high-level time profiling. We present testing results for measuring run time scalings of GPU-enabled imaging steps and successfully executing complete imaging procedures (imaging-to-convergence).

Outline

- Interferometers (current / future)
- Interferometric imaging
- Development of HTClean
- Testing with multiple GPUs on PATh
- Run time measurements and visualization
- Future work / Challenges

Interferometers operated by NRAO

VLA (top left), ALMA (top right),
VLBA (bottom left)
https://public.nrao.edu/



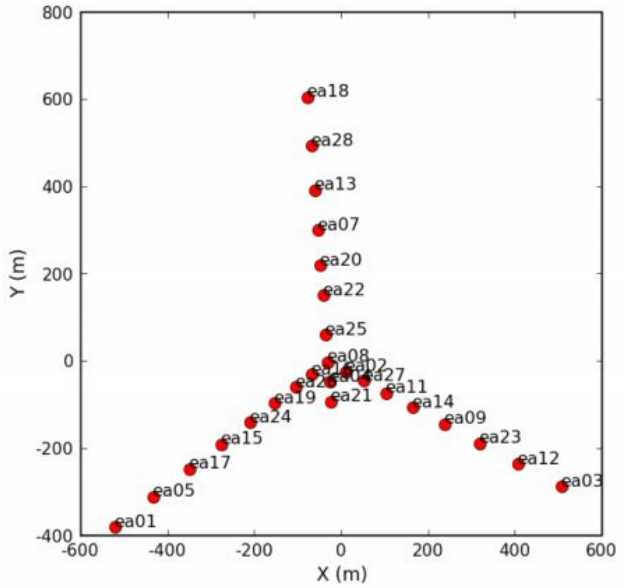
Future

next generation Very Large Array (ngVLA)
https://ngvla.nrao.edu/

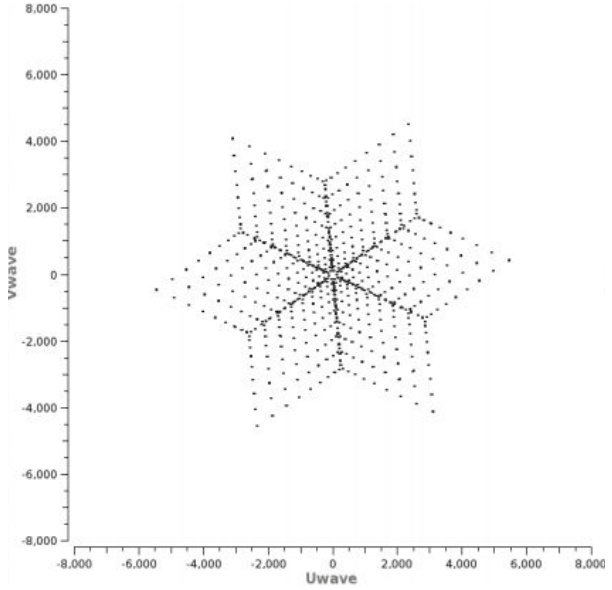
National Radio Astronomy Observatory

Interferometric Imaging is a Computational Problem

Antenna positions (VLA)

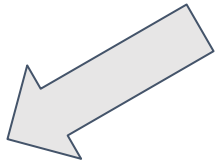


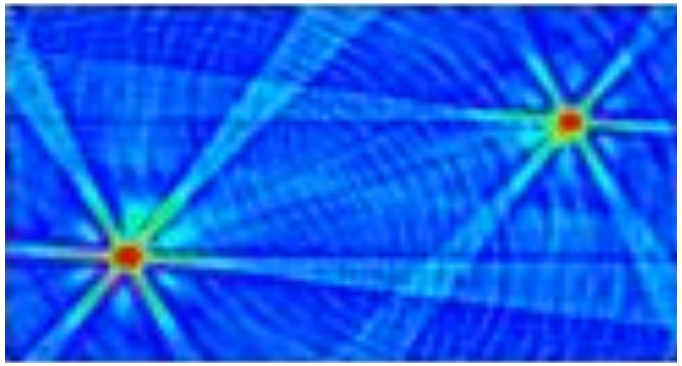
Distance, Orientation of all antenna pairs



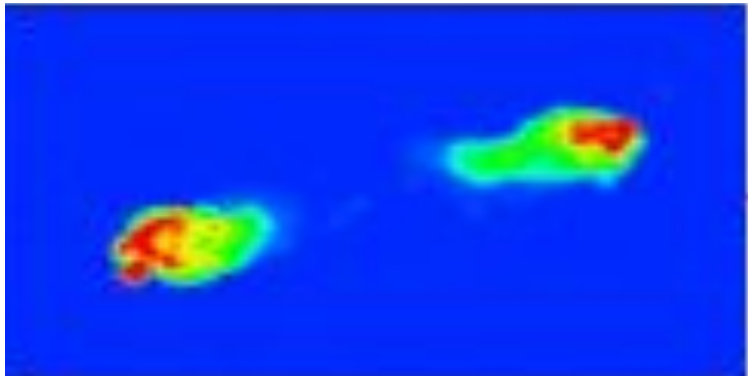
Sampling function

Gridding + FFT^{-1} of measurements





Post processing



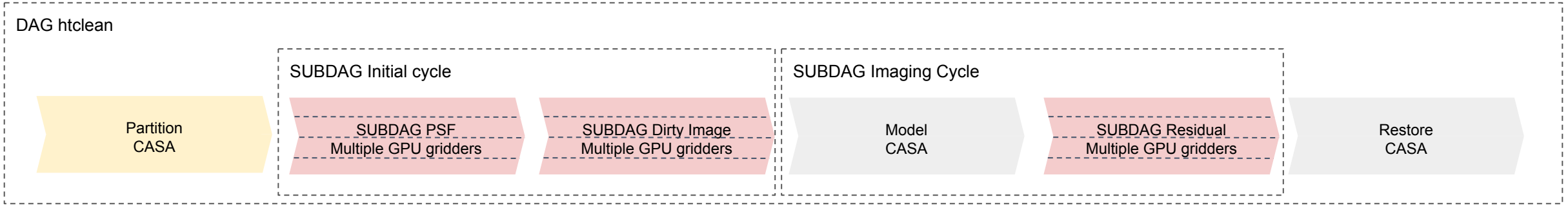
Dirty Image

Final Image

Size of computing

- Resampling measurements (gridding) > 90 % of compute cost
- VLA data rate is ~ 25 MB/s, ngVLA estimated at ~ 8 GB/s
  - Challenges to meet imaging requirements of largest projects at VLA data rate
    - Deep field observations: 9 TB input data (102 files); image size 84 MPixels
    - VLA Sky Survey (VLASS) 30k+ 1sq. degree images (268 MPixels/image)
- Estimated size of computing for ngVLA is 50 PFLOP/s
  - Results in $O(10^6)$-way parallelization with CPU cores running 24 x 7
  - Gridding on GPU reduces required parallelization to $O(10^3)$-way

HTClean implementation



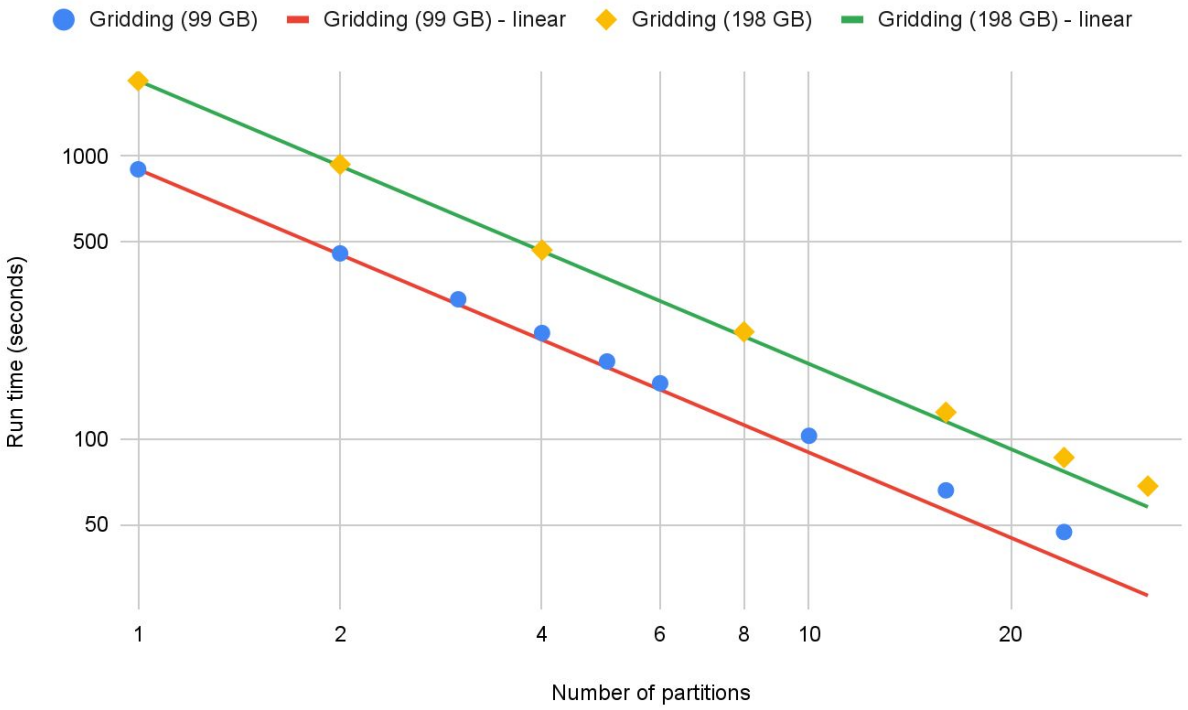
- High-throughput version of the main imaging task *tclean* in our imaging software CASA
- HTClean breaks down the imaging process in independent sessions that can be distributed
  - Addresses asymmetry in the two main stages of imaging:
    - residual cycle: highly parallelizable, high FLOPS, visibility domain
    - model cycle: serial (*continuum* imaging), low FLOPS, image domain
  - Imaging cycle iteration control using RETRY and POST SCRIPT
  - Input data can be partitioned along various axis: time, frequency, baseline length, polarization, field (pointing direction)

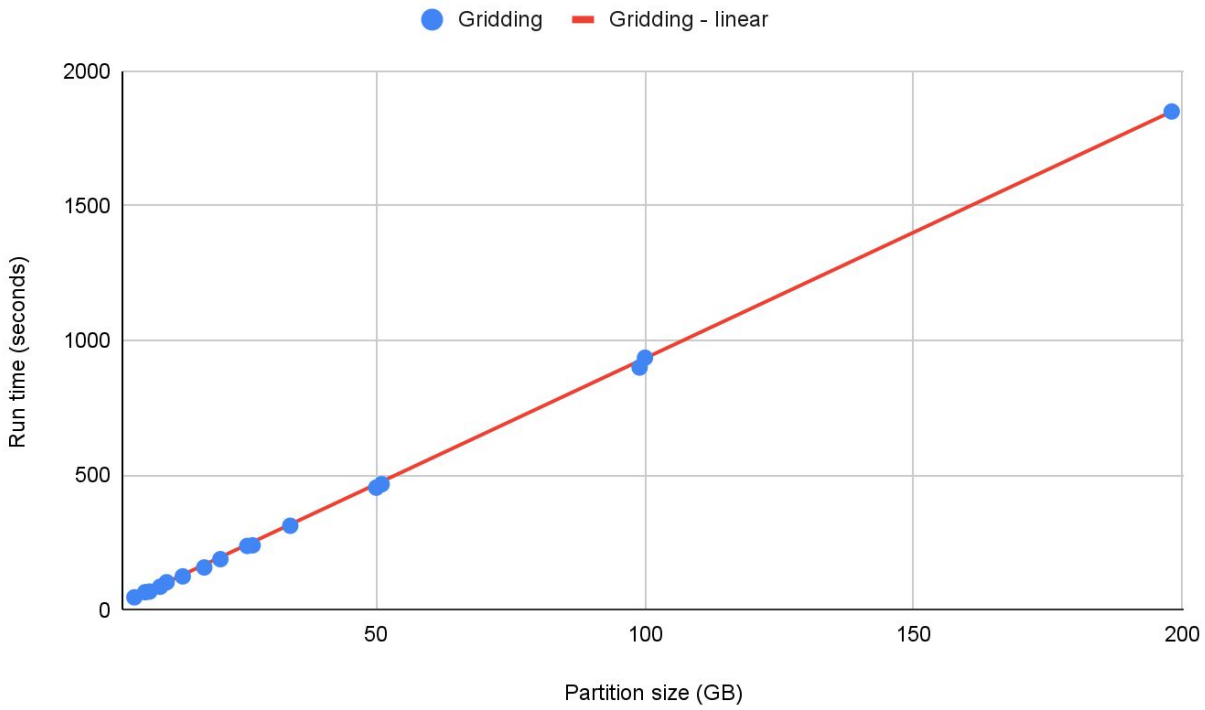
Testing with multiple GPUs on PATh

- Implementation changes
  - relocatable GPU gridder
  - make tarballs of image products (directories) for data transfer
  - improved mapping of intermediate (module) inputs and outputs
  - unique names for intermediate products (stash vs. name reuse) on each imaging cycle iteration
- Goals
  - measure run time scaling of GPU gridder
  - measure run time scaling of distributed processing
  - prototype ngVLA-scale data processing

Run time measurements and visualization

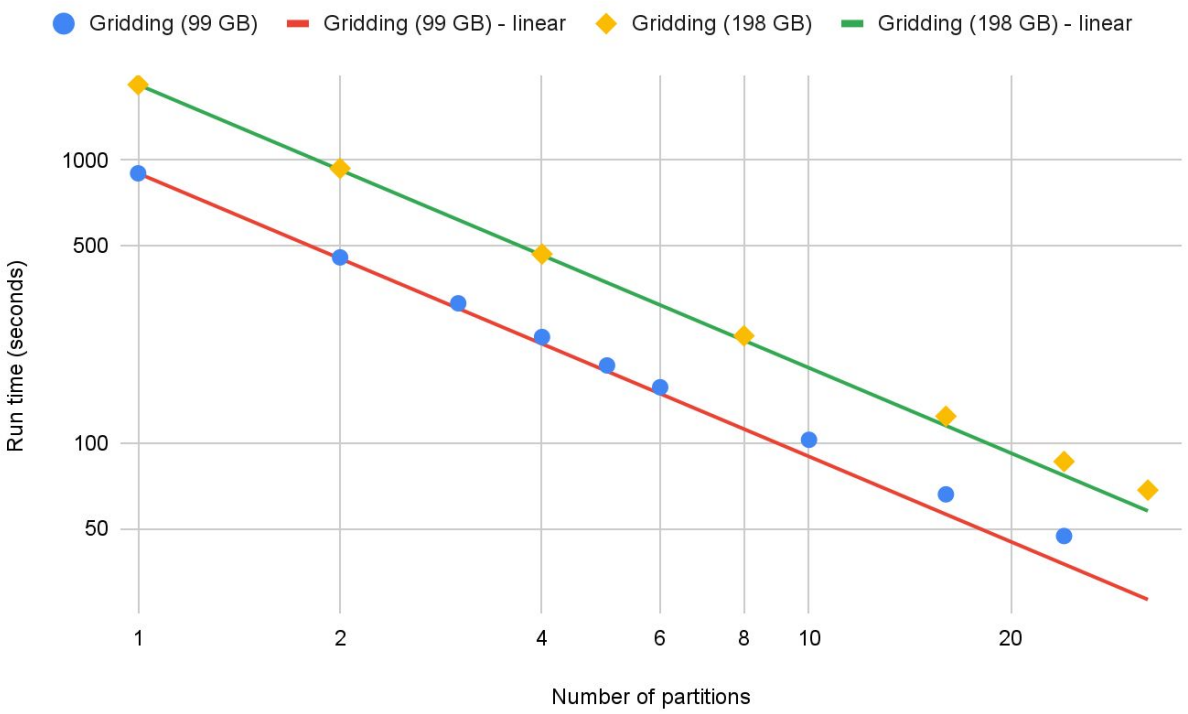
- GPU gridder run time as a function of number and size of partitions

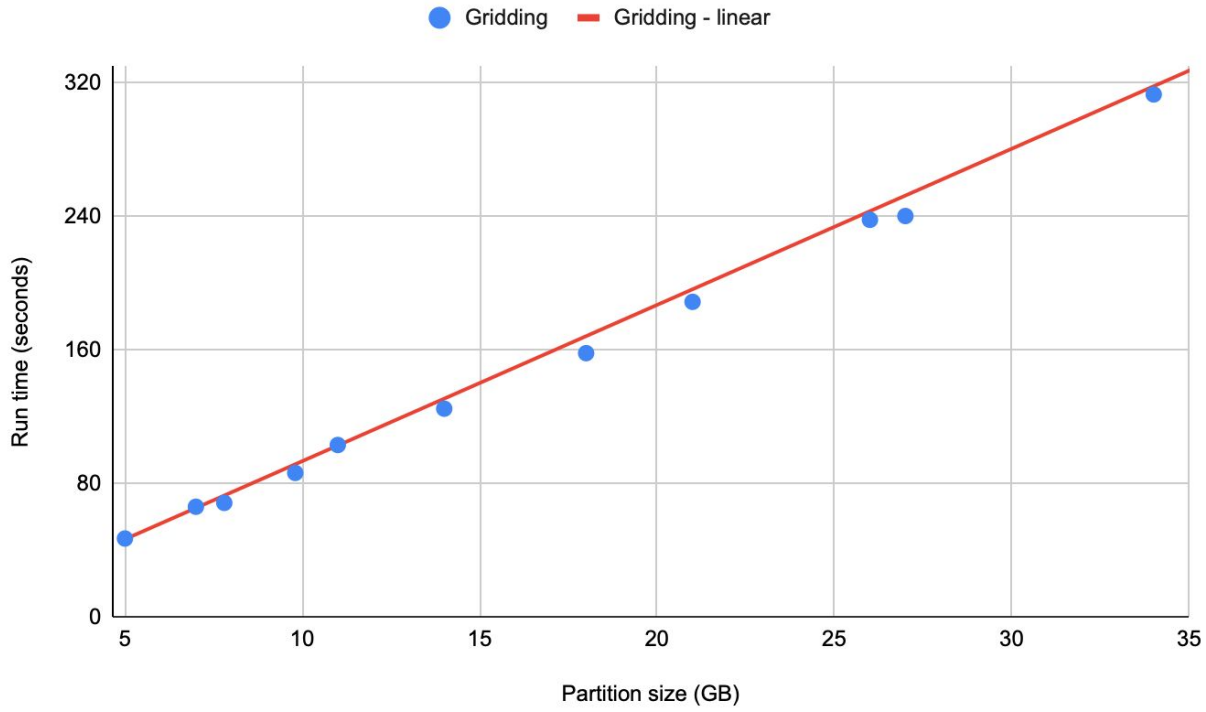




Run time measurements and visualization

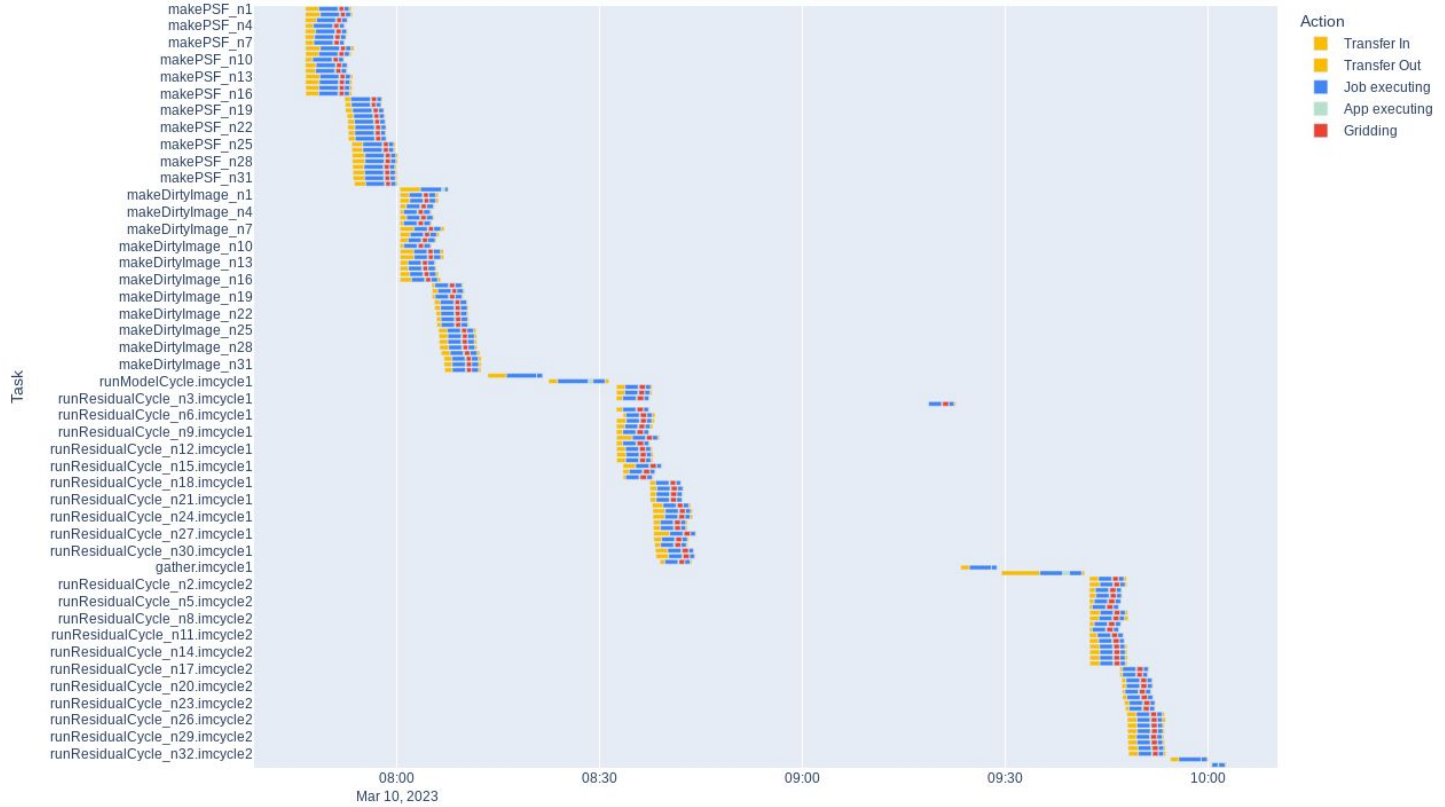
- GPU gridder run time as a function of number and size of partitions





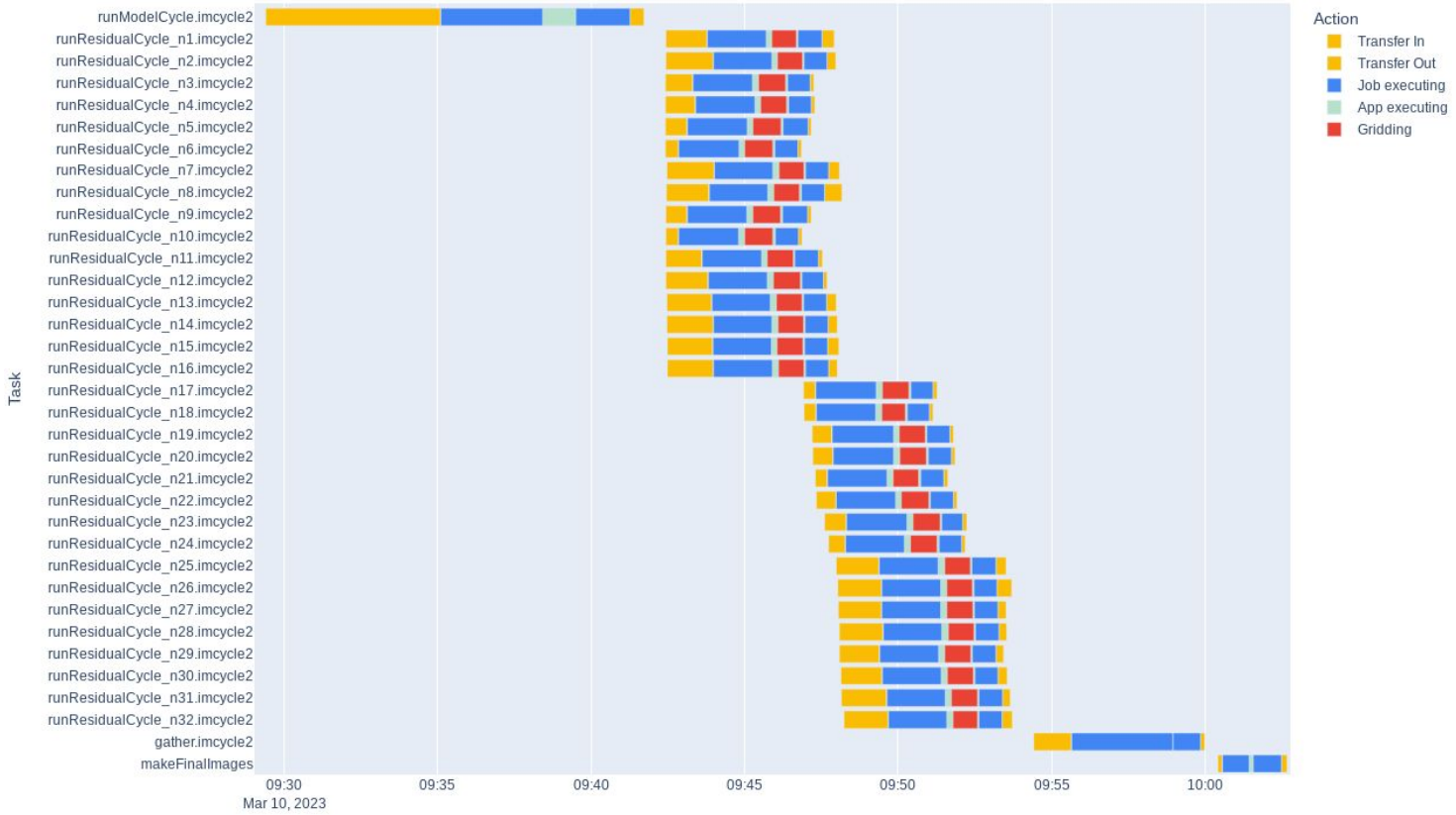
Run time measurements and visualization

- Concurrency plots: overall view of distributed jobs, idle times, transfer times…



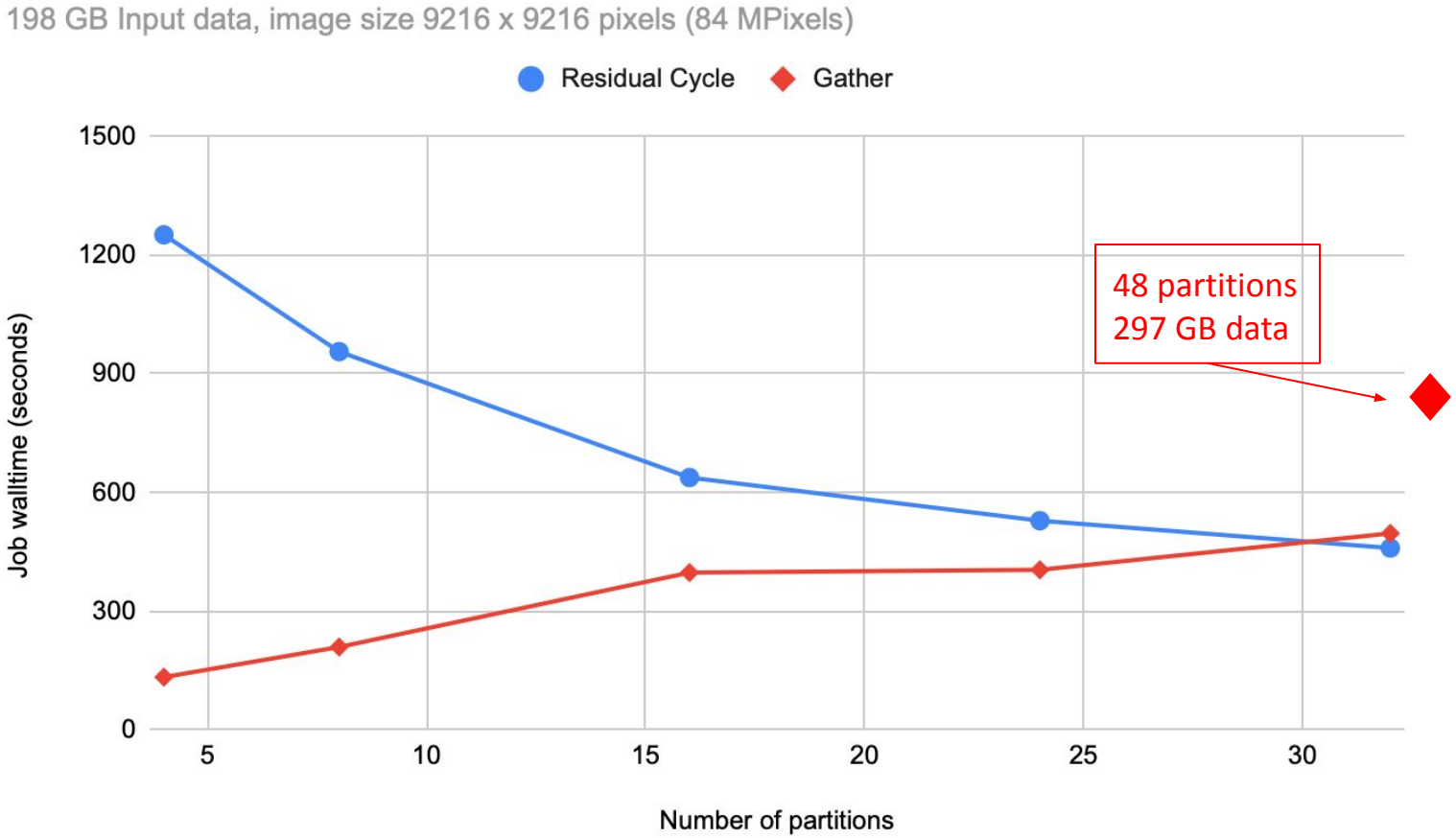
Run time measurements and visualization

- Concurrency plots also allow zooming in to details of specific jobs



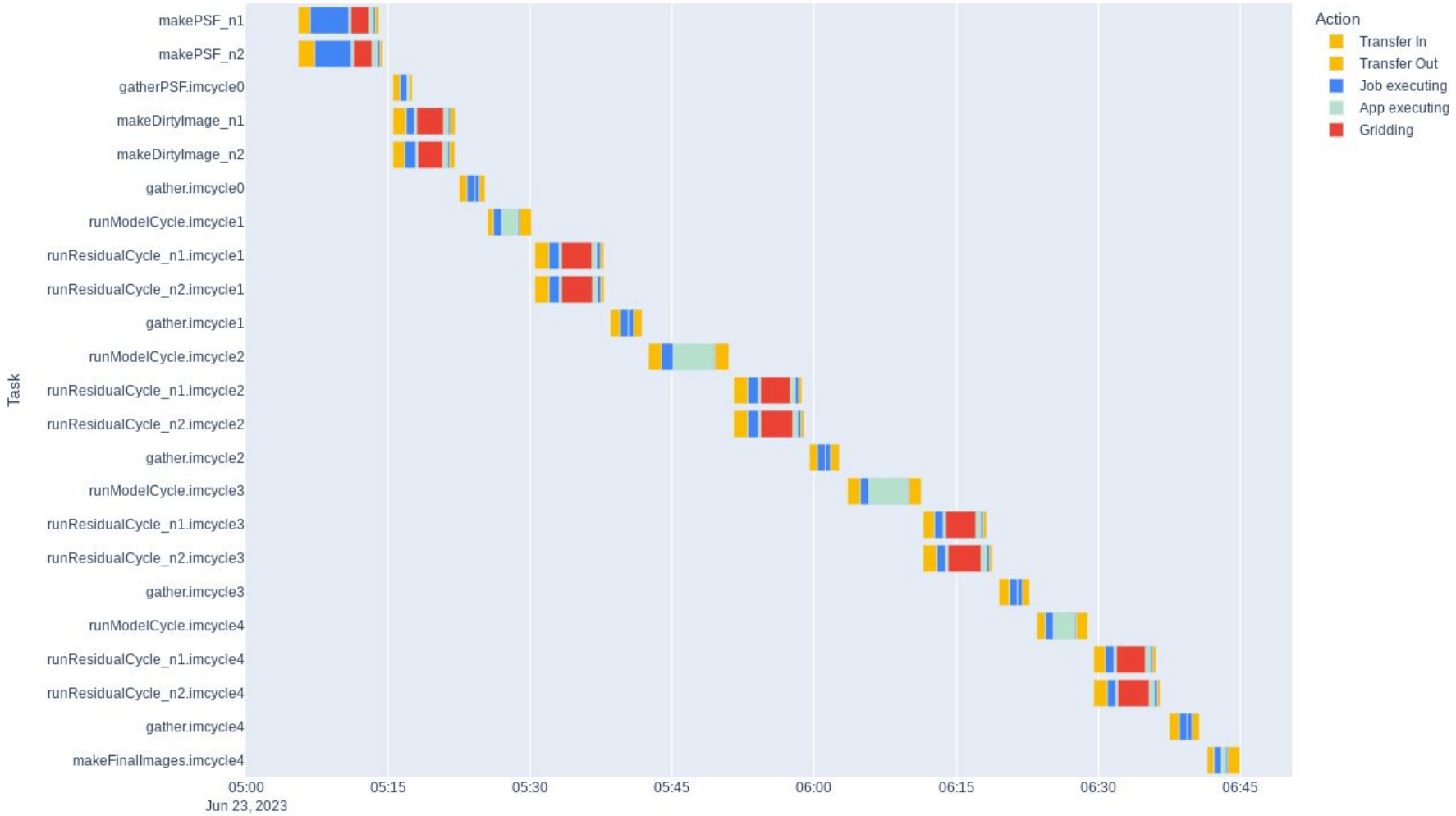
Run time measurements and visualization

- Gather barrier



Each partition makes a full size image ⇒ size of gather grows linearly with number of partitions
Most of the serial job's walltimes (gather, model cycle) is transfer in/out (yellow bars) and tar/untar (blue bars)

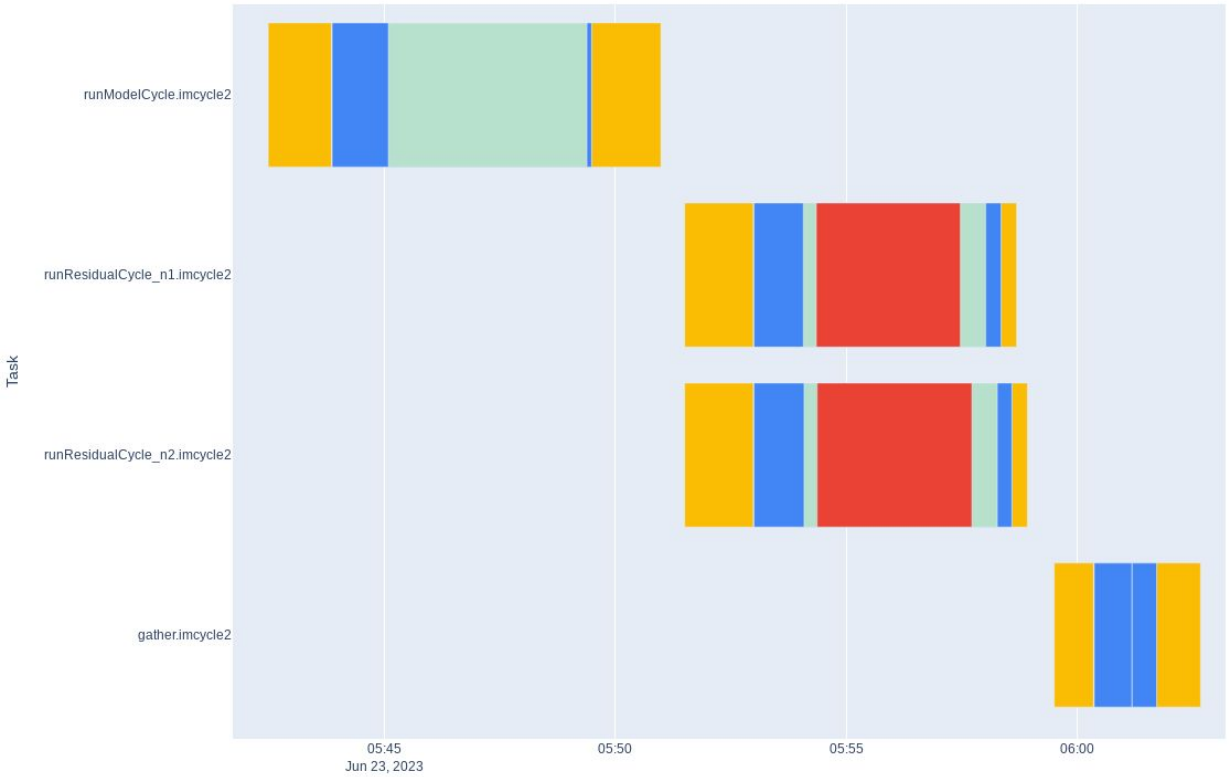
VLASS imaging to convergence - 2 partitions



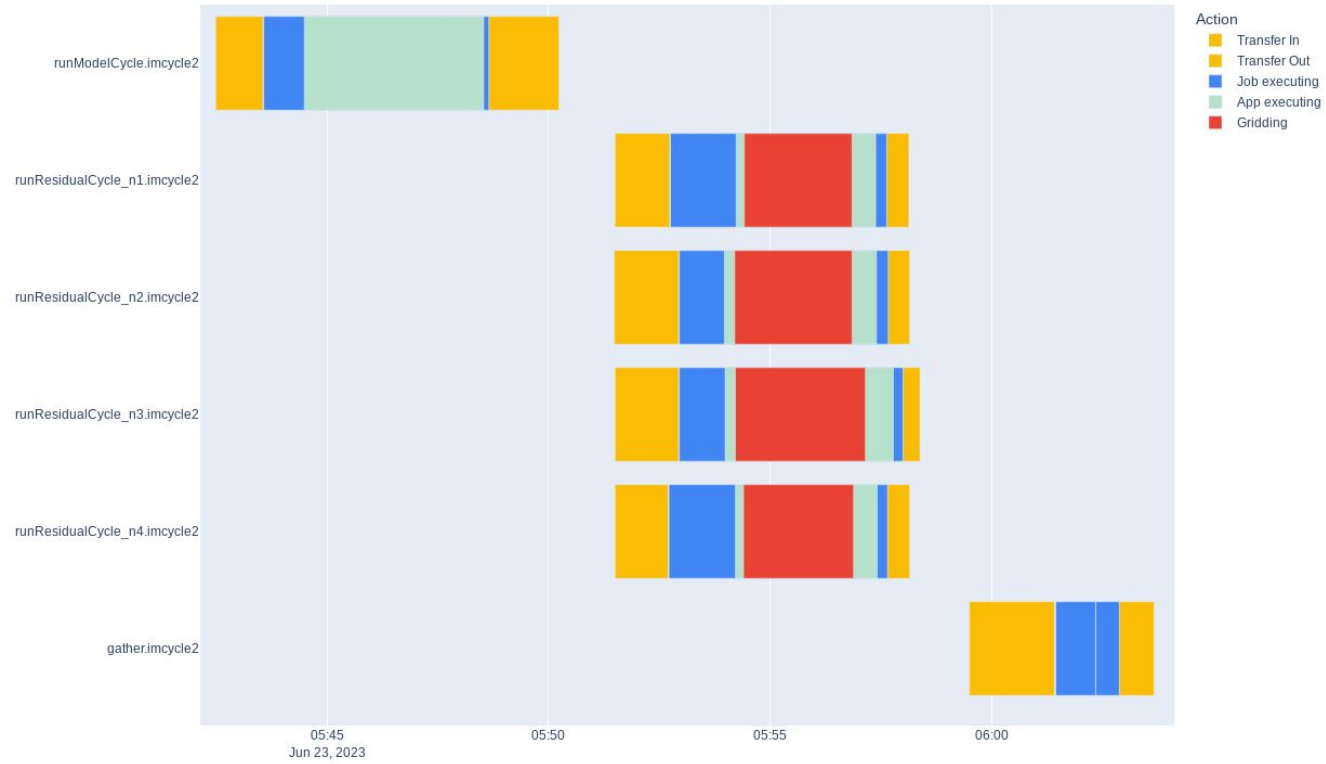
National Radio Astronomy Observatory

VLASS imaging to convergence - Imaging cycle

Image size: 16384 x 16384 = 268 MPixels



Residual cycle: 426 seconds; gather: 188 seconds



Residual cycle: 393 seconds; gather: 238 seconds

Future work / challenges

- Improve IO efficiency of gridding jobs
- Further expand data partitioning to other axes
- Integrate/test new model cycle software module
- “Gather barrier”
  - Optimize DAG design to minimize barrier
  - Investigate scalable design solutions
- Scale up residual cycle partitioning / distribution by at least 1-2 orders of magnitude - will need larger number of available GPUs, or hybrid distribution model (GPU/CPU)