SDP Memo 025: Updated SDP Cost Basis of Estimate June 2016

Document number…… SKA-TEL-SDP-0000091
Document Type……..... REP
Revision……. C
Author……....Paul Alexander, Rosie Bolton, Ferdl Graser, John Taylor
Release Date……. 2016-06-06
Document Classification……. Unrestricted
Status……..... Draft

Lead Author	**Designation**	**Affiliation**
Paul Alexander	SDP Lead	University of Cambridge
Signature & Date:		

Version	Date of Issue	Prepared by	Comments
C	2016-06-06		3rd June cost submission

ORGANISATION DETAILS

Name	Science Data Processor Consortium

Table of Contents

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.

Reference Number	Reference
[AD01]	SKA-TEL-SDP-0000086 SDP Memo: Estimating the SDP Computational Efficiency

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.

Reference Number	Reference
[RD01]	SKA-TEL-SDP-0000046 SDP Costing Basis of Estimate (Draft)
[RD02]	SKA-TEL-SDP-0000038 System Sizing (Draft)
[RD03]	SKA-TEL-SDP-0000050 SDP ILS SPREADSHEET (Draft)
[RD04]	http://wccftech.com/nvidia-pascal-gpu-gtc-2016/

3. Scope

This document provides a supplement to the recently submitted d-PDR documentation and the Basis of Estimate of Costs [RD01] and System Sizing Document [RD02].

4. Summary

This document accompanies the SDP costing submission of 03 June 2016 and is the first report of work by the SDP Costing Resolution Team (RT).
Our costs have changed substantially compared to previous submissions as a result of consideration of various factors. Our methodologies, especially where they have changed, are described in the following sections.

On the hardware side, we have updated our hardware cost estimates to reflect the latest peak performance per GPU node but much more fundamentally, we have assumed that the hardware systems on each site now have to deliver substantially lower compute rates than our previous estimates: using our parametric model to estimate required flops for each High Priority Science Objective (HPSO) experiment we have calculated **the average compute power needed to deliver the HPSO science**. This means that computationally intensive experiments will take longer than their observing time to complete; they effectively take up unwanted SDP time “left over” by the computationally simple experiments. This greatly reduces the cost of the SDP hardware (by a factor of 3) even after buffer sizes are increased to enable this load balancing. This approach does, however, place additional scheduling constraints on the two instruments. The SDP Costing RT has decided that this is a necessary step since the cost reduction is so significant.

5. Software costing

The previous SDP software costs were a purely bottom-up based on a number of additional assumptions. Critical assumptions which significantly increased costs were as follows:

1. Construction phase would be structured so that each element of the product tree could be considered a separate procurement. A consequence of this assumption was the identification of significant costs for integration and documentation of each of these individual software products. This was applied as a net 45% multiplier on all software FTE estimates.
2. Obtaining the required performance would place significant development costs on the domain-specific software and our FTE estimates were calculated on this basis.

For these costings we have challenged and changed these two assumptions.

Assumption 1: Following ongoing discussions within the SDP Costing resolution team we have relaxed assumption (1) to assume instead a more integrated development in the construction phase and more continuity between preconstruction and construction. In this new model the significant integration costs for every component are not justified as overall management coupled with continuity of the software development and a good software engineering process harmonised across the SKA (as proposed by the head of computing) will mitigate the integration risks. Integration tasks associated with the milestones of the AIV plan remain explicitly costed as are overall documentation and management costs. This change of assumption has had a very significant effect on the overall software costs.

Assumption 2: The SDP data-driven architecture places the requirement to get the majority of the overall performance on the execution engine which is not radio astronomy specific, although no one existing implementation meeting all of our needs yet exists. We now cost the non-domain specific software on the basis that a number of “big-data” frameworks are now sufficiently well established that commercial development of them can be contracted to meet SKA needs. This results in a €21.9m estimate for the non-domain software but with a considerable (nearly 50%) contingency. For the domain-specific elements we are therefore

able to reduce our FTE estimates as we have transferred the system performance onto the non-domain components. We have not in this costing attempted a new bottom up costing of the domain-specific software, but reduced the costs in a top-down estimate based in large part on the reduced complexity, but also on the realistic bound of the amount of domain-specific effort that exists (or may exist) within the community. We believe including this as an additional constraint is justified and realistic. In terms of risk and contingency it means that the main contingency is in the form of schedule contingency with the risk being that some elements (or more likely full efficiency) of the domain-specific software possibly being delayed into the operation phase. The operational costing includes on-going domain-specific software effort.

In addition the software costings have been revised to be fully consistent with the SDP product tree submitted at d-PDR.

The on-going work of the SDP costing resolution team (plus SDP team in general) will revisit the bottom-up costings in the next phase of its work.

6. Hardware Costing - Processor Platform

6.1. Overview

The following table provides a summary of the hardware for the Data Processor platform based on the system sizing analysis described below. The items in red have been updated since publication of [RD01]. The actual size of the system and its phasing is discussed in Section 7.

Processor Platform			
Compute	**Low**	**Mid**	**Units**
Number of nodes	896	786	
GPUs Per Node	2	2	
Peak Performance per GPU est.	31	31	DP TFlops
Memory per Node	320	512	GB
Storage per Node	See Below	See Below	
BDN Connection per Node	25 GbE	25GbE	GbE
LLN Connection per Node	56Gbps/50GbE	56Gbps/50GbE	
DPN Connection per Node	10	10	GbE
Number of compute islands (CI)	16	15	
Nodes per compute island	56	56	
Number of management islands per SDP site	1	1	
Buffer			
Total Ingest rate (Required)	0.389	0.371	TB/s

Buffer Size	67	116	PB
Buffer storage per node	75	147	TB
Buffer storage per island	4.2	7.7	PB
Ingest rate per node (Required)	3.5	3.8	Gb/s
Over-subscription at node	7.2	6.6	
Required Ingest rate per Island	194	198	Gb/s
Networking			
Application Network (LLN)	8:1 Oversubscribed		
Number of 100 GbE Ingress Ports	58	74	GbE
Bulk Data Network (BDN)	Over-subscribed 100GbE:25GbE		
Data Preservation Network (DPN)	Over-subscribed 10GbE:10GbE		
Management Network	1 Gbps	1Gbps	
Infrastructure			
Number of Racks	21	20	

BDN = Bulk Data Network; LLN = Low-Latency Network; DPN = Data Preservation Network

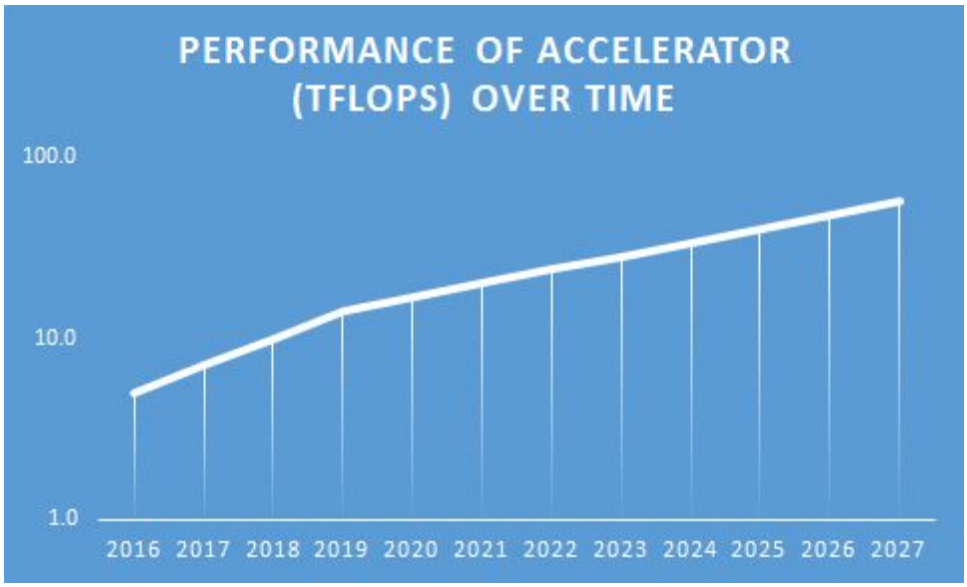
6.2. Compute Node Accelerator - Throughput Optimized Cores

The current hardware costed concept as defined in [RD01] assumes a model based on 2 GP-GPU Accelerators per Compute Node. With the recently announced Pascal P100 GPU [RD04] the performance based on previous generations has leapt to 5TFlops Double Precision performance compared with the 1.4TFlops previously estimated for a Nvidia K20. The memory bandwidth relative to this peak performance has, however, decreased. This factor has not been incorporated in the efficiency calculation which remains at 25% [AD01]

Prices for individual P100 parts are not generally available although there is some indication that 7K€ is an ASP that will be achieved in 2016/17 subject to volume pricing of the P100 and the success of the Intel KNL co-processor as competition. This differs from the 3K€ previously estimated.

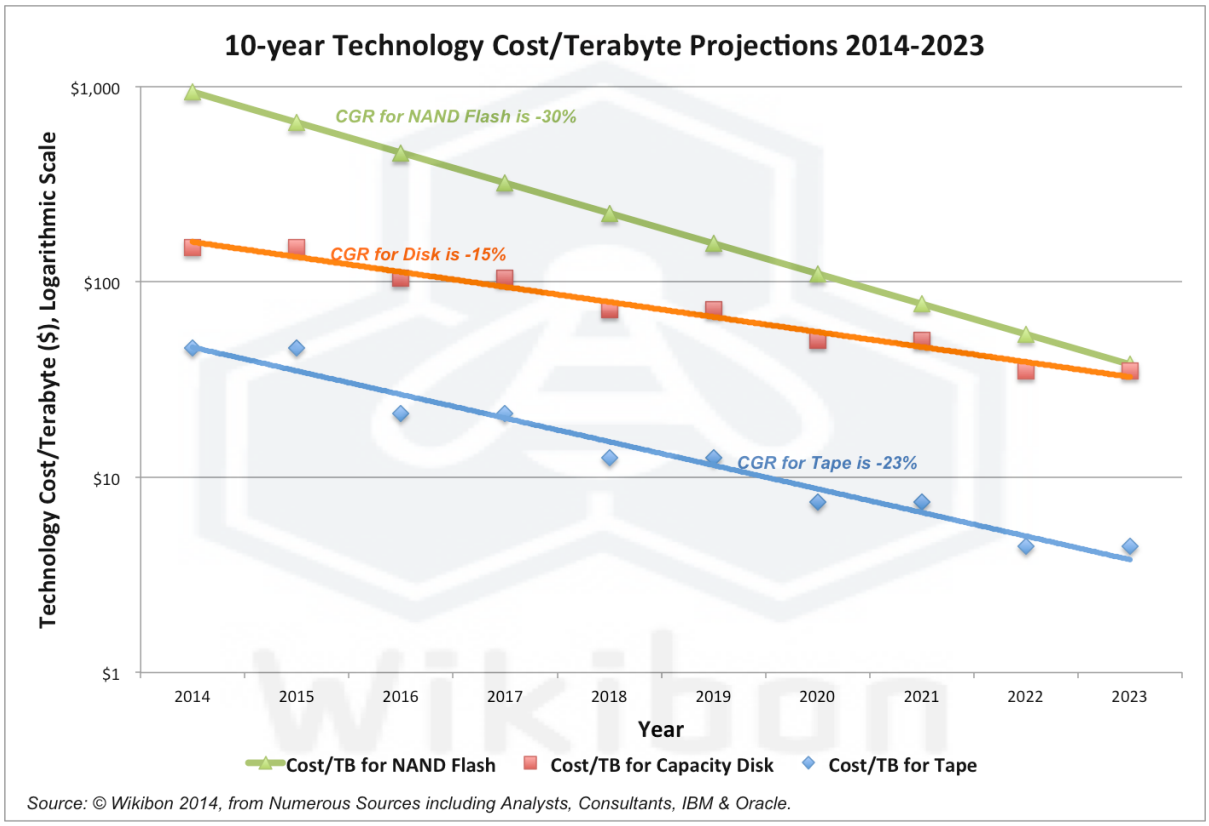
Projected performance has been estimated (at constant price) by a “Moore’s Law type model” assuming the following time periods.

Reference Date used for projections & predictions	**2016-04-01**		
		from	to
Moore's law Period's	**2020-07-01**	**24 months**	**48 months**
Date for prices and performances	2022-07-01		



6.3. Buffer

As was reported in [RD01] the Buffer is anticipated to be implemented by a 100% SSD-type technology. This is assumed to be the most favourable technology based on the projected inflection point between SSD and HDD around the 2023 timescale coupled with increased performance and durability of SSD. This is shown, for example, in the following graph.



Source http://wikibon.org/wiki/v/Evolution_of_All-Flash_Array_Architectures

Based on this the price/Tbyte is predicted to be around 50€. The projected trend should be closely monitored to ensure that the trade-off between SSD and HDD is maintained.

The current model assumes that the Buffer is equally distributed over the compute nodes: thus as the number of nodes decreases the amount of buffer per node increases. Based on the system sizing analysis below this amounts to of the order of requirements as shown here:

	LOW	MID
Total Ingest rate (TB/s)	0.389	0.371
Buffer Size (PetaBytes)	67	116
Buffer storage per node (TB)	75	147

This has implications in terms of the functional aspects of the node and its ability to service the amount of storage together with a multiplicity of accelerators (e.g 2 GP-GPUs) and networking interfaces (viz. Bulk Data, Low-Latency and Preservation Networks). A subsequent analysis will be performed on a distributed buffer based on consolidated storage on a per Compute Island or aggregation of Compute Islands into a “Pod” (unit of storage and compute), especially if the complete buffer is commissioned early as indicated in the analysis below. This will require an additional investigation of the networking implementation as this may limit the span of the network but necessarily have implications on performance. We do not believe, however, that this has any architectural implications.

6.4. CPU - Latency Optimized Cores

With the dominance of Intel processors in the CPU market, the ASP for high-end Xeon processors is increasing. As a result of this we have increased the price for the Xeon host CPU to the Accelerator to around 1100€. There is no indication yet as to whether this increase will slow-down and will depend on the penetration of OpenPower and ARM processors in the server market.

6.5. Platform Spares and Maintenance

6.5.1 Spares

The current model for spares and maintenance to meet the overall availability requirements of the SDP is being worked upon. The overall methodology is based on a spares provisioning acquired as part of the capital expenditure which provides the necessary logistics support to address a mean time-to-repair (MTTR) which includes bringing back into service the failed node of around a week. The method employed is currently based on a failures analysis using estimated mean-time-before-failure (MTBF) data used to derive a failure-per-million-hours (FPMH) parameter and then determine a spares holding based on a confidence level. The model has been adopted for Compute Nodes (assuming k-out-of-n

independent units can fail) and will be extended to include buffer storage and networking. The following Table provides the values for spares based on this.

% Storage Spares	1%	approximated
% Compute Spares	2%	based on MTBF analysis
% Network Spares	10%	approximated

For critical items such as hardware within the LMC and Delivery platforms, a lower MTTR will be assessed. For the current analysis the cost of LMC and Delivery platforms are not discussed here.

6.5.2. Maintenance

The following Table provides typical annual maintenance costs.

Operational Cost Assumptions	1-3 yr	1-5yr	1-7yr
Per Node License (COTS e.g. O/S, Platform Management, Cluster Management)	€1,166	€1,166	€1,166
Compute Hardware Maintenance	3%	10%	17%
Buffer Storage Maintenance	3%	6%	15%
LLN vendor maintenance % of purchase price	18%	25%	42%
BDN/DPN vendor maintenance % of purchase price	10%	15%	28%
Critical Item 4hr Response (e.g. LMC Hardware)	10%	20%	40%

BDN = Bulk Data Network; LLN = Low-Latency Network; DPN = Data Preservation Network

The per-node license cost is based on a license for the the following Product Tree Items.

Product Tree Item	Description	Fraction of per node license
C.1.1.3.1	Compute Operating System software Software Maintenance	1/6th
C.1.1.3.2	Middleware Software Maintenance	2/3rd
C.1.1.3.3	Scheduler Software Maintenance	1/6th

7. Overall system sizing methodology

In this section we provide justification for the system sizing inputs for the overall sustained compute rate and the buffer size for LOW and MID.

In our previous system sizing estimates (SKA-TEL-SDP-0000038) we used our parametric models for the calibration, imaging and non-imaging processing pipelines to estimate the number of operations required to process data and took the “maximal case” as the system size driver (i.e. we assumed that an imaging experiment requiring the highest spatial and spectral resolution output and with the highest possible image dynamic range). We then assumed that these pipelines were required for a scheduling block of 6 hours’ duration and that the SDP therefore had 6 hours in which to process the data so that with a simple double buffer system the SDP could always be ingesting one 6 hour scheduling block’s data whilst processing the previous scheduling block’s data.

Whilst this assumption sized an SDP that could process whatever data came its way (simplifying the overall observatory scheduling) it would result in the procurement of an HPC system likely to be underutilised, since apart from any science goals, if a full spatial and spectral resolution image cube was generated once every 6 hours the observatory would be unable to transfer this cube (and its ancillary data products) off site as the data rate would be too high. Fortunately the science goals of Phase 1 SKA do not typically require high spatial, high spectral *and* high fidelity imaging all simultaneously: whilst the system must be capable of meeting each of the performance requirements in turn, it can deliver SKA science goals even if it is substantially smaller than the “maximal case” sized system.

We have previously already considered the system sizing for the average case, using the High Priority Science Objective (HPSO) experiments and model SKA schedule as representative. Our earlier analysis only used these HPSO averages to calculate mean power consumption (to see if we could deliver science within the power caps) but we can now easily extend this to our hardware system, according to the following methodology:

1. For each instrument, LOW and MID take the HPSOs from the model SKA schedule and use the SDP parametric models to estimate the operations count for each experiment.

2. Sum the total number of operations and total hours required for each instrument and use this to derive an average operations rate (FLOPS).
    a. For LOW we estimate that a total of 1.2×10^{24} floating point operations are required to deliver the HPSO science programme in 2.8 years of time-on-sky[1], giving an average required operations rate of 13.6 PFLOPS. For MID we estimate that a total of 2.9×10^{24} floating point operations are required to

[1] The actual HPSO programmes would require more time than this to run because PI time is expected to make up some fraction (nominally 30%) of the time, and also calibration overheads are not included. By adopting the HPSO average FLOPS value we are assuming that the range of HSPO scheduling blocks is representative of PI observing too. This introduces a substantial degree of uncertainty in the numbers.

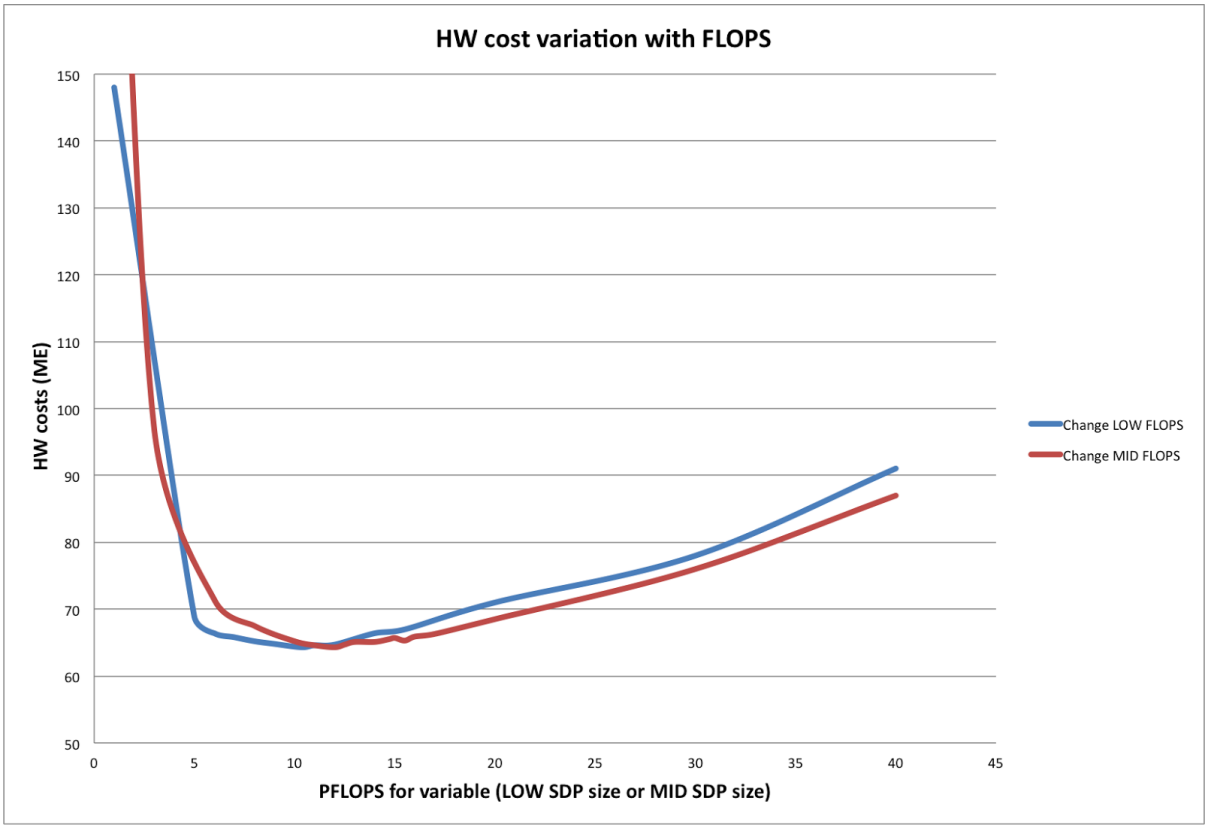
deliver the HPSO science programme in the 7.8 years of time-on-sky, giving an average required operations rate of 11.7 PFLOPS.

3. We then must consider how the system will react to different scheduling blocks: for example, we can still process the maximal case but we now do not have enough compute power to keep pace with the incoming data. This has several consequences:

   Scheduling blocks might take longer to process on the SDP than the on-sky time allocated to them. For example, for LOW our parametric model currently estimates that the maximal case would require 1.5×10^{21} operations for a 6 hour data set. This would take 30 hours to process with a system capable of delivering only 13.6 PFLOPS - i.e. it would take 5 times longer to process than to observe.

   Whilst the SDP is fully occupied processing this data set, the buffer must remain able to accept data for subsequent data reduction runs. Thus, a simple assumption might be that if the compute power of the system we build is a factor f times lower than that required to keep pace with the "maximal case", the buffer size must be able to hold (f+1) times the nominal single scheduling block visibility set size. (Up until now, our assumption has been that f=1, and we have assumed a double buffer.)
   Increasing the buffer size enables load balancing but this overall approach places constraints on the scheduling of the SKA: experiments with a high number of SDP operations required per unit of time-on-sky will need to be sandwiched between experiments with a lower numbers of operations required per unit of time-on-sky to ensure that the SDP does not run out of buffer space.

   It is possible that as the buffer gets larger and its cost becomes a greater fraction of the total we hit a minimum cost, after which further reduction of FLOPS and increase in buffer size will increases the total SDP cost. The graph below shows how the total HW cost varies if we change the FLOPS of the LOW SDP or the MID SDP in turn, and force the buffer to scale accordingly. There is a shallow minimum at around 10-15 PFLOPS. For FLOPS values above 15 PFLOPS or so we see a gentle, near-linear increase in cost but for values below 7 PFLOPS or so the buffer costs become very dominant and the cost increases dramatically. In fact the HSPO average FLOPS values for LOW and MID are extremely close to this minimum.



Varying FLOPS to minimise cost, subject to the constraint that we need at least 13.6 PFLOPS for LOW and at least 11.6 PFLOPS for MID **we get solutions of 13.8 PFLOPS for LOW and 12.1 PFLOPS for MID**.
Using these solutions we find

i. For LOW f=5.1 (70 PFLOPS maximal / 13.8 PFLOPS) so the buffer is 60.7 PBytes.
ii. For MID f=9.5 (115 PFLOPS maximal / 12.1 PFLOPS) and the buffer is 105 PBytes.

7.1. Possible phasing of the HPSO science

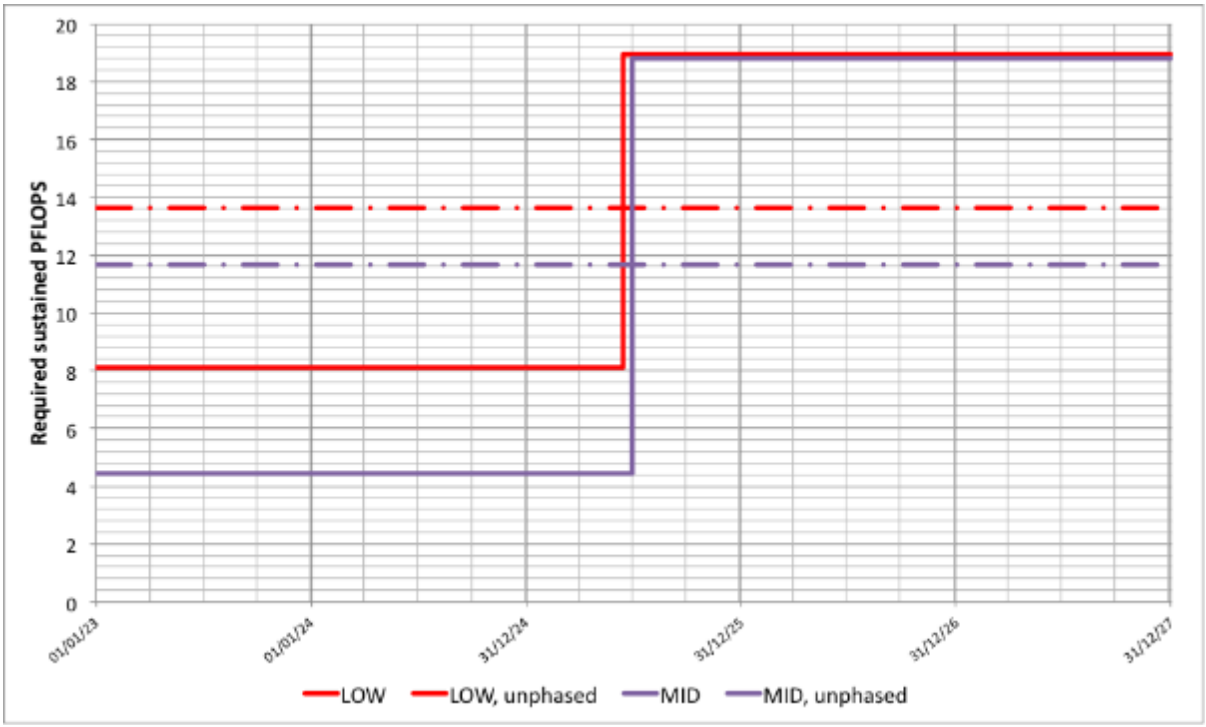
We have looked into the possibility of sizing the SDP to deliver HPSO science in a rolled-out manner in order to reduce cost and risk. A 5 year window was split into two phases and the HPSO science divided between these two phases, scheduling the low operations count experiments first (that is biasing the schedule towards non imaging experiments or short baseline experiments). We gave ourselves the constraint that after 5 years each HPSO experiment would be completed to the same fraction (i.e. for MID each would have been observed for around 45% of the total desired number of hours).

Example phased LOW schedule	Total hours for experiment	**Phase A**	**Phase B**	**FLOP rate for experiment[2]**
HPSO Experiment	(scaled to 5 years completion of HPSOs)	hours	hours	
1	4460	446	4014	48.2
2a	4460	1784	2676	42.8
2b	4460	1784	2676	42.8
4c	22747	13648	9099	0
5c	7672	3836	3836	0
		8.1 PFLOPS	**19.0 PFLOPS**	
		Start date	Start date	
		01/01/2023	14/06/2025	
		End date	End date	
		14/06/2025	31/12/2027	

[2] This is the compute rate (Flops) that would be required to process the data for each specific experiment's schedule block in the same length of time as the schedule block ran for - i.e. process 6 hours' of data in 6 hours. These numbers are achieved flops as estimated using our parametric models. The compute rate for the NIP experiments is very very low compared to the imaging experiments (E.g. ~500 GFLOPS or so.)

Example phased MID schedule	Total hours for experiment	**Phase A**	**Phase B**	**FLOP rate for experiment**
HPSO Experiment	(scaled to 5 years completion of HPSOs)	hours	hours	
4a	515	515	0	0
4b	1546	1546	0	0
5a	1031	618	412	0
5b	1031	618	412	0
13	3221	322	2899	5.1
14	1288	1288	0	3.0
15	8116	3246	4870	1.7
18 (and 32)	6441	6441	0	2.8
22	3865	0	3865	23.6
27 (and 33)	6441	6441	0	6.7
37c	6441	644	5797	29.5
37a	1288	64	1224	35.0
37b	1288	64	1224	36.6
38a	644	32	612	31.0
38b	644	0	644	33.7
	0			
		4.4 PFLOPS	**18.8 PFLOPS**	
		Start date	Start date	
		01/01/2023	29/06/2025	
		End date	End date	
		29/06/2025	31/12/2027	

Results are shown below. For the first 2 ½ years we can make the SDP substantially smaller than the average numbers given here but it will not be cheaper to do so:



For example, our proposed HPSO phasing means that the SDP for LOW (MID) could be only 8 (4.4) PFLOPS in the period Q1 2023 - Q2 2025 but it would then need to be enlarged to 19 (19) PFLOPS for the period Q3 2025 - Q4 2028 so that the total number of operations achieved in the 5 years was constant.

7.2. Cost of the phased HPSO hardware deployment

The tables below show the cost of phasing the hardware deployment for LOW and MID. The procurement dates used allow for 6 months' lead time for procurement, delivery and integration before the systems are operational. For the phased approach the full core network is deployed in phase 1 for practical reasons. The costs shown here are only hardware costs and include contingency.

LOW	**Procurement date**	**h/w cost**
Phase 1 (8 PFLOPS, 61 PB buffer, full core network)	2022/07/01	€22,359,387
Phase 2 (11 PFLOPS, no buffer, no network)	2024/01/01	€11,446,804
Total phased approach (19 PFLOPS, 61 PB buffer)		**€33,806,191**[1]
Phase 1 + 0.5*Phase 2 (to compare similar system lifetimes)		€28,082,789
Full system on day 1 (13.6[3] PFLOPS, 61 PB buffer)	2022/07/01	€27,291,673[2]

MID	Procurement date	h/w cost
Phase 1 (4.4 PFLOPS, 109 PB buffer, full core network)	2022/07/01	€29,703,888
Phase 2 (14.6 PFLOPS, no buffer, no network)	2024/01/01	€11,305,049
Total phased approach (19 PFLOPS, 109 PB buffer)		**€41,008,937**[1]
Phase 1 + 0.5*Phase 2 (to compare similar system lifetimes)		€35,356,412
Full system on day 1 (11.7^3 PFLOPS, 109 PB buffer)	2022/07/01	€36,502,678[2]

Notes:

1. The total cost for the phased approach cannot be directly compared to the unphased approach (full system on day one) as the total system lifetimes are not the same. The lifetime of Phase 2 extends 2.5 years beyond the lifetime of the full system.
2. The cost for the full system used here does not include the cost of Milli and Centi SDP deployments prior to the end of the construction phase.
3. The figures used for this comparison were based on a slightly earlier optimisation of FLOPS vs buffer.

Note that the difference in cost between the phased and unphased approach (over the same system lifetime) is small. Although the phased approach does not reduce the total cost of the hardware it would significantly reduce the financial impact associated with the uncertainty in the performance and cost estimate for SDP. The reduction in financial impact is a result of the following:

- The uncertainty (cost and performance) associated with phase 2 will be significantly lower than for phase 1 since phase 1 will have been operational for 2 years before phase 2 is procured.
- The financial resource wasted due to any underutilisation of SDP, caused by late delivery or technical issues of other elements (including SDP software), is reduced since the phase 1 deployment is smaller than the full system.
- The phased approach allows SKA to take advantage of new hardware technology advances which could reduce cost and/or power usage.
- The phased approach allows for time (2 years) to fix any issues found after deployment of phase 1 before phase 2 is deployed which also significantly reduces the risk associated with phase 2.

Since the phased approach offers significant reduction in risk it should therefore be discussed and analysed further.