SQUARE KILOMETRE ARRAY

SDP

SCIENCE DATA PROCESSOR

SKA1 SDP Prototyping Reports Overview

Document Number.. SKA-TEL-SDP-0000054
Document Type.. REP
Revision.. 04
Authors..J. Coles, P. Alexander, V. Allan, A. Brown, T. Cornwell, W.S. Loo, B. Mort, J. Taylor, R. Simmonds, A. Wicenec
Date.. 2019-03-15
Document Classification.. Unrestricted
Status...Released

Name	Designation	Affiliation	Signature	
Authored by:				
Jeremy Coles	SDP Project Manager	University of Cambridge	Jeremy Coles Jeremy Coles (Mar 15, 2019)	
			Date:	
Owned by:				
Jeremy Coles	SDP Project Manager	University of Cambridge	Jeremy Coles Jeremy Coles (Mar 15, 2019)	
			Date:	
Approved by:				
Paul Alexander	SDP Project Lead	University of Cambridge	Paul Alexander Paul Alexander (Mar 15, 2019)	
			Date:	
Released by:				
Paul Alexander	SDP Project Lead	University of Cambridge	Paul Alexander Paul Alexander (Mar 15, 2019)	
			Date:	

DOCUMENT HISTORY

Revision	Date Of Issue	Engineering Change Number	Comments
02C	2018-03-25	ECP-SDP-180001	Prepared for M20 pre-CDR review
03	2018-10-31	ECP-SDP-180002	Prepared for M21 CDR review Pre-CDR OARs considered addressed in this document or in the additional underlying prototyping reports Major: SDPPRECDR-237 Document does not give a clear and comprehensible account of current status Minor: SDPPRECDR-130 Execution framework selection Typo: SDPPRECDR-341 Cleanup document template artifacts SDPPRECDR-342 Typos SDPPRECDR-343 List of Abbreviations incomplete
04	2019-03-15	ECP-SDP-190001	Prepared for M22 SDP Closeout CDR OARs addressed in this document Minor: SDPCDR-232 Guide to Memo series

DOCUMENT SOFTWARE

	Package	Version	Filename
Word processor	Google Docs		SKA-TEL-SKO-0000000-01_GenDocTemplate
Block diagrams			
Google docs Add-ons	Cross Reference Table of contents List of figures		Used for figure & table numbering and references. Used for heading numbering. Used to generate list of figures and tables

ORGANISATION DETAILS

Name	SDP Consortium
Lead Organisation	The Chancellor, Masters and Scholars of the University of Cambridge The Old Schools Trinity Lane Cambridge CB1 ITN United Kingdom
Website	www.ska-sdp.org

TABLE OF CONTENTS

LIST OF FIGURES

NONE

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ARL	Algorithm Reference Library
BLAS	Basic Linear Algebra Subprograms
CDR	Critical Design Review
CPU	Central Processing Unit
DALiuGE	Data Activated Liu Graph Engine
ECP	Engineering Change Process
FPGA	Field-programmable gate array
GPU	Graphical Processing Unit
MIC	Many Integrated Core
P3-AlaSKA	Performance Prototype Platform for SKA
SDP	Science Data Processor
SE	System Engineering
SEI	Software Engineering Institute
SIP	SDP Integration Prototype
SKA	Square Kilometre Array
SKAO	SKA Project Office
TANGO	TaCO Next Generation Objects

1. Introduction

1.1. Purpose of the document

This document provides a high-level overview of the Science Data Processor (SDP) prototyping activities carried out over the design phase. It acts as an overview for the separately prepared prototyping reports by setting out the approach, context and drivers.

1.2. Scope of the document

The scope of the document is limited to introducing the largest SDP prototyping activities at the time of CDR submission. Reports on each of the main activities have been prepared separately and many of these are supported by memos that discuss results in more detail. Additional smaller scale informative activities can be found in the SDP memo series and are not covered, but a list of memos for reference is included.

2. References

2.1. Applicable documents

The following documents are applicable to the extent stated herein. In the event of conflict between the contents of the applicable documents and this document, **the applicable documents** shall take precedence.

[AD1] SDP Risk Register SKA-TEL-SDP-0000052 Rev 09
[AD2] ECP-170001 Proposal to adopt SEI Standards for development of software CDR deliverables
[AD3] SKA-TEL-SDP-0000073 SDP Stage 2 Milestones, Rev 03
[AD4] SKA-TEL-SDP-0000033 SDP Requirements and Compliance Statement, Rev 04
[AD5] SKA-TEL-SDP-0000117 SKA1 SDP Execution Frameworks Prototyping Report, Rev 02
[AD6] SKA-TEL-SDP-0000137 SKA1 SDP Integration Prototype (SIP) report, Rev 02
[AD7] SKA-TEL-SDP-0000150 SKA1 SDP Algorithm Reference Library (ARL) Report, Rev 02
[AD8] SKA-TEL-SDP-0000151 SKA1 SDP Performance Prototype Platform (P3-ALaSKA) Prototyping Report, Rev 02
[AD9] SKA-TEL-SDP-0000152 SKA1 SDP Delivery Prototyping Report, Rev 02
[AD10] SKA-TEL-SDP-0000153 SKA1 SDP DALiuGE Prototyping Report, Rev 02
[AD11] SKA-TEL-SDP-0000154 SKA1 SDP Vertical Prototyping and Compute Efficiency Report, Rev 02

2.2. Reference documents

The following documents are referenced in this document. In the event of conflict between the contents of the referenced documents and this document, **this document** shall take precedence.

NONE

3. Motivation and scope of Prototyping

Software prototyping is an important and significant component of the work undertaken by the SDP consortium. Such prototyping is motivated by different considerations compared to hardware prototyping in other consortia.

The process and required deliverables of the SDP follow the SEI process as detailed in ECP-170001 [AD2]. The delivery of the SDP design is an SEI-compliant architecture together with the necessary SE artefacts to place the SDP in the wider SKA system; see SKA-TEL-SDP-0000073 [AD3].

It is helpful to be clear what the prototyping work has *not* been intended to address or achieve. Firstly it has not attempted to prototype a whole system; this is out of scope for the design phase. Nor is the prototyping intended to qualify the design - this would effectively require prototyping the complete system - which will happen during construction following Scaled Agile development practices. Finally we have not been prototyping to verify requirements; that would require us to assume a specific implementation and is again out of scope.

The prototyping we have undertaken has been to verify and support the evolution of targeted aspects of the SDP architecture. Only a subset of the L1/L2 requirements are architecturally significant and the prototyping we have undertaken partially verifies the architecture driven by this subset of requirements [AD4]. Throughout we have used risk assessment as a key determinant to direct the limited prototyping effort available [AD1].

Although not a driver, requirement or motivation for the prototyping work, a key outcome has been the formation of a code base that can act as a catalyst for the development of a Minimal Viable Product during the construction phase. Whether the code is used in such a way is a decision for the SKA project office to make when considering construction phase activities.

Management of the prototyping activities has also to a limited extent prototyped aspects of the SAFe scaled agile approach to software development, which is being adopted by the SKAO in bridging and construction work. SDP prototyping activities have been undertaken by three main (semi-)agile teams working in small iterative cycles within the larger SDP (6-8 week) work sprints akin to Program Increments in SAFe. The teams have run their own JIRA Agile Boards and developed backlog items for consideration at the SDP sprint planning events. This approach has proved very successful. The teams and the scope of their activities (to be explained shortly) has been:

- ***SIP:***
  - *SDP Integration Prototype*
  - *P3, Alaska and Shared Services*
- ***Execution frameworks:***
  - *Execution Frameworks*
  - *Algorithm Reference Library*
- ***DALiuGE:***
  - *DALiuGE* development & scale tests

Other prototyping work has been undertaken outside of any team based iteration approach and this includes: Scale-out, *Vertical* and *Delivery* prototyping. Each of the areas denoted in italics is explained in more detail in the accompanying series of prototyping reports. In addition to these activities various other investigations have been undertaken in the course of SDP work to explore specific questions, such as the level of arithmetic precision needed for the SDP and how justifiable it is to pursue a data driven architecture at all. At the time of submission the SDP memo series contains over 90 supporting documents. Whilst they do not form part of the CDR submission, they are listed in Section 5 for completeness. Additional write-ups based on SDP Confluence notes are to be expected before SDP close-out, to provide SKAO with all the background information and experience we have acquired during pre-construction.

4. Key prototyping activities

4.1 Reports structure

At the time of the SDP pre-CDR submission we gave only a high-level overview of the ongoing prototyping work. For the CDR submission we have elaborated substantially by way of individual reports which themselves reference more detailed work captured in SDP memos. The reports in the submission are:

- SKA1 SDP Execution Frameworks Prototyping Report [AD5]
- SKA1 SDP Integration Prototype (SIP) Report [AD6]
- SKA1 SDP Algorithmic Reference Library (ARL) Report [AD7]
- SKA1 SDP Performance Prototype Platform (P3-ALaSKA) Prototyping Report [AD8]
- SKA1 SDP Delivery Prototyping Report [AD9]
- SKA1 SDP DALiuGE Prototyping Report [AD10]
- SKA1 SDP Vertical Prototyping and Compute Efficiency Report [AD11]

For consistency the report authors (with the exception of the authors of the Vertical Prototyping and Efficiency Report) were asked to cover:

1. The motivation and scope of the work undertaken with particular emphasis on how it has impacted the SDP architecture and what SDP risks have been addressed;
2. The most important results/conclusions derived from the prototyping including major achievements and any difficulties encountered;
3. Lessons learned in the process of carrying out the prototyping;
4. Suggestions and recommendations for further work.

In addition several of the reports explain how the work used, or was used by, another area of prototyping activity to leverage additional findings.

The motivations for the main prototyping activities are outlined in the following sections.

4.2 Execution Frameworks

Execution Frameworks have different qualities and properties and no one existing framework yet meets all SDP requirements. Execution frameworks are a key element of the SDP architecture - they provide the framework in which pipelines and workflows are described, they organise data distribution and access and provide a route to scalability. Most of the execution frameworks that are being considered are data-driven or data-flow frameworks which translate either internally or explicitly the workflows into a-cyclic graphs and distribute and execute those graphs managing data flow/placement.

The prototyping in this area has explored the performance of existing frameworks and their respective limitations, as well as examined how best to interface to a variety of frameworks such as DASK and MPI. A key objective was to verify that we can use task-based parallelism to quickly process SDP data.

4.3 SDP Integration Prototype

The aim of this work has been to develop a lightweight prototype of the major components of the SDP Operational System, with a particular focus on:

- verification, testing and analysis of the high-level SDP architecture;
- verification and testing of external (to other SKA components) and internal (to other SDP components) software interfaces;
- providing limited test of horizontal scaling on representative prototype hardware (AlaSKA Performance Prototype Platform (P3)).

The SIP prototype has endeavoured to reflect, inform and verify the high-level architecture, especially the main component decomposition, module decomposition, interfaces and overall behaviour. It is was not intended that this prototype be functionally complete, but significant functionality of control and monitoring behaviour is provided.

4.4 Algorithmic Reference Library (ARL)

The Algorithm Reference Library (ARL) is designed to present algorithms for radio astronomy imaging in a simple Python+Numpy-based form. This is so that the implemented functions can be seen and understood without resorting to interpreting source code shaped by real-world concerns such as optimisations. The major work in producing the ARL was in developing the scientific processing capabilities that are needed to meet the functional requirements on SDP.

There has been a strong overlap of this prototyping activity with the execution frameworks prototyping and also with SIP. The SDP architecture requires that these are reusable functional components which may be accessed from different execution frameworks.

4.5 Performance Prototype Platform (P3-ALaSKA)

This prototyping aimed to better understand and evolve the architecture of the SDP Operational System in relation to its hardware needs, and how that hardware can be abstracted via a set of services, called Platform Services.

The Performance Prototype Platform (P3-AlaSKA or P3) has provided a prototyping framework for use by the SDP consortium. The compute, storage and networking is realised by a bare metal OpenStack, multi-tenanted solution to produce a flexible software-defined infrastructure that can easily scale to accommodate particular prototyping activities. A number of execution environments are supported.

P3 is also being used to prototype OpenStack itself in the guise of a potential candidate for a number of Platform Services components by investigating core OpenStack services for infrastructure (eg. Ironic, Ansible) as well as Monitoring and Logging (OpenStack Monasca), container orchestration (OpenStack Magnum) and higher level shared services (e.g. File-system-as-a-service through OpenStack Manila and object storage for Buffer provisioning) and packaged execution environments such OpenStack Sahara for Spark-as-a-service.

While the majority of activity has been in support of SIP, P3 has been used to investigate aspects of the Buffer. In particular to support Networked and Parallel File Systems as well as Object Storage as part of the Storage Back-end Component and File System Interface. Support has also been given to the performance prototyping for MSv3.

4.6 Delivery

Delivery prototyping has explored many tools and systems used by communities that need to distribute large data sources. This has included looking at data management systems, data transfer systems and data location services. It has also been examining how well International Virtual Observatory Association services will work for next generation radio telescopes. The scope of the SDP Delivery activity changed over time and this included limited deployment of services at SRCs, consequently less work has been performed with end-to-end data management systems than in the early years of SDP.

Work has been performed demonstrating the capability of existing data transfer tools to provide solutions for the SKA. Unlike many other parts of the SDP, the data rates needed for SKA delivery are similar to those that have been explored by the high energy physics community for several years, so the risk to the SKA would mainly come if there is a large difference in the type of access. Other than the size of individual Data Products there is little difference and SKA could split products into smaller objects for transfer if needed.

Prototyping of location services has been more challenging than expected. The Globus Toolkit location services work, but have had limited maintenance. Location services used within the European Grid Infrastructure appear more robust but are not easy to seperate from the rest of the EGI environment. Therefore, while we have a working implementation from the Globus Toolkit, we fully expect a different location service to be used in the production environment.
Work with IVOA has mainly been with data modeling using data from VLA and early MeerKAT data and deploying this with the IVOA services implementation from the Canadian Astronomy Data Centre.

4.7 DALiuGE

The Data Activated Liu Graph Engine (DALiuGE) is a flexible and scalable graph execution engine that has been developed by the SDP team at ICRAR in Western Australia with a focus on radio astronomy use. The graph representation approach of DALiuGE allows it to deterministically exploit as many parallelism opportunities as possible. The prototyping efforts have explored how such an approach scales and whether any major bottlenecks can be identified. Among other things the work has helped to address questions about efficiency of logical to physical graph translation, component reuse and has enabled important scalability tests to be conducted. An emulation layer has allowed execution of the ARL within DALiuGE.

The DALiuGE prototype demonstrates and verifies a number of critical aspects of the SDP architecture and in particular architectural pattern namely:

- Scalability of graph-based approaches to Peta-scale systems
- Reuse of components within a graph-based processing environment
- Efficient translation of a logical graph defining the algorithm to a physical graph describing the mapping of the algorithm onto the physical hardware layer
- Efficient instantiation of large number of processes.

The aims of DALiuGE prototyping have not required it to address

- Dynamic load balancing or graph deployment
- Maintainability of the workflow description (DALiuGE requires this to be explicitly constructed as an acyclic graph)
- Use of queues for sky model and metadata

4.8 Vertical Prototyping and Compute Efficiency

This report pulls together results from a number of SDP activities that have sought to understand how currently available hardware limits specific algorithm (e.g. gridding) performance. It explains the challenges of arriving at a single efficiency estimate for an entire pipeline and the bounds set by various system parameters and modelling approaches (such as Roofline) as well as the problems met in attempting to project future efficiencies based on results from current generations of hardware.

The report also pulls together all of the vertical prototyping activities within SDP where the performance of the functional processing components themselves have been developed and tested. The functional processing components are called from within the execution framework to perform often computationally demanding functions on the data.

The aims of the vertical prototyping work have been to:

- Investigate and develop numerical implementations of specific functional kernels to assess and maximise performance on current and emerging hardware;
- Verify aspects of the SDP performance model used for system sizing and hence costing;
- Engage with industry to understand fundamental limitations on the processing efficiency.

Both in-house prototyping and significant engagement with industry has been undertaken. Examples of industry engagement include:

- Specific contracts / codesign with NVIDIA, Intel, ARM and IBM;
- Assessment of emerging hardware architectures with vendors and processor suppliers

4.9 Scale-out

The SDP China team have focussed on scale-out prototyping work to examine approaches which provide the ability to scale beyond the immediate requirements of SKA1. Much of the work has a direct impact on SKA1, but the greater horizon enables a more academically focussed approach to be taken. As such there is no accompanying report, but memos related to this are referenced in the next section.

Key activities have included:

- Development of imaging approaches based on spherical harmonic, rather than the standard Fourier, basis representation and applied to Tianlai pathfinder array;
- Optimisation of the ARL kernels to C++/BLAS and investigation of automated the parallel automation using STAR;
- Using Alluxio to enhance performance of SPARK and testing of DALiuGe at scale;
- A computation platform based on CPU+MIC+GPU+FPGA architecture has been developed by Inspur corporation and Shanghai Jiao Tong University to accelerate the processing process;
- Prototyping system scheduling including data movement / positioning as a key scheduling constraint;
- Use of AI in pulsar search analysis.

5. SDP memo series

As mentioned earlier and elsewhere in the CDR document set, SDP contributors have recorded detailed work outputs into an SDP memo series. These memos vary in scope and may express the views of the author or a group who have examined a particular issue. Several of these are directly referenced from the prototyping reports, but many relate to smaller scale analysis and prototyping work that has helped move the SDP design forwards. Table 1 shows the current set of memos. Those with titles in **bold** font are directly referenced in the full set of CDR documents, but it should be noted that all were relevant to the SDP architecture decision making process and rationale. Additional memos may be produced by the SDP Consortium before closeout to capture additional useful material currently only held in the SDP Confluence; these will be missing from Table 1.

SDP Memo Number	Title	Lead Author	SKA Document Number
1	PAF De-rotation options trade-off: initial SDP response	Rosie Bolton	SKA-TEL-SDP-0000156
2	**Time and channel averaging**	Chris Skipper	SKA-TEL-SDP-0000057
3	**Fast Fourier Transforms**	Stef Salvini	SKA-TEL-SDP-0000058
4	**Feasibility analysis of baseline-dependent averaging**	Stefan Wijnholds	SKA-TEL-SDP-0000017
5	Data Challenge supplement.	Markus Dolensky	SKA-TEL-SDP-0000024

6	**Receive and Pre-process Visibility Data**	Ger van Diepen	SKA-TEL-SDP-0000028
7	**The SDP Calibration Component**	Stef Salvini	SKA-TEL-SDP-0000029
8	**The SDP Imaging Pipeline**	Anna Scaife	SKA-TEL-SDP-0000030
9	**Science Data Analysis**	Melanie Johnston-Hollitt	SKA-TEL-SDP-0000031
10	**Estimating the SDP Computational Efficiency**	Bojan Nikolic	SKA-TEL-SDP-0000086
11	SDP Compute Platform Requirements	Bojan Nikolic	SKA-TEL-SDP-0000157
12	**iPython Performance Model Handbook**	Francois Malan	SKA-TEL-SDP-0000041
13	Regional Centres	Rob Simmonds	SKA-TEL-SDP-0000060
14	PROT.ISP: File-system Benchmark Report	Willi Homberg	SKA-TEL-SDP-0000061
15	**Can SDP use existing big- data systems?**	Rob Simmonds	SKA-TEL-SDP-0000072
16	**SDP Non-Imaging Processing Compute Requirements**	Ben Stappers	SKA-TEL-SDP-0000080
17	**Dataflow Choice**	Markus Dolensky	SKA-TEL-SDP-0000066

18	**SDP Data Driven Architecture I: Justification**	Bojan Nikolic	SKA-TEL-SDP-0000088
19	SDP Load List	Ferdl Graser	SKA-TEL-SDP-0000078
20	**Data-driven Architecture Prototyping Report**	Markus Dolensky	SKA-TEL-SDP-0000082
21	**Dataflow prototyping report**	Peter Braam	SKA-TEL-SDP-0000083
22	**MeerKAT Report**	Simon Ratcliffe	SKA-TEL-SDP-0000084
23	**Horizontal Prototyping Interim Report**	David Terrett	SKA-TEL-SDP-0000085
24	Organisation of the SDP 2016 PDR Architecture Documentation	Peter Braam	SKA-TEL-SDP-0000087
25	Updated SDP Cost Basis of Estimate June 2016	Ferdl Graser	SKA-TEL-SDP-0000091
26	The Infeasibility of High Quality Ionospheric Calibration of SKA1-LOW	Tim Cornwell	SKA-TEL-SDP-0000118
27	The Infeasibility of High Quality Ionospheric Calibration of SKA1-LOW: response to comments	Tim Cornwell	SKA-TEL-SDP-0000119
28	Gridding Computational Intensity	Peter Wortmann	SKA-TEL-SDP-0000158
29	Integration Prototype Build and Deployment Infrastructure Design	Iain Emsley	SKA-TEL-SDP-0000120

30	Summary of SDP information relevant to study potential available data-driven systems	Montse Farreras	SKA-TEL-SDP-0000136
31	Quantifying Power Efficiency of FFTs on NVidia GPUs	James Kent and Bojan Nikolic	SKA-TEL-SDP-0000121
32	On the Precision Required in SDP Pipelines	Stef Salvini	SKA-TEL-SDP-0000138
33	Sky Model Considerations	Ian Heywood	SKA-TEL-SDP-0000139
34	**Practical Distributed Data Processing using SWIFT/T & CASA**	Bojan Nikolic	SKA-TEL-SDP-0000140
35	**SDP Memo FP32 vs. FP64**	John Taylor	SKA-TEL-SDP-0000141
36	**Convolution Gridding on CPU, GPU and KNL**	Jacques Du Toit, Anna Brown	SKA-TEL-SDP-0000122
37	IO and Storage Software	Peter Braam	SKA-TEL-SDP-0000123
38	**Pipeline Working Sets and Communication**	Peter Wortmann	SKA-TEL-SDP-0000124
39	**Full-Scale DALiuGE Data Simulation and Reduction on Tianhe-2**	Baoqiang Lao and Chen Wu	SKA-TEL-SDP-0000159
40	**PSRFITS Overview for NIP**	Robert Lyon	SKA-TEL-SDP-0000160
41	**Calibration and imaging context**	Tim Cornwell	SKA-TEL-SDP-0000125
42	**Data Model Summary for Pulsar/Transient Search & Timing**	Robert Lyon	SKA-TEL-SDP-0000161

43	**Pulsar Timing Failure Analysis**	Robert Lyon	SKA-TEL-SDP-0000162
44	Technical Use Case Analysis	Various	SKA-TEL-SDP-0000037
45	**Overview of Buffer Prototyping and Modeling**	John Taylor	SKA-TEL-SDP-0000126
46	**Experiences with the SPEAD protocol**	Rodrigo Tobar	SKA-TEL-SDP-0000127
47	Scheduling Schemes and Buffer Considerations	Yongxin Zhu	SKA-TEL-SDP-0000128
48	Two Dimensional Sparse Fourier Transform Algorithms	Haihang You	SKA-TEL-SDP-0000129
49	SDP limitations on wide-area mapping mode (Report to Mid RT14)	Mark Ashdown	SKA-TEL-SDP-0000142
50	**The Accelerator Support of Execution Framework**	Feng Wang	SKA-TEL-SDP-0000130
51	**Cloud Native Applications on the SDP Architecture**	Piers Harding	SKA-TEL-SDP-0000131
52	**Apache Kafka for an SDP log based architecture**	Piers Harding	SKA-TEL-SDP-0000163
53	**Monitoring and Logging for the SDP**	Piers Harding	SKA-TEL-SDP-0000132
54	**Compute Node Pipeline Efficiency Assessment Framework**	TN Chan	SKA-TEL-SDP-0000133
55	Compute Node Hardware Technology Landscape Appraisal	TN Chan and Chris Broekema	SKA-TEL-SDP-0000134
56	**Hardware Scaling Co-Design Recommendations**	TN Chan and Chris Broekema	SKA-TEL-SDP-0000135

58	**Analysis of w-projection kernel size.**	D. Mitchell *et. al.*	SKA-TEL-SDP-0000204
59	**Modelling and evaluating the IO MID1 ICAL pipeline on SPARK**	Qiuhong Li	SKA-TEL-SDP-0000143
60	**A parametric model for the calibration and imaging costs of SKA1**	T. J. Cornwell	SKA-TEL-SDP-0000205
61	SDP Bandpass Calibration Requirements during EoR Averaging	Daniel Mitchell	SKA-TEL-SDP-0000144
62	**Numerical Precision**	Anthony Griffin	SKA-TEL-SDP-0000145
63	Considerations for the SDP Operating System	Nicolás Erdödy	SKA-TEL-SDP-0000146
64	Security for the SDP Architecture Considerations	Nicolás Erdödy	SKA-TEL-SDP-0000147
65	**Fast Implementation of SKA Algorithm Reference Library**	Haihang You	SKA-TEL-SDP-0000148
66	Partitioning SKA Dataflows for Optimal Graph Execution	Chen Wu	SKA-TEL-SDP-0000164
67	**Deconvolution algorithm functional breakdown**	D. Fenech & J. McEwen	SKA-TEL-SDP-0000206
68	**Performance Prototype Platform (P3-ALaSKA) Monitoring & Logging**	John Taylor	SKA-TEL-SDP-0000165
69	**Performance Prototype Platform (P3-ALaSKA) OpenStack Prototyping**	John Taylor	SKA-TEL-SDP-0000166
70	**P3-ALaSKA Container Orchestration and Compute-Storage Provisioning Interfaces**	John Taylor	SKA-TEL-SDP-0000167

71	LOFAR pipeline DALiuGE	Yan Grange	SKA-TEL-SDP-0000168
72	**Vertical prototyping of the gridding algorithm on GPU**	Anna Brown	SKA-TEL-SDP-0000169
73	**Chebyshev polynomial approximation of kernels in w-projection gridding algorithm on GPU**	Anna Brown	SKA-TEL-SDP-0000170
74	**Optimisation of the w-projection gridding algorithm for FPGA using Intel OpenCL**	Anna Brown	SKA-TEL-SDP-0000171
75	**Data Models for the SDP Pipeline Components**	Ger van Diepen	SKA-TEL-SDP-0000207
76	**Improving the efficiency of direct visibility prediction using NVIDIA Pascal and Volta GPUs**	Karel Adamek	SKA-TEL-SDP-0000172
77	**Using DALiuGE for Distributed SAGECal**	Feng Wang	SKA-TEL-SDP-0000173
78	**Scalability Testing using DALiuGE on Tianhe-2 and Pawsey**	Baoqiang Lao	SKA-TEL-SDP-0000174
79	SDP Prototyping using precursor technology ASKAPSoft with DALiuGE	Andreas Wicenec and Juan Guzman	SKA-TEL-SDP-0000175
80	On Scalable Faraday Tomography/Rotation Measure Synthesis for the Square Kilometre Array	Stef Salvini	SKA-TEL-SDP-0000176
81	**Combining Task-Based Parallelism and Platform Services within a Science Pipeline Prototype**	Jamie Farnes	SKA-TEL-SDP-0000177
82	Summarising Initial Scale-Out Prototyping Efforts	Yongxin Zhu	SKA-TEL-SDP-0000178

83	Distribution of the Rau-Cornwell MSMFS algorithm	Tim Cornwell	SKA-TEL-SDP-0000179
84	Observatory Support Tools Use Cases	Ferdl Graser	SKA-TEL-SDP-0000190
85	**PIP.IMG Gridding Algorithms**	Andrew Ensor	SKA-TEL-SDP-0000182
86	**SKA-SDP Gridding on Graphical Processing Units**	NVIDIA Corporation	SKA-TEL-SDP-0000183
87	**Degridding Optimization for SKA on NVIDIA GPUs**	NVIDIA Corporation	SKA-TEL-SDP-0000184
88	**Characterizing FFT performance on GPUs for SKA-SDP**	NVIDIA Corporation	SKA-TEL-SDP-0000185
89	**SKA-SDP Reprojection on Graphical Processing Units**	NVIDIA Corporation	SKA-TEL-SDP-0000186
90	**Comparison of convolution methods for GPUs**	NVIDIA Corporation	SKA-TEL-SDP-0000187
91	**Antenna gain calibration on Graphical Processing Units**	NVIDIA Corporation	SKA-TEL-SDP-0000188
92	**The FFT calculation via NVIDIA cuFFT library**	Karel Adamek	SKA-TEL-SDP-0000189
93	Test Verification System Based on CPU+MIC Architecture Supercomputing Platform	Xudong Zhao	SKA-TEL-SDP-0000191
94	Assumptions used in the SDP HPSO Scheduling model	Francois Milan	SKA-TEL-SDP-0000193
95	**Convolutional Gridding Routine: GPU port**	IBM	SKA-TEL-SDP-0000192

96	A SDP Scheduler Candidate in StarPU	TN Chan	SKA-TEL-SDP-0000196
97	Direction Dependent Self Calibration in ARL	Tim Cornwell	SKA-TEL-SDP-0000195
98	*SPEAD tests*	Chris Broekema	SKA-TEL-SDP-0000199
99	*Dask tests and scaling*	V Allan & T Cornwell	SKA-TEL-SDP-0000200
100	*How to use the Cost Model spreadsheet*	Ferdl Graser	SKA-TEL-SDP-0000201
101	*Providing a roadmap for EF prototyping*	Verity Allan	SKA-TEL-SDP-0000202
102	*Summarising SDP I/O work*	Peter Wortmann	SKA-TEL-SDP-0000203
103	**NIP Data Models and Flow**	Robert Lyon	SKA-TEL-SDP-0000208
104	**Evaluating Data Flow Execution Environments: Regent and Legion as an example**	Peter Braam	SKA-TEL-SDP-0000209

Table 1: SDP memos available for reference at the time of SDP CDR closeout submission. Memos with working titles only at the time of this document submission have their titles in *italics*.