SKA-TEL.SDP.PIP.IMG MEMO

Analysing memory requirements for potential SKA deconvolution algorithms

Document number	SKA-TEL.SDP.PIP.MemReq
Authors	D. Fenech, J. McEwen
Version	0.2
Release Date	18 December 2014

Name	Designation	Date	Signature
D. Fenech	Lead Author	20/10/2014	
J. McEwen	Co-Author	20/10/2014	

Version	Date of Issue	Prepared by	Comments
0.1	03/09/2014	D. Fenech/ J. McEwen	First draft
0.2	18/12/2014	D. Fenech/ J. McEwen	CS alogrithms added

Introduction

The expected computing system architecture for data calibration and imaging will include a distribution of networked nodes (islands) which will perform the imaging and deconvolution in parallel. As part of the analysis of the applicability of the available algorithms to the SKA, this memo aims to assess the memory requirements of these algorithms to determine their compatibility with the planned architecture.

Following from the SKA-TEL.SDP.PIP.IMG-Memo-003 outlining the algorithms and implementations, a subset of these have been chosen that are/will be publicly available within a suitable time frame and are scalable to the requirements of the SKA. Each of this subset of algorithms will eventually be discussed individually within this document and information concerning the computational and local memory requirements will be detailed. In this draft we focus on the multi-scale multi-frequency-synthesis CLEAN as well as a generalised discussion of the compressive sensing algorithms SARA and the LOFAR CS implementations.

All of the algorithms represented here are detailed individually within the SKA-TEL.SDP.PIP.IMG-Memo-FB memo which provides a functional breakdown of each algorithm.

MS-MFS clean

The multi-scale multi-frequency-synthesis (MS-MFS) CLEAN algorithm is reviewed here as described in Rau & Cornwell 2011 (arxiv:1106.2745; hereafter R&C2011). Following R&C2011, N_{vis} (n in R&C2011) visibilities are measured for each frequency channel N_c. A Taylor series expansion is performed with order N_t over frequency, resulting in N_t Taylor expansion images. For each Taylor expansion image, a multi-scale image mode containing N_S scales is assumed. Each image is assumed to contain N_{pix} pixels (m in R&C2011).

Algorithm Overview

MS-MFS CLEAN follows the standard major/minor cycle CLEAN iterative procedure using a multi-scale image model. During the major cycle, the adjoint measurement operator (i.e. operator to transform from visibility to image space) is applied to the residual visibilities to compute the residual image.

This computation of the major cycle can be parallelised in different ways. Here we assume that the adjoint operator applied to the sample of visibilities associated with each frequency channel is independent.

This major cycle computation can then be viewed as an independent application of the adjoint measurement operator for each frequency channel, followed by a reduction step to sum the contributions over frequency channels. The minor cycle computation then acts on the residual image and its multi-scale Taylor expansion. See R&C2011 for further details.

During the major cycle, residual images can be computed for each frequency. This involves a de-gridding, Fourier transform, and projection onto the multi-scale dictionary, for each frequency. The asymptotic complexity of the major cycle computation is thus

$$O\left(N_c \; x \left(N_{vis} \, N_k \; + \; N_{pix} \, log_2 \left(N_{pix}\right) \; + \; N_S \, {N_{pix}}^2\right) \; + N_c \, N_S N_{pix} N_t\right)$$

where N_k is the number of coefficients of the gridding kernel (which may also incorporate A- and/or w-projection). It should be noted that it is possible to improve on the performance of the $N_S \, x \, {N_{pix}}^2$ term by using a multi-scale decomposition that supports fast algorithms or by moving it through the FFT for some specific representations.

This major cycle computation can be performed independently for each frequency and thus can be distributed across N_c nodes of a distributed multi-node system. This is an equivalent process to performing a standard MS CLEAN at a single frequency on each node with equivalent memory requirements. A reduction step follows to sum the contributions over frequency channels, to create N_t Taylor images.

During the minor cycle, a new sky model component is firstly reconstructed from the dirty image residual. This is achieved by solving the normal equation in the image domain. A diagonal approximation of the composite normal equation matrix is made such that this normal equation can be decomposed into separate equations over scales and solved trivially. Secondly, the image models and residuals are updated. The asymptotic complexity of the minor cycle computation is

$$O\left(N_S \, {N_{pix}}^2 \, N_t \; + \; N_S \, N_t^2\right)$$

The minor cycle computation can be parallelised across scales and pixels (depending on the level of approximation assumed for the composite normal equation matrix). However, the number of

computations to be performed in parallel is low, hence only modest gains can be realised by parallelising the minor cycle computation.

For the minor cycle computation, the data structure considered is the multi-scale Taylor expansion images of size $N_S N_{pix} N_t$.

Memory Requirements

During each major cycle, the kernels, frequency dirty images, Taylor coefficient images and visibility samples are to be stored in memory. The total memory requirement will therefore depend on the following parameters.

N_c is the number of frequency channels. At each iteration of the Major cycle the dirty images for each frequency increment are re-created (following solution and subtraction). The expected bandwidth for observations that will require MFS imaging are likely to be 1-2 GHz. The incrementation of this BW is a function of the FOV required, but a typical example will require ~20000 channels, though this could be significantly more (The correlator will be able to output data at least an order of magnitude greater than this).

N_{pix} is the number of pixels per image. The requirement for the Large FOV imaging capability of the SKA will require large imaging arrays to be stored in memory for each Major cycle. For extended fields of view this can be as large as 60k^2 (3.6G) pixels.

N_{vis} is the number of visibilities per frequency. These will be read into memory for gridding etc. at each major cycle. The solutions when found are subtracted directly from the visibility data.

The total visibility memory requirement will be $2N_{freq} x\, N_{vis}\, x\, N_T$ (A factor of two is included here assuming that N_{vis} represents a single observed polarisation e.g. RR, LL, X or Y and that a minimum of stokes I imaging is required). N_T is the number of time integrations made and will therefore be dependent of the time-sampling and the length of observation. Data will typically be observed at < 1 sec integrations.

N_S is the number of scales to be used. Each MFS process will essentially be repeated for each resolution required i.e. each scale, S. The number of resolution elements (N_S) will depend on the expected source structure but can be as many as 12.

$$Total\ image\ memory\ requirement = N_{freq}\, x\, N_{pix}\, x\, N_S$$

Additional images stored in memory include the Taylor coefficient images recreated at each Major cycle from the collection of N_c images.

Computational requirements

Major Cycle

Within each major cycle N_c dirty images are created for one of N_S scales. This process incorporates (de-)gridding, computing the convolution kernels, FFT, UV-subtraction and solving of the matrices.

As discussed previously the MS-MFS algorithm complexity of this cycle is given by

$$O\,(N_c\ x\ (N_{vis}\, N_k\ +\ N_{pix}\, log_2\,(N_{pix})\ +\ N_S\, N_{pix}{}^2)\ +\ N_c\, N_S N_{pix} N_t)$$

The gridding and FFT processes will both contribute a significant portion of the computing overheads for any given major cycle performed within the MS-MFS implementation and will be discussed individually.

Where appropriate FLOP estimates discussed here have been taken from the 2014 SDP Performance Models (B. Nikolic; hereafter Nikolic2014).

Gridding

The total Gridding flop rate will be highly dependent on the choice of gridding kernel e.g. A-/w-projection. In line with Nikolic2014 the gridding flop rate can be estimated as

$$R_{grid} = N_{vis} N_c\, N_t (N^2{}_{kernel})\ x\ 8\ Ops/s$$

Where N_{kernel} is the total linear kernel size and a factor of N_t is included as this is performed for each Taylor coefficient. The factor of 8 Ops/s is taken from Nikolic2014.

FFT

The FFT flop rate is adapted from Nikolic2014 and can be estimated as:

$$R_{FFT} = N_c \; N_{pix} \log_2(N_{pix}) \; x \; 5 \; Ops/s \; .$$

The factor of 5 Ops/s is as determined by Nikolic2014.

Scale convolution

Within the MS-MFS implementation, following the gridding and FFT stages the residual images produced are also convolved with each scale function. Assuming a single floating point operation is required per evaluation the rate for this process will take the order of

$$R_{sc} \; = \; N_S \, N_{pix}{}^2 \; Ops/s$$

The overhead for each evaluation of this process may be greater than 1 Ops/s. This rate should therefore be considered a minimum.

As discussed previously, this process could be improved by deploying fast algorithms here for use with other multi-scale representations.

Minor cycle

The minor cycle complexity is of the order of $O\,(N_S \, N_{pix}{}^2 \, N_t \; + \; N_S \, N_t^2)$

Principal solution

This represents two stages within the minor cycle. The first is computing the principal solution i.e. finding the 'clean components'. Assuming that a single floating point operation is required for each evaluation then the flop rate for this process is given by:

$$R_{ps} \; = \; N_S \, N_{pix}{}^2 N_t \; Ops/s$$

The overhead for performing this process may be greater than 1 Ops/s, therefore the rate given above should be considered a minimum.

Image update

The second stage is updating the model and residual images which is given by:

$$R_{update} = N_S N_t^2 \; Ops/s$$

As for the previous stage, the overhead is assumed to be a single floating point operation. The rate should therefore be considered a minimum.

Total flop rate

In accordance with Nokilic2014 the total flop rate will be given by:

$$R_{flop} = R_{ccf} + 2N_{major}\, x\, (R_{grid} + R_{FFT} + R_{sc}) + N_{minor}\, x\, (R_{ps} + R_{update})$$

where

N_{major} is the number of major cycles performed. The factor of 2 represents the transformation in both directions from the visibility to the image plane in each cycle.

N_{minor} is the **total** number of minor cycles i.e. the number of minor cycles performed over all major cycles.

R_{ccf} is the rate for computing the convolution kernels.

Note: This holds true for imaging a single field-of-view. A further factor will be required if multiple snapshots are to be made within one use of MS-MFS, for example incorporating w-projection or faceting.

Compressive Sensing (CS)

In the simplest form, compressive sensing algorithms solve essentially the same problem as CLEAN-based algorithms, though the method used is somewhat different. The CS algorithms do however operate in an iterative manner similar to CLEAN algorithms with an application of a full measurement operator at each iteration. The general memory and computational requirements for a CS algorithm will be discussed representing the PURIFY (Carillo et al. 2014) and LOFAR CS (Garsden et al. *in prep*) implementations.

Algorithm Overview

The basic CS algorithm consists of the application of both a forward and adjoint operator (i.e. the transform from the visibility to the image plane, a computation of a the proximity operators which are used to update the solution model, followed by a subtraction and residual update stage. This process is akin to the major cycle of the MS-MFS (and other) CLEAN algorithms (the CS algorithms don't currently have an equivalent minor cycle process).

As for the MS-MFS algorithm we are considering N_{vis} visibilities measured for each frequency channel N_c and each image is assumed to contain N_{pix} pixels. Unlike the MS-MFS algorithm there is no scale model representation. The presence of multiple scales within an image is dealt with inherently by the wavelet representation utilised in the CS algorithms (though the choice of wavelet basis can affect the ability of the algorithm to restore the source distribution at the correct scales).

Compressive sensing naively offers the same type of parallelisation as CLEAN, i.e. the measurement operators could be parallelised on many-core systems (e.g. GPUs). However, since the algorithms are inherently iterative (c.f. the major cycles of CLEAN) they do not naturally lend themselves to parallelisation across multi-node systems. However, recent developments in proximal splitting algorithms (Combettes & Pesquet 2011), combined with a further data partitioning (Carrillo et al. 2014; http://arxiv.org/abs/1406.0359) lead to algorithmic structures that are highly parallelisable across multi-node systems.

Proximal splitting algorithms are generally iterative and do not naturally adhere to a structure that can be highly parallelised. However, algorithms such as the parallel proximal algorithm (PPXA) and the simultaneous-direction method of multipliers (SDMM) do offer a parallel implementation structure, where all the proximity operators can be computed in parallel rather than sequentially (Combettes & Pesquet 2011).

These new algorithmic structures lead to high levels of parallelisation across multi-node systems (Carrillo et al. 2014). Furthermore, the structure of these algorithms will not only allow computations to be distributed, but memory and storage requirements also. In the case of radio interferometric data with N_{vis} visibilities, these could be partitioned into R blocks of data with R measurement operators. In practice, the visibility data would be distributed across R computing nodes which would then perform the application of the measurement operator in parallel. It is worth noting that the communication overheads at each iteration of this major cycle would require the exchange of information regarding the image only and not the visibility data. Such an approach will be critical to tackling the big data-sets anticipated from the SKA. These new highly distributed algorithms are being implemented in the PURIFY package but are not yet ready for use.

The available CS algorithms do not currently incorporate full multi-frequency synthesis capability. Both standard continuum imaging i.e. averaging across frequency or spectral-line imaging i.e. imaging N_c channels independently, can be done trivially (the latter is also easily parallelisable).

However, in order to fully exploit the wide-bandwidths and retain the spectral information, future developments are required to the CS algorithms. There is potential for utilising sparsity in the frequency domain to enable this, but it will necessitate major updates to the available algorithms.

Memory Requirements

During each major cycle, the gridding kernels, solution model images and visibility samples are to be stored in memory. The total memory requirement will therefore depend on the following parameters.

N_c is the number of frequency channels. At each iteration of the major cycle the dirty images for each frequency increment are re-created (following solution and subtraction).

N_{pix} is the number of pixels per image. The requirement for the Large FOV imaging capability of the SKA will require large imaging arrays to be stored in memory for each major cycle.

N_{vis} is the number of visibilities per frequency. These will be read into memory for gridding etc. at each major cycle. The solutions when found are subtracted directly from the visibility data.

As with the MS-MFS CLEAN, the total visibility memory requirement will be $2N_{freq} x\, N_{vis}\, x\, N_T$ (A factor of two is included assuming that N_{vis} represents a single observed polarisation e.g. RR, LL, X or Y and that a minimum of stokes I imaging is required). N_T is the number of time integrations made and will

therefore be dependent on the time-sampling and the length of observation. Data will typically be observed at < 1 sec. integrations.

$$Total\ image\ memory\ requirement = N_{freq}\, x\, N_{pix}$$

Computational requirements

Major Cycle equivalent

Within each major cycle a single model image is created. This process incorporates (de-)gridding, computing the convolution kernels, FFT, UV-subtraction and minimisation.

The basic CS algorithm complexity of this cycle is given by

$$O\,(N_c\ x\ (N_{vis}\, N_k\ +\ N_{pix}\, log_2\,(N_{pix}))\ +\ N_c\, N_{pix})$$

The gridding and FFT processes will both contribute a significant portion of the computing overheads for any given major cycle performed within a CS implementation and will be discussed individually.

Where appropriate FLOP estimates discussed here have been taken from the 2014 SDP Performance Models (B. Nikolic; hereafter Nikolic2014).

Gridding

The total Gridding flop rate will be highly dependent on the choice of gridding kernel e.g. A-/w-projection. In line with Nikolic2014 the gridding flop rate can be estimated as

$$R_{grid} = N_{vis} N_c\, (N^2_{kernel})\ x\ 8\ Ops/s$$

Where N_{kernel} is the total linear kernel size. The factor of 8 Ops/s is taken from Nikolic2014.

FFT

The FFT flop rate is adapted from Nikolic2014 and can be estimated as:

$$R_{FFT} = N_c \; N_{pix} \, log_2(N_{pix}) \; x \; 5 \; Ops/s \; .$$

The factor of 5 Ops/s is as determined by Nikolic2014.

Total flop rate

In accordance with Nokilic2014 the total flop rate will be given by:

$$R_{flop} \; = \; R_{ccf} \; + \; 2N_{major} \, x \, (R_{grid} + \; R_{FFT} \; + \; R_{update})$$

where

N_{major} is the number of major cycles performed. The factor of 2 represents the transformation in both directions from the visibility to the image plane in each cycle.

R_{ccf} is the rate for computing the convolution kernels.

R_{update} represents the computation of the proximity operators at each iteration. This is expected to incur a minimal cost both in terms of computational and memory requirements.

Note: This holds true for imaging a single field-of-view. A further factor will be required if multiple snapshots are to be made within one use, for example incorporating w-projection or faceting.

References

Carrillo, McEwen & Wiaux 2014, MNRAS, 439, 3591 (http://arxiv.org/abs/1406.0359)

Combettes & Pesquet 2011 (https://www.ljll.math.upmc.fr/~plc/prox.pdf)

Garsden et al. 2014, A & A (submitted) (http://arxiv.org/abs/1406.7242v2)

Rau & Cornwell 2011, A & A 532, 71 (http://arxiv.org/abs/1106.2745)