Where is the Slack in the SDP System?

Markus Dolensky
SKA Signal & Data Path along Design

LFAA
DISH

Correlator
CSP

Data Intensive
Astronomy

Science Data
Processor

Data Intensive
Astronomy

Science Data
Processor
SDP Design Challenge revisited

• Power Budget: <5 MW
• Compute Efficiency: target is 10 % [FLOP/W]
• Sustain Throughput: 0.4 TByte/s/telescope

• Optimize Data Locality
• Error Resilience
• Automated Calibration
• Multiplicity of Input Streams
• Variety of Observing Modes
• ...


Growing Storage Pyramid

- Registers
- Level 1 Cache
- Level 2 Cache
- Main Memory
- Solid-State Disk
- Fixed Rigid Disk
- Optical Disks (Jukeboxes)
- Magnetic Tapes (Robotic libraries)
- USB Flash Drives
- Removable Hard Drives
- System
  - Online (Secondary)
  - Tertiary
  - Off-Line

Access Times:
- 0.3ns–2ns
- 3ns–8ns
- 6ns–20ns
- 30ns–70ns
- 35μs–100μs
- 3ms–15ms
- 100ms–5s
- 10s–3m
- 0.5ms–33ms
- 12ms–40ms

Larger

More Costly

Less Costly
Doubling Interferometer means 8x Processing - $O(n^3)$ => quickly dominates CAPEx
CAGR and Doubling Time anCAGR
40% CAGR equals 24 month doubling rate
Regaining or Losing a Factor of 10?

The rebaselining downsized by a factor of 10; deployment baseline another 5x
To regain a factor of 50 takes many years depending on CAGR.
The rebaselining downsized by a factor of 10; deployment baseline another 5x
To regain a factor of 50 takes many years depending on CAGR.
Performance Evolution

(Energy) Efficiency vs. Time

1970 ~2004 ~2015 >>2025

Device Scaling Multi-Core Multi-Thread Workload Optimized

holistic optimization of SW & HW stack

Source: Schmatz, IBM
# Numerical Precision: NZ-AU Study Team

## SPECIFICATIONS

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU Architecture</td>
<td>NVIDIA Pascal</td>
</tr>
<tr>
<td>NVIDIA CUDA® Cores</td>
<td>3584</td>
</tr>
<tr>
<td>Double-Precision Performance</td>
<td>4.7 TeraFLOPS</td>
</tr>
<tr>
<td>Single-Precision Performance</td>
<td>9.3 TeraFLOPS</td>
</tr>
<tr>
<td>Half-Precision Performance</td>
<td>18.7 TeraFLOPS</td>
</tr>
<tr>
<td>GPU Memory</td>
<td>16GB CoWoS HBM2 at 732 GB/s or 12GB CoWoS HBM2 at 549 GB/s</td>
</tr>
<tr>
<td>System Interface</td>
<td>PCIe Gen3</td>
</tr>
<tr>
<td>Max Power Consumption</td>
<td>250 W</td>
</tr>
<tr>
<td>ECC</td>
<td>Yes</td>
</tr>
<tr>
<td>Thermal Solution</td>
<td>Passive</td>
</tr>
<tr>
<td>Form Factor</td>
<td>PCIe Full Height/Length</td>
</tr>
<tr>
<td>Compute APIs</td>
<td>CUDA, DirectCompute, OpenCL™, OpenACC</td>
</tr>
</tbody>
</table>

TeraFLOPS measurements with NVIDIA GPU Boost™ technology

Source: NVIDIA Tesla P100 Data Sheet
Unum Number Format by Gustafson

Sign-Magnitude Integer (80 bits):

IEEE Standard Float (64 bits):

Unum (29 bits, here):

Self-descriptive “utag” bits track and manage uncertainty, exponent size, and fraction size. More compact, accurate and efficient.
Overhaul of CASA Core Models

In addition to CASA Overhaul: SKA wide review of data reduction/calibration approach

Prof. Philip Diamond, SKA Director-General, and Dr. Tony Beasley, Director of the US National Radio Astronomy Observatory, signing a Memorandum of Understanding between the two organisations on CASA workpackage collaboration.
parallel IO interface abstracts IO from application

plots: Storage Manager test runs with CASA, SKA-TEL-SDP-0000024, Data Challenge Supplement

https://github.com/ornladios/ADIOS2/wiki
NVMe augmented Storage Node Performance
CRC Calc Rate on Xeon Phi (KNL)
Data Formats

- HDF5
- CASA MS
- PSRFITS
- VOTable
- JPEG2000
- JPIP/JPEG2000 for Visualization
Container Technology, Cluster Management
Machine Learning - Deep Learning

RFI Detection
Object Classification

380 papers, 4300 citations in Astronomy
courtesy: A. Robotham
1. Goal: Minimize buckets and valves in plumbing (middle section of cartoon)
   - parallel streams instead of loops
   - reduction of IOPS
   - potential further gains in conjunction with SDN
   - like any parallel programming methodology S/W complexity increases

2. Current Work:
   - mandatory for Ingest Pipeline, i.e., bulk interface CSP/SDP (talk D. Deveraux)
   - OSKAR2 simulator streaming output adapter (F. Dulwich)
   - supports SPEED protocol (i.e. correlator output) to MS conversion
   - successful integration with DALiuGE workflow on CSIRO's Bracewell cluster
Nyriad GPU-accelerated storage solutions enable:

- Superior data resilience
- Reduced hardware costs
- Lower power consumption
- Instant recovery from errors and failures
- Reduced network fabric costs
- Real-time data provenance and security

Say Goodbye to RAID™

NSULATE is a GPU-accelerated block storage device that utilises massively parallel processing to provide extremely resilient data storage.

It replaces traditional, static RAID-based storage controllers that were optimised for the IT world 25 years ago. NSULATE is a highly dynamic parallel storage engine for high performance parity calculation, integrity checking and storage management. It enables support for hundreds of HDD, SSD or NVM storage devices to be combined to provide highly resilient local or distributed storage as a standard Linux block device.
Nyriad GPU-accelerated storage solutions enable:

- Superior data resilience
- Reduced hardware costs
- Lower power consumption
- Instant recovery from hardware failures
- Reduced network fabric costs
- Real-time data provenance and security

Say Goodbye to RAID™

NSULATE is a GPU-accelerated block storage device that uses massively parallel processing to provide extremely resilient data storage. It replaces traditional, static RAID-based storage controllers that were optimised for the IT world 22 years ago. NSULATE is a highly dynamic checking and storage management. It enables support for hundreds of HDD, SSD or NVM storage devices to be combined to provide highly resilient local or distributed storage as a standard Linux block device.

Maximum of 2 parity (p and q) no matter the amount of data

RABD can be configured to store up to 127 parity blocks with its data with a maximum recommended configuration being 96 data blocks with 32 parity across 128 logical devices.
RABD Deployment on MWA Buffer

- SKA Pre-cursor for SDP Persistent Storage
- Buffer for MWA data delivery that uses Next Generation Archive System (NGAS) as transport framework
- RABD as resilient block device backing OS filesystem
- Total of 10 nodes each with 45 x 8 TByte drives
- 5 nodes using traditional, well tested RAID0 of 3 x RAID6 volumes
- 5 nodes using Nyriad RABD solution with 6 parity drives out of 45
- P1000 GPU for parity calculations with fallback to CPU
- Utilising agile development processes with continuous integration testing to full specification test nodes
- Testing using threading sync reads and/or writes as performed by archiving and storage system (NGAS)
Developing Roadmaps to (SKA) Regional Centres

EU

Advanced European Network of E-infrastructures for Astronomy with the SKA

Funded: 15.07.16

Australia

Exascale Research Infrastructure for Data in Asian-Pacific astroNomy Using the SKA

Launched 10.04.17

http://eridanus.net.au
Separation of Concerns facilitated by DALiuGE

- **Algorithms**: best algorithm to get the desired answer
- **Pipeline Logic**: reduction components and sequence
- **Component Parameters**: default parameter values of components
- **Data Parallelisation**: hints about the potential of parallelism
- **Parallel Execution**: what is executed where
- **Parallel Coding**: writing parallel code
- **Code Optimisation**: optimise parallel code
- **I/O Optimisation**: optimise I/O on hardware
- **OS and hardware co-design**: optimise hardware for code to be run

Roles:
- Astronomer
- Operator
- HPC Software Engineer
- OS level S/W Engineer
- Computer H/W Engineer
SDP Execution Framework Prototype DALiuGE

**DALiuGE** – **Data-Activated Liu (流) Graph Engine**

- [https://github.com/ICRAR/daliuge](https://github.com/ICRAR/daliuge)
- 20,000 lines of Python code + minimal C library code
- refs
Data-centric Approach

+ control of data locality
+ exploitation of data parallelism
+ works with commodity hardware
+ no global file system required, just a GUID server like Ceph
+ DAG based control flow; not limited by single master node
+ load-balancing
+ fault tolerance
- idle times when waiting for matching data; stragglers
Code Reuse - JACAL (Joint Astronomy imaging and CALibration)

JACAL - a Rialto project deliverable:

- Investigate reuse of ASKAPsoft in context of SKA1-Low
- Integrate with DALiuGE to scale system performance
- Resulting source package JACAL (ASKAPsoft & DALiuGE)
- Determine workload characteristics/data access patterns of algorithmic components
- Reuse MWA archival data and codes as stepping stone to SKA1-Low
Scheduling: DAG; NP hard
Thank You!