Mon. Not. R. Astron. Soc. **000**, 1–4 (2019) Printed 18 March 2019 (MN LATEX style file v2.2)

Load balancing for distributed interferometric image reconstruction

Luke Pratley[1⋆], Jason D. McEwen[1]

[1] *Mullard Space Science Laboratory (MSSL), University College London (UCL), Holmbury St Mary, Surrey RH5 6NT, UK*

Accepted —. Received —; in original form —

ABSTRACT

We present a new algorithm to perform wide-field radio interferometric image reconstruction, with exact non-coplanar correction, that scales to big-data. This algorithm allows us to image 2 billion visibilities on 50 nodes of a computing cluster for a 25 by 25 degree field of view, in a little over an hour. We build on the recently developed distributed w-stacking w-projection hybrid algorithm, extending it to include a new distributed degridding algorithm that balances the computational load of the w-projection gridding kernels. The implementation of our algorithm is made publicly available in the PURIFY software package. Wide-field image reconstruction for data sets of this size cannot be performed effectively using the allocated computational resources without computational load balancing, demonstrating that our algorithms are critical for next-generation wide-field radio interferometers.

Key words: techniques: image processing - techniques: interferometric

1 INTRODUCTION

Next-generation low frequency wide-field of view telescopes, such as the Murchison Widefield Array (MWA; Tingay et al. 2013), have non-coplanar baselines and other instrumental effects that need to be modeled during image reconstruction. Furthermore, the large volumes of visibilities and large image sizes increase the computational burden of imaging observations from next-generation telescopes. However, there are major computational and instrumental challenges that need to be overcome for these telescopes to reach the high resolution and sensitivity required by science goals of next-generation telescopes, such as detection of the epoch of reionization (EoR) (Koopmans et al. 2015) and to probe Galactic and extra-galactic magnetic fields.

Recent novel developments in fast construction of w-projection kernels and the distributed w-stacking w-projection hybrid algorithm (Cornwell et al. 2008; Pratley et al. 2019a) has allowed fast and accurate modeling of non-coplanar effects over extremely wide-fields of view from the MWA for over 100 million measurements (Pratley et al. 2019b). The algorithm allows parallel construction of w-projection kernels while also distributing their storage for application, proving to be an effective method of tackling the most computational and memory intensive components of radio interferometric imaging (Wortmann 2016; Braam & Wortmann 2016; Hollitt et al. 2017; Pratley et al. 2019a). However, while this distribution reduces the size and computational cost of the w-projection kernel, it does not ensure that computational resources are being used most effectively across the compute cluster. This makes it vulnerable to bottlenecks in computation without the modifications presented in this work.

This work presents a new distributed gridding algorithm that evenly balances the computational load across a computing cluster, extending the distributed gridding methods developed in Pratley et al. (2019c). Such an approach allows full memory and computational use across the nodes of the computing cluster when performing fast Fourier transforms (FFTs) of w-stacks and when degridding with w-projection kernels, which has not been possible previously, removing resource bottlenecks when imaging wide-fields of view for large data sets. Such distributed degridding and gridding algorithms will be vital for next-generation radio interferometers with large data sets, such as the Square Kilometer Array (SKA). In particular, such an algorithm is needed for effectively correcting instrumental effects via the image and Fourier domain, while using the full performance of a computing cluster.

The remaining sections of this article are as follows. Section 2 introduces the wide-field interferometric measurement equation and the distributed w-stacking w-projection hybrid algorithm. Section 3 discusses the computational and memory bottlenecks of the distribution method. Section 4 presents the new algorithm that evenly distributes the computational load across compute nodes. Section 5 demonstrates the application of this algorithm that has been implemented in the interferometric imaging software package PURIFY[1].

⋆ E-mail: Luke.Pratley@gmail.com

[1] https://github.com/astro-informatics/purify

© 2019 RAS

2 WIDE-FIELD IMAGING MEASUREMENT EQUATION

The interferometric measurement equation is a result of the van Cittert-Zernike theorem (Zernike 1938) and it can be extended to include many aspects of the measurement process (Smirnov 2011). One simplified variation is the non-coplanar wide-field interferometric measurement equation, it reads

$$y(u,v,w') = \int x(l,m)a(l,m)\frac{\mathrm{e}^{-2\pi i w'(\sqrt{1-l^2-m^2}-1)}}{\sqrt{1-l^2-m^2}} \times \mathrm{e}^{-2\pi i(lu+mv)}\,\mathrm{d}l\mathrm{d}m\,, \tag{1}$$

where (u,v,w') are the baseline coordinates and (l,m,n) are directional cosines restricted to the unit sphere. In this work, we define $w' = w + \bar{w}$, where $\bar{w}$ is the average value of w-terms, and w is the effective w-component (with zero mean), x is the sky brightness and a includes direction dependent effects such as the primary beam. The measurement equation allows one to calculate model measurements y when provided with a sky model x.

A number of methods make use of the measurement equation to recover an image x given visibilities y. Two examples in radio astronomy are CLEAN (Högbom 1974; Pratley & Johnston-Hollitt 2016) and Sparse Regularization algorithms (McEwen & Wiaux 2011; Onose et al. 2016; Pratley et al. 2018; Dabbech et al. 2018; Pratley et al. 2019a,c).

To make use of the FFT the measurement equation is traditionally calculated and approximated using degridding (Thompson et al. 2008). The measurement equation can be represented by the following linear operations

$$\boldsymbol{y} = \mathbf{W}[\mathbf{GC}]\mathbf{FZS}\boldsymbol{x}\,. \tag{2}$$

$\mathbf{S}$ represents a gridding correction and correction of baseline independent effects such as $\bar{w}$, $\mathbf{Z}$ represents zero padding to increase resolution of the Fourier grid, $\mathbf{F}$ is an FFT, $\mathbf{G}$ represents a sparse circular convolution matrix that interpolates measurements off the grid, while $[\mathbf{GC}]$ corrects baseline dependent effects and interpolates measurements off the grid, and $\mathbf{W}$ are weights applied to the measurements. This linear operator is typically called a measurement operator $\mathbf{\Phi} = \mathbf{WGCFZS}$ with $\mathbf{\Phi} \in \mathbb{C}^{M\times N}$. Furthermore, $\boldsymbol{x}_i = x(\boldsymbol{l}_i)$ and $\boldsymbol{y}_q = y(\boldsymbol{u}_q)$ are discrete vectors in $\mathbb{R}^{N\times 1}$ and $\mathbb{C}^{M\times 1}$ in this setting. The dirty map can be calculated from the adjoint operation $\mathbf{\Phi}^\dagger\boldsymbol{y}$, and the residual map by $\mathbf{\Phi}^\dagger(\mathbf{\Phi}\boldsymbol{x} - \boldsymbol{y})$.

2.1 Distributed wide-field measurement operator

In the distributed w-stacking w-projection algorithm (Pratley et al. 2019a), the measurement operator corrects for the average w-value in each w-stack, then applies an extra correction to each visibility with the w-projection. Each w-stack $\boldsymbol{y}_k$ has the measurement operator of

$$\mathbf{\Phi}_k = \mathbf{W}_k[\mathbf{GC}]_k\mathbf{FZ}\tilde{\mathbf{S}}_k\,. \tag{3}$$

The gridding correction, $\tilde{\mathbf{S}}_k$, has been modified to correct for the w-stack dependent effects, such as the average w-value of the stack $\bar{w}_k$

$$\tilde{\mathbf{S}}_{kii} = \frac{a_k(l_i,m_i)\mathrm{e}^{-2\pi i\bar{w}_k(\sqrt{1-l_i^2-m_i^2}-1)}}{g(l_i^2+m_i^2)\sqrt{1-l_i^2-m_i^2}}\,. \tag{4}$$

We leave the option of choosing different primary beam effects in a stack $a_k(l_i,m_i)$. The chirp shifts the relative w-value in the stack indexed by k. The stacks can be clustered carefully to reduce the effective w-value in the stack, especially when the stack is close to the mean $\bar{w}_k$, i.e. to the value of $w_i - \bar{w}_k$. This reduces the size of the support needed in the w-projection gridding kernel for each stack,

$$[\mathbf{GC}]_{kip} = [GC]\Big(\sqrt{(u_i/\Delta u - q_{u,p})^2 + (v_i/\Delta u - q_{v,p})^2}\,, w_i - \bar{w}_k, \Delta u\Big)\,. \tag{5}$$

$(q_{u,p}, q_{v,p})$ represents the nearest grid points, and we use adaptive quadrature to calculate

$$[GC]\Big(\sqrt{u_{\mathrm{pix}}^2 + v_{\mathrm{pix}}^2}, w, \Delta u\Big) = \frac{2\pi}{\Delta u^2}\int_0^{\alpha/2} g(r) \times \mathrm{e}^{-2\pi i w(\sqrt{1-r^2/\Delta u^2}-1)} J_0\left(2\pi r\sqrt{u_{\mathrm{pix}}^2 + v_{\mathrm{pix}}^2}\right) r\mathrm{d}r\,, \tag{6}$$

where $g(r)$ is the radial anti-aliasing filter in the image domain (i.e. the Fourier transform of the Kaiser-Bessel function), Δu is the resolution of the Fourier grid as determined by the zero padded field of view, and $(u_{\mathrm{pix}}, v_{\mathrm{pix}})$ are the pixel coordinates on the Fourier grid.

For each stack $\boldsymbol{y}_k \in \mathbb{C}^{M_k}$ we have the measurement equation $\boldsymbol{y}_k = \mathbf{\Phi}_k\boldsymbol{x}$. It is clear that each stack has an independent measurement equation. However, the full measurement operator is related to the stacks in the adjoint operators such that

$$\boldsymbol{x}_{\mathrm{dirty}} = \mathbf{AllSumAll}_k\left(\mathbf{\Phi}_k^\dagger\boldsymbol{y}_k\right) = \mathbf{\Phi}^\dagger\boldsymbol{y}\,. \tag{7}$$

We use an MPI all-sum-all to generate the same dirty map on each node. The full operator MPI $\mathbf{\Phi}$ is normalized using the power method. For further details see Pratley et al. (2019a).

3 BOTTLENECK OF THE DISTRIBUTED STACKING METHOD

To minimize the time taken to perform kernel calculation and increase accuracy of the non-coplanar correction, the visibilities need to be sorted into w-stacks using a cluster algorithm. We do this by using the k-means clustering algorithm after using complex conjugation to reflect the visibilities to have positive w (Pratley et al. 2019b). Because the w-stacks are clustered to minimize error, the memory and computational load of each $[\mathbf{GC}]_k$ has previously been ignored when assigning one stack k per compute node. When the majority of visibilities lie in only a few stacks, the total available memory and resources for construction and application of $[\mathbf{GC}]_k$ is bottlenecked. This is especially the case when there is one $[\mathbf{GC}]_k$ per MPI node. This problem is emphasized for extremely wide-fields of view and large values of w, where the w-projection kernel size scales as $\frac{2w}{\Delta u}$, with $\frac{1}{\Delta u} \propto$ field of view, and for large numbers of visibilities. Hence, these factors have a large impact on the required computational resources in kernel construction and application, as we demonstrate in Section 5.

In the next section we describe an algorithm that solves this bottleneck. We split the operator $[\mathbf{GC}]_k$ into smaller operators $[\mathbf{GC}]_{jk}$ that can be spread across multiple nodes j for w-stacks indexed by k. We remove the requirement that image domain correction and Fourier domain correction are applied on the same node. We restrict the index j for nodes that apply Fourier domain correction and index k for nodes that apply image domain correction. This allows even distribution of the memory load, kernel construction, and application of the operator $[\mathbf{GC}]$ to ensure scalability as demonstrated in Section 5.

4 ALL-TO-ALL DISTRIBUTED MEASUREMENT OPERATOR

In this section we introduce a new MPI distribution strategy for the application of a wide-field measurement operator. This process allows the FFTs of the w-stacks to be evenly distributed across all nodes while allowing the sparse matrix operations to be distributed evenly across all nodes. Communicating only the grid points that are needed for degridding minimizes communication in an intermediate all-to-all operation.

4.1 Distributing measurements for computational load

First the k-means algorithm is used to sort the visibilities into w-stacks $\boldsymbol{y}_k$. The visibilities of each stack $\boldsymbol{y}_k$ are distributed across MPI nodes $\boldsymbol{y}_{jk}$, where $1 \leq j \leq n_\mathrm{d}$, to evenly distribute the computation of $[\mathbf{GC}]$. The computational load of an individual visibility $\boldsymbol{y}_{\mathbf{k}_i}$ is determined by the support size

$$\mathrm{support}(w_i - \bar{w}_k, \Delta u) = \max\{J_\mathrm{min}, 2(w_i - \bar{w}_k)/\Delta u\}\,, \quad (8)$$

where J_min is the 1d support size of the anti-aliasing kernel (Pratley et al. 2019a). It is then straightforward to determine the total computational load of $[\mathbf{GC}]$ and then distribute it evenly across nodes j. This is done by calculating the average computational load across all nodes from $j = 1$ to $j = n_\mathrm{d}$ in order, filling each node j with visibilities until it reaches the average computational load.

In practice, it is difficult to fill each node with the exact average computational load, because each visibility has its own integral (indivisible) computational load. This can be accommodated by allowing the last node to overfill slightly and keeping the rest of the nodes under the average load. Testing has shown that the overfill amount on the last node is insignificant.

4.2 All-to-all distribution of Fourier grid subsections

With the computational load of $[\mathbf{GC}]$ distributed across the nodes, the measurement equation needs to map sections of the grid that need to be sent to each node j from each stack k to minimize communication. Without loss of generality, we let $1 \leq k, j \leq n_\mathrm{d}$. The MPI measurement equation reads

$$\boldsymbol{y}_{jk} = \mathbf{W}_{jk}[\mathbf{GC}]_{jk}\mathbf{AllToAll}_{jk}\left(\mathbf{M}_{jk}\mathbf{F}\mathbf{Z}\tilde{\mathbf{S}}_k\boldsymbol{x}\right)\,, \quad (9)$$

where the chirp multiplication and FFT are applied on node k (assuming one $\tilde{\mathbf{S}}_\mathbf{k}$ per node for simplicity), the operator $\mathbf{M}_{jk} \in \mathbb{R}^{K_{jk}\times K}$ selects only the grid sections (of size K_j) of the FFT grid (of size K) of stack k that are needed for degridding on node j, which are then sent to node j with the MPI all-to-all operation. This is followed by degridding to the visibilities on node j that belong to stack k using $[\mathbf{GC}]_{jk} \in \mathbb{C}^{M_{jk}\times K_{jk}}$. In practice, $[\mathbf{GC}]_{jk}$ are combined into one sparse matrix on each node that has $\sum_k M_{jk}$ rows and $\sum_j K_{jk}$ columns. This entire process is visualized in Figure 1.

The application of the adjoint operator reads

$$\boldsymbol{x}_\mathrm{dirty} = \mathbf{AllSumAll}_k\Big(\tilde{\mathbf{S}}_k^\dagger\mathbf{Z}^\dagger\mathbf{F}^\dagger\times \sum_{j=1}^{n_\mathrm{d}}\left[\mathbf{M}_{jk}^\dagger\mathbf{AllToAll}_{kj}\left([\mathbf{GC}]_{jk}^\dagger\mathbf{W}_{jk}^\dagger\boldsymbol{y}_{jk}\right)\right]\Big)\,, \quad (10)$$

where node j grids visibilities from stack k, these grid sections are sent from node j to stack k through an all-to-all operation. The grid sections from each node j are added to the full FFT grid of each stack k. An inverse FFT is applied followed by cropping of the image. Multiplication of the conjugate chirp is applied on each stack k followed by an all-sum-all of the images to produce the same dirty map on each MPI node.

Extensive unit testing has shown that the distributed computation is equivalent to the non distributed computation and the standard w-stacking w-projection algorithm. It is worth noting that when $n_\mathrm{d} \times K > 2^{32} - 1$, 64 bit integer types are specifically needed for indexing across $n_\mathrm{d} \times K$ FFT pixels without overflow.

5 IMPLEMENTATION AND APPLICATION

In this section we demonstrate the effectiveness of evenly distributing the computational load using the algorithm presented in Section 4. This algorithm has been implemented in the interferometric imaging software package PURIFY using C++ and MPI. PURIFY is powered by distributed convex optimisation algorithms implemented in the software package SOPT[2].

To demonstrate the effectiveness of the algorithm, we simulate reconstruction of a 25 by 25 deg field of view, using a Gaussian variable sampling density in uvw following Pratley et al. (2018). u and v are represented in radians, with a standard deviation of $\pi/3$. w is represented in wavelengths, with a standard deviation of 200 wavelengths, but was constrained to values between ± 600 wavelengths. An 1024 by 1024 pixel image of M31 is considered, where the pixel size is 90 by 90 arcseconds. We add Gaussian noise to the measurements, so that the visibilities have an input signal to noise ratio of 30 decibels Pratley et al. (2018). We then apply the alternating direction method of multipliers (ADMM) algorithm as performed in Pratley et al. (2018, 2019a,c). We used a minimal gridding kernel support size of $J_\mathrm{min} = 4$ for the Kaiser-Bessel kernel.

First we use conjugate symmetry to reflect the visibilities to have $w \geq 0$. Then we use the k-means clustering algorithm to assign each visibility to a w-stack indexed by k and to calculate each $\bar{w}_k$. Then we iterate through the visibilities to assign the computational load across the nodes, following Section 4. The visibilities and w-stack indexes are redistributed using an all-to-all operation. Then the w-projection kernels shown in Equation 6 are constructed using adaptive quadrature to an accuracy of 10^{-6} in absolute and relative error, which has shown to be accurate to 1% in the image domain (Pratley et al. 2019a). This corrects each visibility for the w offset determined by $\bar{w}_k$ and the w-stack index k.

We perform reconstruction using 2 billion visibilities with 50 nodes of the Grace supercomputing cluster at University College London (UCL). Each node has two 8 core Intel Xeon E5-2630v3 processors and 64 Gigabytes of RAM[3]. Note that this is exactly the same configuration used in the recent work of Pratley et al. 2019b, where an MWA Fornax A observation was reconstructed using 126 million visibilities.

The memory used to store $[\mathbf{GC}]$ is distributed across 50 compute nodes. The memory needed to store $[\mathbf{GC}]$ was approximately 21 Gigabytes on each node (3 Tb across all nodes). However, for efficient layout for memory access $[\mathbf{GC}]^\dagger$ was also stored, requiring an additional 3 Tb across all nodes. The 2 billion visibilities amounts to 32 Gigabytes spread evenly across the nodes. To store the weights and uvw-coordinates during construction of $[\mathbf{GC}]$ requires 64 Gigabytes of memory spread evenly over the cluster.

[2] https://github.com/astro-informatics/sopt
[3] More details can be found at https://wiki.rc.ucl.ac.uk/wiki/RC_Systems#Grace_technical_specs

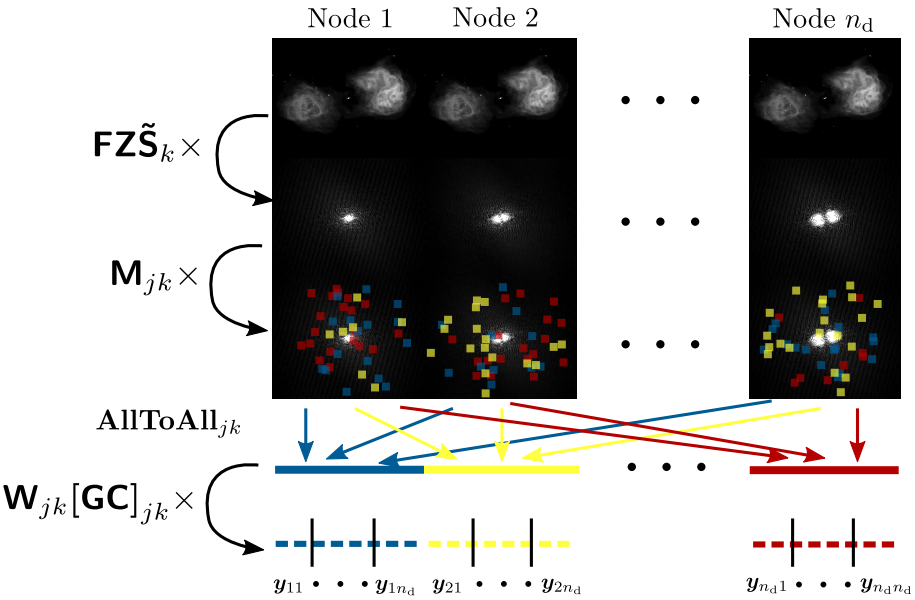

Figure 1. Each node starts with a copy of $\boldsymbol{x}$. The linear operation $\tilde{\mathbf{S}}_k$ applies the gridding correction and multiplication of the chirp on node k. Each node performs zero padding and an FFT with the operation $\mathbf{FZ}$. The operation $\mathbf{M}_{jk}$ selects sections of the FFT grid on node k that are required on node j for degridding (this is determined by the columns of $[\mathbf{GC}]_{jk}$). The colored squares show regions of the grid that are to be sent to each node, with each color corresponding to a value of j. The sections of the FFT grid are distributed through a distributed MPI all-to-all communication step. This is followed by the application of $[\mathbf{GC}]_{jk}$ for the k^{th} w-stack on node j, to interpolate the visibilities $\boldsymbol{y}_{jk}$ off of the grid, with the w-projection kernel performing the correction for the offset $w - \bar{w}_k$. The adjoint process corresponds to performing each step in reverse, followed by an all-sum-all operation over the w-stacks.

Sorting and distributing the visibilities took approximately 2 minutes. Kernel construction took 1 hour and 5 minutes. Application of the combined gridding and degridding operation took approximately 25 seconds. The ADMM algorithm converged in approximately 20 minutes with 9 iterations. The signal to noise ratio of the reconstruction was calculated as in Pratley et al. (2018) to be 31.49 decibels.

Applying the standard distribution method of the w-stacking w-projection hybrid algorithm was not possible for the scenario considered due to memory requirements, where each $[\mathbf{GC}]_k$ requires approximately 1 to 50 Gigabytes of memory. Additionally, even if there was enough memory on each node, run time would increase greatly due to lack of CPU cores on the heavily loaded nodes acting as a bottleneck. It is clear that the distribution method presented in this work circumvents this bottleneck in resources and enables accurate interferometric image reconstruction over extremely wide-fields of view for very large data sets.

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) through grant EP/M011089/1.

REFERENCES

Braam P., Wortmann P., 2016, Kernel Prototyping SOW, SKA SDP SCIENTIFIC MEMORANDUM

Cornwell T.J., Golap K., Bhatnagar S., 2008, IEEE Journal of Selected Topics in Signal Processing, 2, 647, 0807.4161

Dabbech A., Onose A., Abdulaziz A., Perley R.A., Smirnov O.M., Wiaux Y., 2018, MNRAS, 476, 2853, 1710.08810

Högbom J.A., 1974, A&AS, 15, 417

Hollitt C., et al., 2017, in N.P.F. Lorente, K. Shortridge, R. Wayth, editors, *Astronomical Data Analysis Software and Systems XXV*, volume 512 of *Astronomical Society of the Pacific Conference Series*, 367, 1601.04113

Koopmans L., et al., 2015, Advancing Astrophysics with the Square Kilometre Array (AASKA14), 1, 1505.07568

McEwen J.D., Wiaux Y., 2011, MNRAS, 413, 1318, 1010.3658

Onose A., Carrillo R.E., Repetti A., McEwen J.D., Thiran J.P., Pesquet J.C., Wiaux Y., 2016, MNRAS, 462, 4314, 1601.04026

Pratley L., Johnston-Hollitt M., 2016, MNRAS, 462, 3483, 1606.01482

Pratley L., Johnston-Hollitt M., McEwen J.D., 2019a, ApJ, in press, 1807.09239

Pratley L., Johnston-Hollitt M., McEwen J.D., 2019b, PASA, submitted, 1903.06555

Pratley L., McEwen J.D., d'Avezac M., Cai X., Perez-Suarez D., Christidi I., Guichard R., 2019c, Astronomy and Computing, submitted, 1903.04502

Pratley L., McEwen J.D., d'Avezac M., Carrillo R.E., Onose A., Wiaux Y., 2018, MNRAS, 473, 1038, 1610.02400

Smirnov O.M., 2011, A&A, 531, A159, 1106.0579

Thompson A.R., Moran J., Swenson G., 2008, *Interferometry and Synthesis in Radio Astronomy*, Wiley, ISBN 9783527617852

Tingay S.J., et al., 2013, PASA, 30, e007, 1206.6945

Wortmann P., 2016, Gridding Computational Intensity, SKA SDP SCIENTIFIC MEMORANDUM 28

Zernike F., 1938, Physica, 5, 785