SDP Memo: SDP Non-Imaging Processing Compute Requirements

Document number SKA-TEL-SDP-0000080
Document type .... REP
Revision C
Author .... L. Levin Preston, B. Stappers
Release date 2016-04-07
Document classification .... Unrestricted
Status .... Draft

Name	Designation	Affiliation
B. Stappers	SDP.PIP.NIP task lead	University of Manchester
Signature & Date:	Ben Stappers Ben Stappers (Apr 4, 2016) ben.stappers@manchester.ac.uk	

Name	Designation	Affiliation
R. Bolton	SDP project scientist	University of Cambridge
Signature & Date:	Rosie Bolton Rosie Botlon (Apr 5, 2016) rosie@mrao.cam.ac.uk	

Version	**Date of issue**	**Prepared by**	**Comments**
C	2016-04-07	L. Levin Preston	Submitted for ΔPDR

ORGANISATION DETAILS

Name	Science Data Processor Consortium

Table of Contents

List of abbreviations

CDR	Critical Design Review
CSP	Central Signal Processor
DM	Dispersion Measure
FFT	Fast Fourier Transform
FLOP	Floating Point Operations
FLOPS	Floating Point Operations per Second
ICD	Interface Control Document
LSM	Local Sky Model
MSP	Millisecond Pulsar
NIP	Non-Imaging Processing
OCLD	Optimised Candidate List and Data
PDR	Preliminary Design Review
PTA	Pulsar Timing Array
RAM	Random-access Memory
RFI	Radio Frequency Interference
PSS	Pulsar Search
PST	Pulsar Timing
SDP	Science Data Processor
SKA	Square Kilometre Array
TM	Telescope Manager
TOA	Time of Arrival

Reminder:
Computational Cost = Number of FLOP that need to be executed to turn x into y (a.k.a. Computational Intensity)
Computational Rate = Number of FLOP executed per unit of time – FLOPS
Computational Power = Computational Rate / Efficiency – FLOPS

List of symbols

M_{burst}	Size of single pulse data file
M_{cand}	Size of pulsar candidate data cube
M_{input}	Size of input data per observation
M_{metadata}	Size of metadata
M_{obs}	Average data size per observation
$M_{\mathrm{obs;MAX}}$	Maximum data size per observation
M_{output}	Size of output data per observation
M_{pulsar}	Size of pulsar timing data cube
N_{avg}	Number of averages
N_{beams}	Number of beams simultaneously observed by the array
N_{MAXbeams}	Maximum number of beams simultaneously observed by the array
N_{bin}	Number of bins
$N_{\mathrm{bin:smooth}}$	Number of bins included in smoothing window
N_{bit}	Number of bits
N_{burst}	Number of candidate bursts
N_{byte}	Number of bytes
$N_{\mathrm{byte,pix}}$	Number of bytes per pixel
N_{cand}	Number of pulsar candidates per beam
$N_{\mathrm{ch:smooth}}$	Number of frequency channels included in smoothing window
N_{chan}	Number of channels
N_{FLOPS}	Number of floating point operations per second
N_{FLOPsamp}	Number of floating point operations per sample
N_{it}	Number of iterations
N_{ops}	Number of operations
N_{param}	Number of parameters
N_{pol}	Number of polarisation products
N_{samp}	Number of samples
N_{source}	Number of sources
N_{sub}	Number of sub-integrations
N_{sum}	Number of sums
N_{unique}	Number of unique candidates
R	Compute rate (FLOPS)
t_{comp}	Computation time
T_{obs}	Observation time
T_{sub}	Sub-integration time
Δf	Frequency resolution
Δt	Time resolution (sec)

List of Tables

Summary

This document provides details of the processing requirements for the Non-Imaging Processing (NIP) pipelines within SDP. Each pipeline consists of a number of different components, and basic equations for estimating the FLOP count for each component is given. Numbers for the processing requirements are estimated from expected values listed in the text.

This document was created to support the NIP parts of the SDP document *SKA-TEL-SDP-0000040: Parametric models of SDP compute requirements*, which contains a more concise version of this text.

1 Introduction

1.1 Purpose of the document

This document is part of the SDP documentation package for the Preliminary Design Review (PDR). The purpose of this document is to support the Non-Imaging Processing (NIP) parts of the SDP delta-PDR document *SKA-TEL-SDP-0000040: Parametric models of SDP compute requirements.*

1.2 Scope of the document

This document describes the required performance in the NIP pipelines within SDP, and gives the basic equations used to arrive at these numbers. All FLOP rates and RAM requirements are calculated for expected input numbers, and all assumptions are stated in the text. This document is supporting the SDP document *SKA-TEL-SDP-0000040: Parametric models of SDP compute requirements*. Part of the text is identical in both documents, but more detail is given here.

2 Modelling of Non-Imaging Pipelines

The Non-Imaging Components of the SDP that connect to Pulsar Search (PSS) and Pulsar Timing (PST) in CSP are trivially scalable from prototype to SKA-sized versions. For PSS (searching), despite the fact that the SDP post-processing brings together information from all tied-array beams being searched to reject spurious candidates, this information is in the form of candidate lists, not data, and there exist good constraints on our expectations of the numbers of candidates supplied by PSS. Therefore we anticipate no issues in scaling prototypes of the PSS post-processor in SDP. For PST (timing), the SDP post-processing per pulsar per observation is entirely independent of all other observations and so depending on the configuration each parallel observation of separate pulsars can either go to different nodes or even multiple pulsars to the same node. There are no difficulties in projecting how this scales from the algorithm operation on one example dataset to the full SKA timing experiment.

2.1 Pulsar Search

The pulsar search pipeline will receive pulsar candidate data from CSP in the form of a detected, folded data cube in PSRFITS format (Hotan et al., 2004), accompanied by associated metadata in ASCII format. A description of the pulsar search parameters and the corresponding expected values are listed in Table 1. Each data cube will be of size $M_{\rm cand} = N_{\rm chan} \times N_{\rm bin} \times N_{\rm sub} \times N_{\rm byte}$ = 1.049 GB, and each metadata file will be of size $M_{\rm metadata}$ $\sim$10 kB. In total per observation with SKA-Mid, that corresponds to $M_{\rm input} = N_{\rm beams} \times N_{\rm cand} \times (M_{\rm cand} + M_{\rm metadata})$ = 1.59 TB.

Parameter	Symbol	Expected value
Number of beams	N_{beams}	1500 (SKA-Mid); 500 (SKA-Low)
Number of candidates per beam	N_{cand}	1000
Number of frequency channels	N_{chan}	128
Number of pulse profile bins	N_{bin}	128
Number of subintegrations	N_{sub}	64
Typical observation length	T_{obs}	600 seconds

Table 1: Pulsar search parameters

The processing pipeline for pulsar search consists of five major steps, as described below. The most computationally costly aspects of this work are tied up with steps 1, 2 and 3. All expected numbers of FLOP and RAM requirements are summarised in Table 2.

1. **Merge OCLDs (Optimal Candidate List and Data) from multiple beams.** If we consider the maximal case of approximately 1000 candidates from each of 1500 beams then we need to agglomerate the data from all the beams and then perform a selection, or sifting operation on them. (The number of 1000 candidates per beam is chosen as an upper limit based on the analysis in Lyon et al. (2016), which shows that real pulsars are often found at lower signal-to-noise values than much of the RFI signal, and hence a large number of candidates needs to be saved in order not to miss any pulsars). The sifting will determine which candidates are detected in too many beams and thus most likely are RFI and should be rejected. Also, for those candidates which are common to a few beams determine in which beam they were detected with the highest significance and consider only that candidate for further processing. This step involves coincidence matching of candidates from different beams. This is a sort-like algorithm, and using `quicksort` (Hoare, 1962) as a basis, the estimated number of FLOP is $1.4 \times N_{\mathrm{beams}} \times N_{\mathrm{cand}} \times N_{\mathrm{param}} \times \log_2(N_{\mathrm{beams}} \times N_{\mathrm{cand}} \times N_{\mathrm{param}})$, where N_{param} is the number of parameters to sort by (i.e. spin period, DM, R.A. and Dec.). This step will remove candidates associated with RFI and duplicate detections of real candidates. The remaining pipeline will only need to be run on the unique candidates that survive the coincidence matching, expected to be $\sim 10\%$ of the total. Hence, going forward the total number of candidates are $N_{\mathrm{unique}} = 0.1 \times N_{\mathrm{beams}} \times N_{\mathrm{cand}}$.

2. **Generation and extraction of candidate heuristics.** The generation and calculation of the heuristics will happen after the initial list is trimmed. The calculation of the heuristics that will be used for the machine learning tools may in fact be the most computationally demanding part of the pulsar search for SDP. The exact heuristics and associated calculations that will be used are still a matter of study, but we can make some useful estimates from first principles and also from recent experience (Lyon et al., 2016). The first four moments, mean, standard devia-

tion, skewness and kurtosis of the pulse profile and the dispersion measure (DM) against signal-to-noise ratio curves are used as a very effective heuristic in our current machine learning implementation. These first four moments only require low FLOP counts, and need to be computed for both the integrated pulse profile and the DM curve for each $N_{\rm unique}$:

- Mean calculation: $2(N_{\rm bin} - 1) + 4$ FLOP
- Standard deviation calculation: $3(N_{\rm bin} - 1) + 9$ FLOP
- Skew calculation: $3(N_{\rm bin} - 1) + 7$ FLOP
- Kurtosis calculation: $3(N_{\rm bin} - 1) + 7$ FLOP

where $N_{\rm bin}$ is the number of bins making up the integrated profile or DM curve, and assuming bin values to be floats. These are deduced from the standard formulas assuming the following costs: addition, subtraction and multiplication operations all require 1 FLOP and division and square root require 4 FLOP. The standard deviation calculation above assumes the mean has already been calculated in step 1. Likewise the skew and kurtosis calculations assume the mean and standard deviation are already computed and are simply re-used. Assuming a typical candidate has 128 bins across the profile and a similar number in the DM curve, this only adds to 2848 FLOP per $N_{\rm unique}$ candidate. However, these do not represent the total range of heuristics that will be needed. It may be necessary to add heuristics that require a large number of correlations and perhaps even curve fitting. In those cases the number of FLOP required is very much implementation dependent. We have used an experimental approach, using the 22 heuristics described in Bates et al. (2012), to estimate those requirements. To test the FLOP count, a set of 1000 pulsar candidates were run and timed on a single CPU. From this test, we estimate that the extra heuristics require 2 GFLOP per $N_{\rm unique}$ candidate. This will make up the majority of the processing requirement for pulsar search. All heuristic values calculated in this step will be appended to the metadata file for each candidate.

3. **Candidate classification and machine learning.** Determining the computational cost of the machine learning algorithm is very difficult as it is strongly dependent on the algorithm that is being implemented. The best analysis that is currently available is to scale from the analysis methods that we have been implementing. To do this we consider a test machine that has a theoretical capability of 70.4 GFLOPS (according to Intel, using a Quad Core 2.2 GHz i7-2720 QM). For the algorithms used so far, this is the upper bound, since the code is not optimised. In fact, it likely these tests only use 30-50% of that since the operating system is running underneath in addition to other services. Further, the slowest algorithm tested could handle on average 20,000 candidates per second. Assuming that each individual candidate takes an equal share of the available theoretical number of FLOP, we require 3.5 MFLOP per $N_{\rm unique}$ candidate for classification and

machine learning. (This is assuming that the machine learning algorithm The Gaussian Hellinger Very Fast Decision Tree (Lyon et al., 2014) is being used).

4. **Candidate selection.** This step simply involves checking a score given by the candidate classification to a detection threshold, and hence will only require 1 FLOP per N_{unique} candidate.

5. **Alert generation** The last step creates an alert to be sent to TM for each candidate that are selected in the previous step. This requires very low FLOP counts as well, estimated to TBD FLOP per selected candidate.

Task	Total FLOP [GFLOP]	RAM required [GB]
Merge OCLDs from all beams	0.19	30
Generation and extraction of heuristics	300,000	317.6
Classification and machine learning	525	3.6
Candidate selection	0.00015	3.6
Alert generation	TBD (but small)	TBD
Total GFLOP	300,525.62	–
Total GFLOP per second	500.88	–

Table 2: Pulsar search processing summary. The RAM requirements assume that the data not needed in RAM in a specific step, is stored on the local disk or some other media.

The output from the pulsar search pipeline will consist of metadata files for all input candidates, and data cubes for all unique candidates. Hence, the total output data size per observation will be $M_{\mathrm{output}} = N_{\mathrm{unique}} \times M_{\mathrm{cand}} + N_{\mathrm{beams}} \times N_{\mathrm{cand}} \times M_{\mathrm{metadata}}$ = 172.3 GB.

2.2 Single Pulse / Fast Transients

The SDP will carry out the last part of the single pulse/fast transient pipeline, the majority of which is carried out in CSP. The pipeline will receive single pulse candidates from CSP in the form of a dedispersed filterbank data file in PSRFITS format (Hotan et al., 2004), accompanied by associated metadata in ASCII format. The metadata will contain information about:

- The beam number of the detection
- The timestamp corresponding to the detected signal, dedispersed at a particular dispersion measure with respect to a reference frequency channel
- The dispersion measure of the detection
- A set of parameters describing the detection filter (for example, a smoothing coefficient) and the detection significance (for example, signal to noise ratio).

A description of the fast transient parameters and the corresponding expected values are listed in Table 3. Each data file will be of size $M_{\rm burst} = N_{\rm samp} \times N_{\rm chan} \times N_{\rm pol}$ = 2.62 MB, and each metadata file will be of size $M_{\rm metadata}$ $\sim$10 kB. Assuming $N_{\rm burst}$ are detected in each beam and sub-integration, that corresponds to $M_{\rm input} = N_{\rm burst} \times N_{\rm beams} \times (M_{\rm burst} + M_{\rm metadata})$ = 3.93 GB of input data every $T_{\rm sub}$ seconds.

Parameter	Symbol	Expected value
Number of beams	$N_{\rm beams}$	1500 (SKA-Mid) 500 (SKA-Low)
Number of candidate bursts per beam and sub-integration	$N_{\rm burst}$	1
Number of samples	$N_{\rm samp}$	640
Number of frequency channels	$N_{\rm chan}$	1024
Number of polarisations	$N_{\rm pol}$	4
Sub-integration length	$T_{\rm sub}$	10 seconds
Typical observation length	$T_{\rm obs}$	600 seconds

Table 3: Fast Transient search parameters

The processing pipeline for fast transient search consists of five major steps, identical to the steps in the pulsar search pipeline. All expected numbers of FLOP and RAM requirements are summarised in Table 4, assuming that $N_{\rm unique} = 20$ bursts survive coincidence matching every $T_{\rm sub}$ seconds.

Task	Total FLOP per $T_{\rm sub}$ [GFLOP]	RAM required [GB]
Merge OCLDs from all beams	1.1×10^{-4}	0.03
Generation and extraction of heuristics	40	0.11
Classification and machine learning	0.07	4.8×10^{-4}
Candidate selection	2×10^{-8}	4.82×10^{-4}
Alert generation	TBD (but small)	TBD
Total GFLOP	40.07	–
Total GFLOP per second	4.0	–

Table 4: Fast Transient processing summary. The RAM requirements assume that the data not needed in RAM in a specific step, is stored on the local disk or some other media.

The output from the fast transient pipeline will consist of metadata files for all input single pulse candidates, and data files for all unique single pulse candidates. Hence, the total output data size per observation will be $M_{\rm output} = (N_{\rm unique} \times M_{\rm burst} + N_{\rm beams} \times N_{\rm burst} \times M_{\rm metadata}) \times T_{\rm obs}/T_{\rm sub}$ = 4.05 GB.

2.3 Pulsar Timing

The pulsar timing pipeline will receive coherently dedispersed, detected pulsar data cubes from CSP in PSRFITS format (Hotan et al., 2004), which include all associated metadata. In the requirements it is specified that up to 16 pulsars will be able to be observed simultaneously in high precision-timing mode. This means that the data from SKA1_Mid array will be formed into 16 tied-array beams which will point anywhere within the primary beam, although they could also correspond to beams generated from different sub-arrays. These beams will have a bandwidth which is typically equal to the total available bandwidth in each of the SKA1_Mid bands, i.e. up to about 1 GHz. This wide bandwidth will be fed into a timing-specific beam-former. It will channelise the data, down to channels that are a few MHz wide and sum all available/required dishes together coherently. It will then output a complex, i.e. amplitude and phase, data set to the pulsar timing backend. The bulk of the pulsar timing processing will be undertaken on hardware associated with the CSP which will be responsible for performing coherent dedispersion (full correction for dispersion in the interstellar medium) on the data and fold it at the known pulsar period. It will also form, and partially calibrate, the polarisation of the data.

A description of the pulsar timing parameters and the corresponding expected values are listed in Table 5. Here the expected values are given for three different observing modes: ordinary pulsars, millisecond pulsars (MSPs) and Pulsar Timing Array (PTA) pulsars. As mentioned above, a maximum of 16 tied-array beams will be available for pulsar timing. Since the beams are independent of each other, a combination of the different observing modes can be used simultaneously. This also implies that each pulsar can be processed completely independent of the others, and hence all 16 pulsars do not need to be processed on the same compute node or even the same compute island. Each input data cube will be of size $M_{\rm pulsar} = N_{\rm bin} \times N_{\rm chan} \times N_{\rm sub} \times N_{\rm pol} \times N_{\rm bytes}$.

The main tasks of the SDP timing analysis will be to perform final RFI mitigation steps, final polarisation calibration, generation of a time of arrival for the observation in real time through cross-correlation with a template and other diagnostics of the data quality. The processing pipeline for pulsar timing consists of five major steps, as described below. All expected maximum numbers of FLOP and RAM requirements are summarised in Table 6.

- **Radio Frequency Interference mitigation** The first step involves removing any parts of the received data cube that have been affected by RFI. The exact procedure to mitigate the RFI is still under investigation, but estimates can be made assuming a commonly used algorithm for removing affected frequency channels (”channel zapping”) and time samples (”lawn mowing”) using a running median. Channel zapping is carried out on the bandpass of each sub-integration and compared to a median smoothed version of the bandpass. In this case, the FLOP count is a combination of that required to create the bandpass and that required for median smoothing, hence $(2 \times N_{\rm bin} \times N_{\rm pol} + N_{\rm chan} \times N_{\rm ch:smooth}^{1/2}) \times N_{\rm sub}$, where $N_{\rm ch:smooth}$ is the number of frequency channels included in the smoothing window

Parameter	Symbol	Ordinary pulsars MSPs	Expected value	PTAs
Maximum number of beams	N_{MAXbeams}	16	16	16
Average number of beams	N_{beams}	8	3	3
Number of frequency channels	N_{chan}	512	512	4096
Number of pulse profile bins	N_{bin}	2048	4096	4096
Number of subintegrations	N_{sub}	180	180	180
Number of polarisations	N_{pol}	4	4	4
Number of bytes per pixel	N_{bytes}	4	4	4
Typical observation length [seconds]	T_{obs}	1800	1800	1800
Data size per pulsar [GB]	M_{pulsar}	3.02	6.04	48.32
Average data size per observation [GB]	M_{obs}	24.2	18.1	145.0
Maximum data size per observation [GB]	$M_{\mathrm{obs;MAX}}$	48.3	96.6	773.1

Table 5: Pulsar timing parameters

(Mohanty, 2003). In a similar way, the profile lawn mowing is used to replace RFI-affected phase bins with the local median plus some random noise. Here, each sub-integration, frequency channel and polarisation of the pulse profile is compared to a median smoothed version of the same profile. The number of FLOP required for this analysis is $N_{\mathrm{chan}} \times N_{\mathrm{sub}} \times N_{\mathrm{pol}} \times N_{\mathrm{bin}} \times N^{1/2}_{\mathrm{bin:smooth}}$, where $N_{\mathrm{bin:smooth}}$ is the number of phase bins included in the smoothing window.

- **Calibration** After the data cubes have been cleaned from RFI, they are passed on to the calibration step. Provided that the calibration solutions are given as input from TM, estimations of FLOP required for both flux calibration and polarisation calibration scale linearly with the total number of phase bins. That is for flux calibration the estimated number of FLOP is $N_{\mathrm{bin}} \times N_{\mathrm{chan}} \times N_{\mathrm{sub}} \times N_{\mathrm{pol}}$, and for polarisation calibration $N_{\mathrm{bin}} \times N_{\mathrm{chan}} \times N_{\mathrm{sub}} \times 20$, (where 20 = 16 multiplications + 4 adds).

- **Averaging and archive product generation** Thus far the pipeline has been using full-resolution data cubes, since RFI mitigation and calibration greatly benefit from high resolution data. The following steps require higher signal-to-noise values rather than high resolution, and therefore this step produces a partly averaged data cube to be used through the remaining steps. This branch also produces three additional versions of averaged data cubes, which will be sent to the Long-term Preservation and saved for future post-processing. The output consists of the following averaged data cubes in PSRFITS format:
  - Summed over the entire frequency band
  - Summed over all sub-integrations

  - Summed over the entire frequency band and all sub-integrations
  - Summed over part of the frequency band and part of the sub-integrations

  The estimated number of FLOP required for creating each of these data cubes is $N_{\rm sum} \times N_{\rm bin} \times N_{\rm chan} \times N_{\rm sub} \times N_{\rm pol}$, where $N_{\rm sum}$ is the number of different sums that need to be carried out.

- **Time of arrival determination** The partly averaged data cube created in the last step, will be used for time-of-arrival (TOA) determination. It will determine the ToAs by cross correlating the current observation to a pulsar-specific template provided from TM. The FLOP count in this step can be estimated as $N_{\rm bin}^2 \times N_{\rm chan} \times N_{\rm sub}$.

- **Residual determination and model update** This branch will use the currently best timing model to compute expected arrival times based on the model. It will then compare the expected arrival times to the observed arrival times passed on from the last step in the pipeline and generate timing residuals as the difference between model and observation, and update the model of the pulsar. The FLOP count for this step is TBD, but will be small in the context of the full pipeline.

As discussed above, each of the up to 16 beams will be pointing at a different pulsar and there is no need for any interaction between the processing of each of the different beams. Therefore they can run in parallel, but also asynchronously if needed. The requirements above are specified for a single beam/pulsar and so if the architecture supports larger amounts of memory and processing capacity then more than one beam could be processed on a single node.

Task	Total FLOP per pulsar [GFLOP]	RAM required [GB]
Radio Frequency Interference mitigation	109.3	48.3
Calibration	72.5	96.6
Averaging and archive product generation	120.8	48.7
Time of arrival determination	193.3	0.38
Residual determination and model update	TBD (but small)	0.015
Total GFLOP	495.9	–
Total GFLOP per second	0.28	–

Table 6: Pulsar timing processing summary. Maximum numbers required per pulsar observed, assuming a PTA pulsar. 16 pulsars can be observed simultaneously, but processed independently. The RAM requirements assume that the data not needed in RAM in a specific step, is stored on the local disk or some other media.

References

Bates, S. D., Bailes, M., Barsdell, B. R., Bhat, N. D. R., Burgay, M., Burke-Spolaor, S., Champion, D. J., Coster, P., D'Amico, N., Jameson, A., Johnston, S., Keith, M. J., Kramer, M., Levin, L., Lyne, A., Milia, S., Ng, C., Nietner, C., Possenti, A., Stappers, B., Thornton, D., and van Straten, W.: The High Time Resolution Universe Pulsar Survey - VI. An artificial neural network and timing of 75 pulsars, MNRAS, 427, 1052–1065, doi:10.1111/j.1365-2966.2012.22042.x, 2012.

Hoare, C. A. R.: Quicksort, The Computer Journal, doi:10.1093/comjnl/5.1.10, 1962.

Hotan, A. W., van Straten, W., and Manchester, R. N.: PSRCHIVE and PSRFITS: An Open Approach to Radio Pulsar Data Storage and Analysis, PASA, 21, 302–309, doi:10.1071/AS04022, 2004.

Lyon, R. J., Knowles, J. D., Brooke, J. M., and Stappers, B. W.: Hellinger Distance Trees for Imbalanced Streams, in: 22nd IEEE International Conference on Pattern Recognition, ICPR '14, pp. 1969–1974, IEEE, 2014.

Lyon, R. J., Stappers, B. W., Cooper, S., Brooke, J. M., and Knowles, J. D.: Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach, MNRAS, submitted, 2016.

Mohanty, S. D.: Efficient Algorithm for computing a Running Median, Tech. rep., Max Planck Institut für Gravitationsphysik, https://dcc.ligo.org/public/0027/T030168/000/T030168-00.pdf, 2003.