SDP Memo: Estimating the SDP Computational Efficiency

Document number	SKA-TEL-SDP-0000086
Document type	REP
Revision	C
Author	Bojan Nikolic
Release date	2016-04-07
Document classification	Unrestricted
Status	Draft

Lead author:

Name	Designation	Affiliation
Bojan Nikolic	SDP Project Engineer	University of Cambridge
Signature & Date:	[illegible] Bojan Nikolic (Apr 7, 2016) b.nikolic@mrao.cam.ac.uk	

Released by:

Name	Designation	Affiliation
Paul Alexander	SDP Project Lead	University of Cambridge
Signature & Date:	Paul Alexander Paul Alexander (Apr 7, 2016) pa@mrao.cam.ac.uk	

Version	**Date of issue**	**Prepared by**	**Comments**
0.1	2016-04-01	Bojan Nikolic	Initial circulated draft
0.2	2016-04-05	Bojan Nikolic	Update with announced Pascal P100 figures – **note consequent update in efficiency prediction**
0.3	2016-04-06	Bojan Nikolic	Clarify text, change ξ to η
C	2016-04-07	Bojan Nikolic	Released for ΔPDR

ORGANISATION DETAILS

Name	Science Data Processor Consortium

Table of Contents

List of Abbreviations

CPU	Central Processing Unit
DRAM	Dynamic Random Access Memory
FFT	Fast Fourier Transform
FLOPS	Floating Point Operations per Second
GPU	Graphics Processing Unit
HBM	High Bandwidth Memory
SDP	Science Data Processor
SKA	Square Kilometre Array

List of Symbols

η_{comp}	Computational efficiency
η_{mem}	Maximum fraction of system power that can be used for memory access
ρ_{op}	Representative operational intensity of SDP calculations
E_{mem}	Energy required per access to bit
R_{FLOP}	Number of floating point operations computed per unit time
R_{mem}	Data transferred between memory and a processing unit per unit time

Applicable and Reference Documents

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**

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-0000040 – Bolton, R. et al.: Parametric models of SDP compute requirements
RD02	Barnes, L.: SKA-SDP Gridding on Graphical Processing Units, NVIDIA Corporation, 2015, Report prepared under contract to the University of Cambridge
RD03	Barnes, L.: Characterising FFT performance on GPUs for SKA-SDP, NVIDIA Corporation, 2016, Report prepared under contract to the University of Cambridge
RD04	Harris, M.: Inside Pascal: NVIDIAs Newest Computing Platform, `https://devblogs.nvidia.com/parallelforall/inside-pascal/`
RD05	Williams, S. and Waterman, A. and Patterson, D.: Roofline: An Insightful Visual Performance Model for Multicore Architectures, Communications of the ACM, 2009, vol. 52, no. 4, doi:10.1145/1498765.1498785
RD06	SKA-TEL-SDP-0000058 – Salvini, S.: FFT Analysis, 2015
RD07	O'Connor, M.: Highlights of the High-Bandwidth Memory (HBM) Standard, NVIDIA Corporation, 2014, The Memory Forum – June 14, 2014

1 Introduction

Estimating the actual achieved performance of the future SDP system is important for optimising the SKA designs at both system and SDP levels and for good planning of budgets. In this memo we set out a method to make this estimate and summarise the experimental evidence from prototyping to support this method.

2 Overall Approach

The approach follows the roofline methodology described by RD05 with floating point operations and working memory bandwidth as possible limiting factors. The logic employed is as follows:

1. The number of floating point operations required to carry out the required SDP processing steps is estimated using the SDP parametric model (RD01).
2. The working memory transfer volume required per operation is measured and/or inferred from benchmark runs of representative kernels on hardware today. This gives the representative operational intensity for the SDP computations, $\rho_{\rm op}$.
3. The achieved performance of SDP computations on future hardware is estimated by considering the aggregate FLOPS ($R_{\rm FLOP}$) and memory bandwidth ($R_{\rm mem}$) for the system and identifying which of these is the limiting factor. If the limiting factor is memory bandwidth then the aggregate system memory bandwidth is converted into the corresponding maximum achieved floating operation rate.

It is possible to summarise the result of the last two step of this logical process as *computational efficiency* which is the ratio of predicted achieved floating point operation rate divided by the maximum possible floating point rate if memory bandwidth is not a limiting factor:

$$\eta_{\rm comp} = \frac{R_{\rm FLOP}^{\rm achieved}}{R_{\rm FLOP}^{\rm max}}. \tag{1}$$

This single number however somewhat obscures the critical logical dependence on the aggregate memory bandwidth of the system being considered. The best way of expressing this result is through a set of inequalities such as the "design equations" in RD01.

To summarise: the approach presented here uses *today's* empirical operational intensity and our estimates of *future* FLOPS and memory bandwidth capabilities.

3 Experimental Evidence

In this section we review the experimental evidence of operational intensity of the key SDP computational kernels, i.e., gridding and FFT.

The first report we consider is by RD02. The relevant findings are:

1. Gridding performance is limited by memory bandwidth between the accelerator caches and memory on the accelerator.
2. The maximum achieved performance for gridding in the best case is $R_{\rm FLOP}^{\rm achieved}$ =120 GFLOPS.

The maximum memory bandwidth for the accelerator used in this report (NVidia K40) is R_{mem} =288 GigaByte/s. Together with the findings above, this gives an estimate of best-case operational intensity of ρ_{op} ∼0.4 FLOPS/byte.

The second report we consider is RD03. The relevant findings are:

1. The maximum FFT performance (on an NVidia K40) reached for realistic grid sizes is $R_{\mathrm{FLOP}}^{\mathrm{achieved}}$ =226 GFLOPS (similar throughput is measured by RD06).
2. The DRAM bus is reasonably well used but not completely saturated.
3. Future architectures from Pascal are expected to fully saturate the DRAM bus and therefore will be memory bandwidth limited.

Given the above we can estimate an operational efficiency (which strictly is only a lower bound estimate but probably is close to the true value) of $\rho_{\mathrm{op}} \geq 0.8$.

We next consider FFT throughput measurements by RD06 on an Intel CPU. The relevant finding is that the maximum performance is around $R_{\mathrm{FLOP}}^{\mathrm{achieved}} = 32\,\mathrm{GFLOP\,s^{-1}}$ for a system with maximum memory bandwidth of $R_{\mathrm{mem}} = 51\,\mathrm{GB\,s^{-1}}$. The report points out that performance is limited by data movement. Therefore we can estimate an operational intensity of $\rho_{\mathrm{op}} \sim 0.6$.

A simple average of the results in this section gives an approximate single estimate of the operational intensity of SDP computation of $\rho_{\mathrm{op}} \sim 0.6$. Gridding and FFT are the two most demanding task in SDP computations according to the parametric model (RD01).

The measurements presented here are the *best measurements* within realistic parameter bounds. Achieving these already assumes optimised kernels with pipeline processing tailored for the hardware used.

4 Estimate of Future Computational Efficiency

In the present method, the estimate of future computational efficiency depends on future FLOPS and memory bandwidth figures for the computational units being considered. Since this dependence is typically a dependence on the ratio, uncertainty in the figures are potentially magnified in the efficiency. For most practical purposes the aggregate memory bandwidth required should be considered as discussed in the next section. Nevertheless in this section we make some extrapolations to estimate potential computational efficiency. This is the methodology originally used in September 2014 to estimate the computational efficiency of SDP hardware, leading to an estimate of 25%. Updates to the input numbers used in this methodology have since reduced the efficiency estimated by this method.

We make an extrapolation from the performance of current GPUs although there is nothing to say that the SDP will indeed run on GPU-like hardware. The estimate is made as follows:

1. The forthcoming Pascal GPU architecture will have a memory bandwidth of around $R_{\mathrm{mem}} \sim 720\,\mathrm{GB\,s^{-1}}$ and floating point throughput of around $R_{\mathrm{FLOP}}^{\mathrm{max}} \sim 5$ TeraFLOPS^{-1} (RD04).
2. Taking the above-estimated *average* computational intensity of $\rho_{\mathrm{op}} \sim 0.6$, the achieved performance on these units should be around $R_{\mathrm{FLOP}}^{\mathrm{achieved}} \sim 430\,\mathrm{GFLOP\,s^{-1}}$.
3. The computational efficiency can therefore be estimated as $\eta_{\mathrm{comp}} = \frac{R_{\mathrm{FLOP}}^{\mathrm{achieved}}}{R_{\mathrm{FLOP}}^{\mathrm{max}}} \sim 0.09$.

4. We then extrapolate that future GPUs and other CPUs/accelerators will maintain similar computational efficiency. The (relatively weak) justification for this assumption is that the computational efficiency calculated in this way has remained approximately constant between the Kepler and Pascal architectures which are separated in time by four years.

This leads to an estimate of computational efficiency for SDP of $\eta_{\mathrm{comp}} \sim 0.09$. To reiterate, this efficiency is lower than that assumed in the SDP costing in March 2016 due to a change in inputs into this methodology during the period September 2014 – March 2016 during which we have not updated the assumed efficiency.

5 Using Memory Bandwidth Directly

Since memory bandwidth rather than floating point operation execution will almost certainly limit the performance of future SDP systems, it is more instructive to consider the aggregate memory bandwidth requirement without reference to computational efficiency.

The calculations are straightforward. As an *example* if an SDP system requires an achieved 100 PFLOPS, then according the above estimated computational intensity it will require an aggregate memory bandwidth of 170 PByte/s.

For a given a memory architecture this can be converted directly into a system sizing. For example, *current* HBM memory architecture is expected to achieve a maximum $R_{\mathrm{mem}}^{\mathrm{HBMstack}} = 256\,\mathrm{GBs}^{-1}$ of bandwidth per stack (RD07). This means that the putative SDP system illustrated in this section will need a minimum of $\frac{R_{\mathrm{mem}}}{R_{\mathrm{mem}}^{\mathrm{HBMstack}}} \sim 7 \times 10^5$ HBM memory stacks. If, for example, 6 stacks per processor/accelerator package were integrated that would correspond to a total of about 120,000 processor/accelerator packages.

The memory bandwidth requirement can also be directly converted into an estimated power requirement by estimating energy per memory access, E_{mem}. RD07 writes that HBM requires around $E_{\mathrm{mem}} \sim 6\,\mathrm{pJ\,bit}^{-1}$ of energy to access each bit. The requirement for $R_{\mathrm{mem}} = 170\,\mathrm{PByte\,s}^{-1}$ bandwidth can then be converted to an estimated power for memory system of $P_{\mathrm{mem}}^{\mathrm{sys}} = R_{\mathrm{mem}} E_{\mathrm{mem}} \sim 8.2\,\mathrm{MW}$. We assume that for all practical systems no more than a third of available power could potentially go to memory, i.e., system efficiency of $\eta_{\mathrm{mem}} \sim 1/3$. This leading to system power estimate of $P^{\mathrm{sys}} = P_{\mathrm{mem,agg}}^{\mathrm{sys}}/\eta_{\mathrm{mem}} \sim 25\,\mathrm{MW}$.

Summarising: following the logic presented in this memo, and with an HBM memory architecture, a system *achieving* 100 PFLOPS for SDP computations will consume a minimum of 25 MW and have a minimum of 100,000 CPU/Accelerator packages.

Reducing these estimates would require either increasing the effective operational intensity, e.g., by doing far more faceting which enables gridding and FFTs using only on-die cache memory, or new faster and more power-efficient memory technologies past HBM.