Parallel Spectral Line Imaging in Obit

W. D. Cotton (NRAO), September 23, 2022

***Abstract*—This memo explores potential efficiency gains in processing multiple spectral channels in parallel and in comparing the relative speeds of CPU and GPU based gridding. If deconvolution of the channel images is not needed, parallel imaging produces some efficiency gains. If deconvolution is needed, the optimum solution is single channel at a time processing. Using GPU based gridding showed a slight (20%) reduction in run time for a comparable test. This largely shows that the gridding of visibilities in spectral line processing is less expensive than the rest of the operation.**

***Index Terms*—Spectral Line, GPU, Interoferometric Synthesis**

I. Introduction

RADIO interferometer spectral line observations can involve many channels which can, in principle, benefit from parallel processing of the basically independent channels. This is complicated by the possibility of varying amount of emission in different channels needing various depths of CLEAN. Parallel processing using GPUs has proven effective for continuum imaging [1] and a similar technique can be applied to line data. This memo evaluates such a technique using the Obit package [2] [1]. Examples using the MeerKAT array are described.

II. Radio Spectral Line Imaging

Observations in a given direction and in a given frequency band will in general contain both continuum (emission changes slowly with frequency) and "spectral lines" (emission changes rapidly with frequency). The main difference (if any) between continuum and line observations is the frequency resolution.

Since the presence of continuum emission will appear in most channels and complicates the detection of spectral lines in emission (positive in image), a common practice is "continuum subtraction" in which the line free channels are imaged and used to estimate and subtract the continuum emission. If this is done to the visibility data, it leaves a data-set in which only the spectral line emission is present.

On the other hand, if the objective of the observation is to detect absorption by an atomic or molecular species of background continuum emission, continuum subtraction is usually not practical. The present memo considers only spectral lines in emission.

A. CLEANing Spectral Lines

The uv coverage with radio interferometers is always incomplete and uneven. This will lead to artifacts on the point spread function (psf) known as "side-lobes" which depending on the level of the side-lobes and the strength of the emission may corrupt the derived image. Deconvolution, typically CLEAN [4][5], can help recover the actual sky distribution of the emission. Irregardless of the deconvolution technique, it must be nonlinear and always involves multiple iterations. The number of iterations, hence processing cost, depends on the strength and distribution of the emission and can vary widely from channel to channel.

B. Parallel Imaging

In principle, performance gains can be had by processing multiple, independent channels in parallel threads, either in the CPU or a GPU. One simple way of doing this is to image and deconvolve a block of channels at a time. This will only work well when the cost of each channel being processed is roughly the same. In this case, images in each channel in the block are imaged each cycle of the deconvolution even after the CLEAN for that channel was finished. Some operations on "finished" channels can be omitted but not all. The Obit software has been modified to skip operations on channels for which the CLEAN is finished to the extent it is reasonably practical.

Obit spectral line imaging task Imager has a mode for processing multiple channels in parallel; this is indicated by the doLine=True parameter. In doLine model, Imager uses a single thread per parallel channel being imaged. If a single channel is being processed (doLine=False) all of the cores in the host may be used on that channel.

III. Timing Examples

Various timing tests were performed to evaluate the effects of using the GPU for gridding and more generally the parallel processing of multiple channels. The need for CLEANing line channels complicates the analysis.

A. Test Data

The test MeerKAT data-set has 5,860,497 visibilities with 24 spectral channels of continuum subtracted HI data on a nearby galaxy. The emission in individual channels ranged from nothing but noise to bright, widespread emission that requires many major cycles to CLEAN. Imaging used Obit task Imager and was done in Stokes I to a radius of 0.5° using a total of 37 facets. CLEANing used up to 3,000 point components or a minimum residual flux density of 0.5 mJy/beam and autoWindowing [3] to set the CLEAN window. The spectrum of the test data is shown in Figure 1.

National Radio Astronomy Observatory, 520 Edgemont Rd., Charlottesville, VA, 22903 USA email: bcotton@nrao.edu

[1] http://www.cv.nrao.edu/∼bcotton/Obit.html

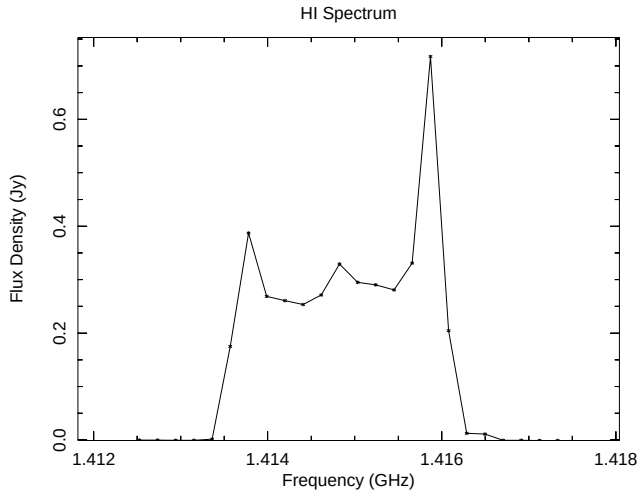

Fig. 1. HI spectrum of the 24 channels in the test data.

B. Test Machine

The test machine is smeagle with 24 (hyper-threaded) cores of Intel Xeon Gold 6136 CPU @3.00 GHz with 256 GByte of RAM, 150 GBytes of which were in a RAM disk for scratch files. The other disk was software RAID-5. Smeagle has a 512 bit memory bus and supports AVX512. This machine has an NVIDIA GeForce RTX 2080 Ti GPU with 68 Multiprocessors with 64 cores each (4352 cores total) and a clock speed of 1508 kHz. The CUDA capability is 7.5 with 10 GByte of global memory.

C. Parallel Channel Imaging Without CLEAN

If no CLEANing is done, imaging of each channel is approximately the same work and processing them in parallel is straightforward. The first set of tests involves investigating the relative efficiency of imaging multiple channels using either CPU or GPU based visibility gridding. Various numbers of channels were imaged in parallel using Obit task Imager. Parallel imaging used doLine=True mode for all but the single channel test. For the CPU gridding tests one thread (core) was used per channel in addition to all 24 for the single channel at a time run. Note: most of the tests using CPU gridding used less that the total compute capability whereas the GPU based gridding tests used the full power of the device. In all cases all 24 channels were imaged. Timing results are shown in Table I.

D. Parallel Channel Imaging With CLEAN

The test in Section III-C was repeated but allowing CLEANing of up to 3000 components or a minimum residual of 0.5 mJy/beam. In each major cycle, all channels in the block being processed were used although some operations, e.g. gridding correction and writing images for channels whose CLEAN was finished were skipped. Processing in each block continued until all channels were finished. Note, the single channel CLEANs used a different mode and had a more relaxed convergence criterium for finishing the CLEAN; a direct comparison of timings can be misleading.

TABLE I
PARALLEL IMAGING ONLY

no. Par	GPU	no. thread	Real sec.
1	N	1	300
1	N	24	136
2	N	2	220
4	N	4	149
8	N	8	88
24	N	24	57
1	Y		118
2	Y		89
4	Y		64
8	Y		58
24	Y		60

Notes:
Column “no. Par” is the number of channels being processed in parallel.
Column “GPU” is ’Y’ is the GPU was used in Gridding of ’N’ if not.
Column “no. thread” gives the number of threads used in CPU based visibility gridding.
Column “Real” is the real (wall clock) time for the run.

TABLE II
PARALLEL CLEAN

no. Par	GPU	no. thread	Real sec.
1	N	1	652
1	N	24	273
2	N	2	445
4	N	4	339
8	N	8	323
24	N	24	600
1	Y		211
2	Y		189
4	Y		188
8	Y		228
24	Y		575

Notes:
Column “no. Par” is the number of channels being processed in parallel.
Column “GPU” is ’Y’ is the GPU was used in Gridding of ’F’ if not.
Column “no. thread” gives the number of threads used in CPU based visibility gridding.
Column “Real” is the real (wall clock) time for the run.

During the CLEAN, GPU “degridding” was used in all cases; the difference between “GPU” and “CPU” is in which was used for gridding. In all cases all 24 channels were imaged. In this test, between 1 and 18 major CLEAN cycles were needed for the various channels. Timing results are shown in Table II.

IV. DISCUSSION

A number of trials were run testing the feasibility of processing multiple spectral channels in parallel and comparing the use of multi-threaded CPU and GPU based gridding. Imaging of spectral channels is independent of each other so

relatively simple to run in parallel. The Obit implementation of GPU based gridding [1] allows parallel gridding onto multiple spectral channels; this enables the full power of the device to be applied. The multi-threaded CPU approach is to use a single CPU core (thread) per channel being imaged. This allows a relatively efficient usage of the resources if a number of channels equal to the number of available cores is imaged in parallel.

There are two cases of interest, 1) when the dynamic range is low enough that deconvolution (CLEAN) of the psf from the dirty images is not needed and 2) when deconvolution is needed. Spectral line imaging frequently involves spectra which can vary widely from channel to channel. The spectrum of the test data used in this memo is shown in Figure 1. In this case, the emission per channel ranges from none (no deconvolution needed) to a lot (extensive deconvolution needed).

In the first case (no deconvolution), the amount of work needed per channel is approximately the same, simplifying parallel processing. Table I shows performance increasing with number of channels processed in parallel although this appears to saturate for GPU gridding.

If deconvolution is needed, then the amount of work varies from channel to channel. The commonly used CLEAN [4][5] algorithm involves multiple "major cycles", the number of which depends on the strength and distribution of the emission. For the deconvolution described in Section III-D, between 1 and 18 major cycles were needed. If multiple channels are being processed in parallel, at least some of the processing is needed for all channels including those for which the CLEAN is completed. This extra processing can reduce the efficiency of parallel CLEANing as can be seen in Table II. When more than a few channels are processed in parallel, the total run time goes up. In the case of CPU based gridding, this is partially overcome by allowing an increasing number of threads. However, for CPU based gridding, Table II shows that the most efficient use of the resources is to apply all threads to a single channel. There is a minor improvement in the speed of GPU based gridding of processing a few channels (e.g. 4) in parallel.

A comparison of the run times for the single channel tests with 24 CPU threads in Tables I and II suggest that using the GPU for gridding is about 20% faster than CPU cores in this test case. This is likely partially the result of the relatively low fraction of the total computations being used in the gridding. Due to the details of the Obit parallel channel implementation (1 thread per channel), using the GPU gives a substantial improvement as all of its power can be used in all cases whereas the number of CPU cores used is limited to the number of parallel channels. This is especially true in those cases where CLEANing is needed.

REFERENCES

[1] W. D. Cotton, "GPU-Based Visibility Gridding for Faceting," *Obit Development Memo Series*, vol. 73, pp. 1–4, 2022. [Online]. Available: https://www.cv.nrao.edu/∼bcotton/ObitDoc/GPUGridv2.pdf
[2] W. D. Cotton, "Obit: A Development Environment for Astronomical Algorithms," *PASP*, vol. 120, pp. 439–448, 2008.
[3] W. D. Cotton, "Performance Enhancement of the autoWindow technique," *Obit Development Memo Series*, vol. 9, pp. 1–3, 2009. [Online]. Available: https://www.cv.nrao.edu/∼bcotton/ObitDoc/autoWin2.pdf
[4] J. A. Högbom, "Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines," *A&A Suppl.*, vol. 15, p. 417, Jun. 1974.
[5] B. G. Clark, "An efficient implementation of the algorithm 'CLEAN'," *A&A*, vol. 89, pp. 377–+, Sep. 1980.