Efficacy of Obit Threading on an EVLA Dataset

W. D. Cotton, R. Perley October 22, 2010

***Abstract*—This memo explores the efficacy of the multi-threaded operation of the Obit wide bandwidth imager MFImage in realistic tests on an EVLA dataset including multiple iterations of self–calibration. These tests were performed on a dual hex core 2.7 GHz processor machine using the lustre file system. The data set, after initial averaging in the imager fits in the 24 GByte memory. There is a nearly linear improvement in performance with up to 10 parallel threads but performance is starting to saturate at 12 threads. 12 threads give a runtime reduction of a factor of 6.5 over single threaded operation.**

***Index Terms*—Computing efficiency, astronomical images**

I. INTRODUCTION

MODERN CPU chips may incorporate multiple processors with multiple “cores” which if efficiently used can greatly speed the performance of a program by enabling multiple, parallel “threads” of processing. How much “multi-threading” speeds (or slows) a given process depends on a number of things such as the fraction of the work susceptible to multi-threading, memory bandwidth, the overhead of setting up work packages for threads and the cache hit ratio [1] of the operations performed in parallel threads. Large memory requirements in each thread can damage the cache hit ratio in each thread to the point that multi-threading can actually take longer. Thus, evaluation of multi-threading techniques require testing under realistic circumstances.

This memo explores the threading uses in the Obit package ([1], http://www.cv.nrao.edu/∼bcotton/Obit.html) to do wide-band imaging of a recent EVLA dataset. This test uses multiple iterations of self calibration and produces images of brightness, spectral index and spectral curvature over a combined C and X band dataset. The threading implementation follows the general method of multi–threading in Obit as described in [2].

II. EVLA WIDE-BAND, WIED-FIELD IMAGING

The most important expansion of the EVLA over the VLA is an increase in bandwidth. The wider bandwidth improves sensitivity but the channelization needed to reduce bandwidth smearing greatly increases the data volume and the fractional bandpass may become wide enough that the sky brightness can no longer be considered constant across the band. The added sensitivity of the EVLA means that more of the primary beam will have to be imaged much of the time at the lower frequencies. The extra computing needed for the increased data volume can partially be accommodated using multi-threading as discussed in [2]. Spectrally sensitive imaging is discussed in [3] and the extensions needed for self–calibration in [4].

The basic wide-band imaging scheme is to divide the band into a number of groups of channels which are individually imaged and jointly deconvolved. This joint deconvolution produces a set of CLEAN components which contain spectral information which is used to the spectrally sensitive model calculations employed in the deconvolution and self–calibration. Wide fields of view are acheived by “fly’s-eye” faceting.

III. TEST DATASET, IMAGING

The data used four 1 GHz bands in C band and one 1 GHz band in X band covering the frequency range 4.0 to 9.0 GHz observing the radio galaxy Hercules A. These observations were made using 2 frequency settings in C band and one in X band. They were individually calibrated in AIPS, averaged to 10 seconds and combined into a single dataset using AIPS/VBGLU. This produces a dataset with 248,192 visibilities of 40 “IFs” with 64 spectral channels each in dual polarization. The lowest two IFs were severely affected by RFI and were dropped. The remaining RFI was flagged using Obit/AutoFlag. The full dataset is 29 GByte in size.

Imaging used Obit task MFImage to solve for brightness, spectral index and spectral curvature over the observed band. The initial step of imaging was to do baseline dependent imaging which reduced the data to 136,321 visibilities. Imaging used autoWindowing [5] with 5,000 components and a CLEAN gain of 0.05. The EVLA was in D configuration and the resulting image had 12” resolution. A 512×512 image with 1” cells with 38 sub-band images was used. The image quality is limited by atmospheric and instrumental artifacts so 3 cycles of phase self–calibration were followed by a single amplitude and phase self–calibration. The resultant images are shown in Figure 1.

IV. TIMING TEST

To evaluate the effectiveness of Obit threading, MFImage was run on the same dataset multiple times changing only the maximum number of threads it was allowed to use. All runs used the same executable on the same machine, NRAO/AOC cv-pipe-a. This machine is a dual hex core 2.7 GHz Intel Xenon machine with 24 GByte of memory and is attached to the lustre file system which was used for all I/O. After the initial data averaging, the dataset fit in memory. There were no other users on the machine at the time. Each run of MFImage reported its total real time and CPU usage.

The results of the timing tests are shown in Table I and Figure 2. This table gives the number of threads used, the total wall clock execution time, the total CPU time and the ratio of the wall clock time used to that for a single thread.

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

Manuscript received ; revised

---

[1] The cache hit ratio is the fraction of the time memory access is satisfied by a value already in memory cache; access to cache memory is much faster than main memory.

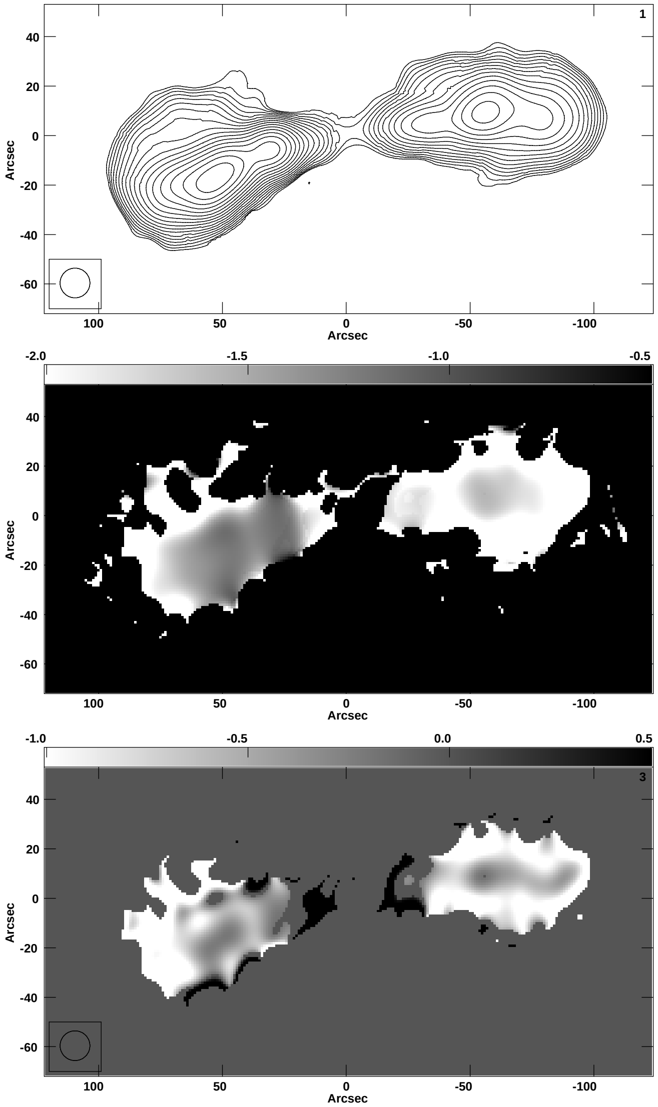

Fig. 1. **Top:** Total intensity at the reference frequency of 6.6 GHz. Contours are $\sqrt{(2)} \times 0.002$ Jy/beam; negative coutours dashed. Resolution shown in lower left.
Middle: Image of spectral index at reference frequency, range of values displayed is -0.5 to -2.0 given by scalebar.
Bottom: Image of spectral curvature at reference frequency, range of values displayed is -2.0 to 0.5 given by scalebar.

TABLE I
TIMING TESTS OF MFIMAGE

# threads	Run time (hrs)	CPU time (hrs)	Rel. Speed
1	12.88	12.78	1.000
2	7.44	14.07	1.73
4	4.30	14.71	3.00
6	3.15	15.17	4.08
8	2.56	15.10	5.03
10	2.08	15.17	6.18
12	1.98	15.42	6.50

V. DISCUSSION

The execution time of MFImage was dominated by two operations, calculation of the instrumental response to the sky model (CLEAN components) and to a much lesser degree, griding the data onto the uv grids to be FFTed. For both of these operations, the threading is done by dividing the I/O buffer among the various threads as described in [2]. The glib library thread groups were used to reduce the overhead of starting and stopping threads by keeping a group of threads running and sending them packets of work. The size of the I/O buffer used was scaled by the number of threads to be used to keep the size of the packet of work given to each thread constant. As the number of threads is increased, the number of work packets sent to each thread in a given operation decreases, increasing the relative cost of starting the thread.

Actual I/O is not a serious issue in this test as the datasets being operated on fit in memory. Use of the lustre file system minimized the time needed for I/O that did occur.

As can be seen in Table I, thread management consumed a fair amount of resources. The one thread test spawned no

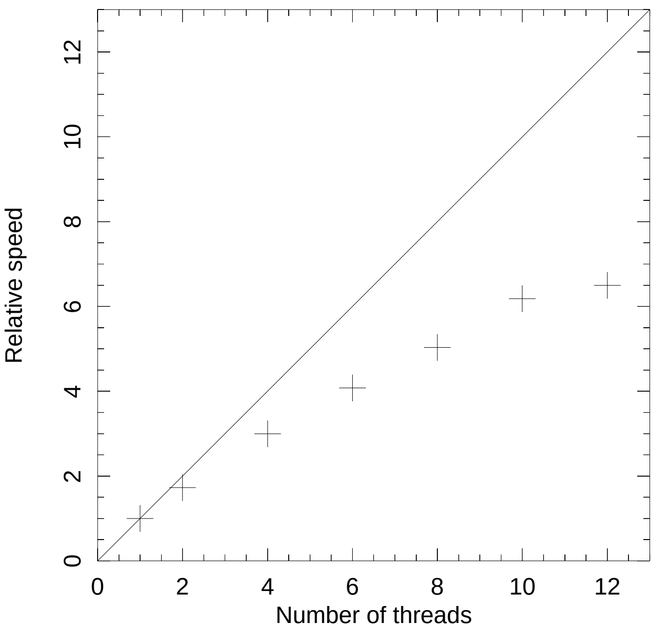

Fig. 2. Processing speed relative to a single processor. The solid line indicates complete utilization of the processors.

additional threads and its CPU time can be taken as the amount of computing required by the algorithm. For the 12 thread case, over 20% of the CPU time appears to be thread management overhead. Had glib thread groups, or the equivalent, not been used this would likely be far higher.

The results given above demonstrate a significant improvement in the speed of processing of a fairly compute intensive problem by increasing the number of cores used up to 10; larger numbers of cores show signs of saturation. With 12 threads, the run time was reduced by a factor of 6.50 over the single thread run time. The relatively linear increase in performance with increasing number of threads shown in Figure 2 prior to saturation indicates that the fraction of the work which was run in parallel was very large.

Figure 2 shows a decreasing benefit with the addition of cores, approaching saturation at 12 cores. This is likely a combination of limited memory bandwidth, Amdahl's Law (with increasing parallelization, serial operations become relatively more important), fewer work packets per thread per operation increasing the overhead, and a reduction in the cache hit ratio coming from sharing the finite memory cache among more processors.

The basic conclusion is that threading in the form used in Obit can significantly improve software performance on EVLA class problems. Performance is nearing saturation on a dual processor, 12 core machine.

ACKNOWLEDGMENT

The authors would like to thank Ed Fomalont for helpful comments.

REFERENCES

[1] W. D. Cotton, "Obit: A Development Environment for Astronomical Algorithms," *PASP*, vol. 120, pp. 439–448, 2008.
[2] ——, "Note on the Efficacy of Multi-threading in Obit," *Obit Development Memo*, no. 1, 2008.
[3] ——, "High dynamic range wideband imaging." *Obit Development Memo*, no. 19, 2008.
[4] ——, "Wideband phase and delay calibration." *Obit Development Memo*, no. 20, 2008.
[5] ——, "Automatic CLEAN Windowing," *EVLA Memo*, no. 116, 2007.