SDP ARCHITECTURAL DESIGN LAST MILE

- What?
- How?
- Hot off oven

MYM 2018-07-12 AUT

tn@compucon.co.nz

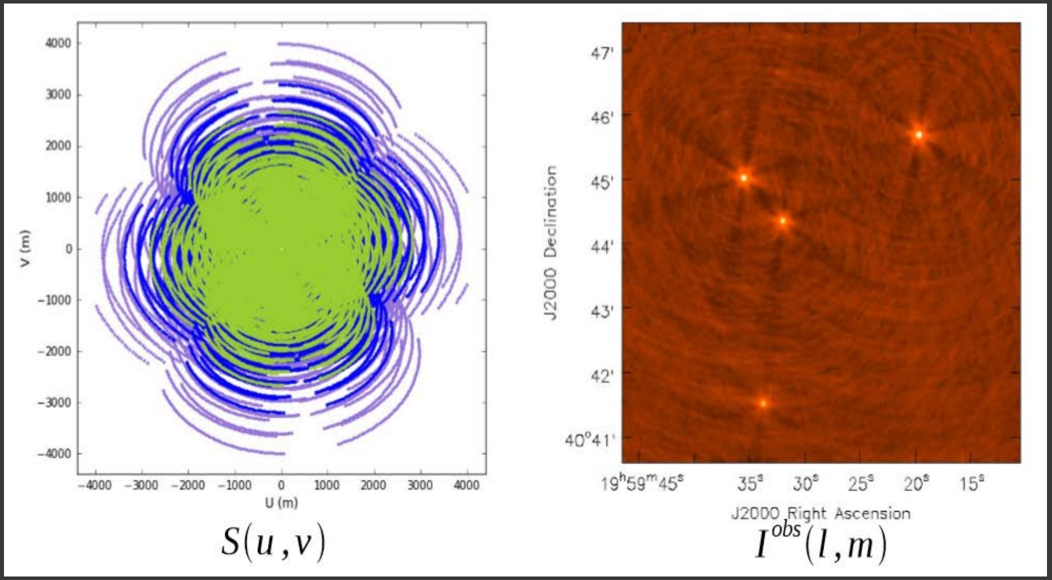
Song Huang

Background of SDP

Two supercomputers for converting sky signals data into visibility

- **Design Teams from Compute Node perspective**
  - **Cambridge**
    - Architecture, execution framework, science pipelines
  - **Oxford**
    - vertical prototyping and integration
  - **NZ**
    - Could be the 3rd prominent team for science algorithms, middleware, **compute node**.

Imaging the Sky



- **van-Cittert Zernike theorem** relates spatial frequency coverage with observed image
  - Grid dish signals onto spatial uv map (27 ant, 4 hr, 3 chan)
  - FFT to 2D observed image
  - Clean observed image of coverage deficiency

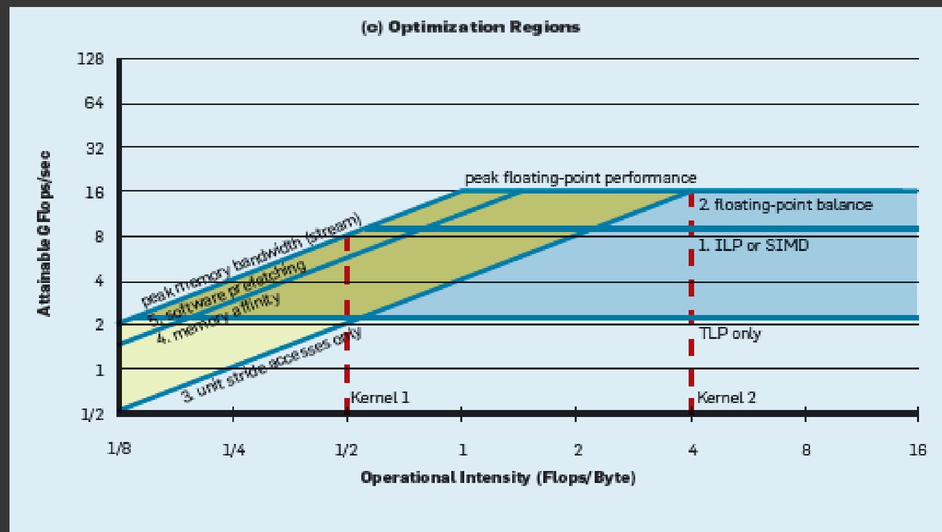
Last Mile: Compute Node Co-design

new investigation topics besides #3 and #5

Longer mile

1. Compute Efficiency Model (NZA) → SDP Memo w/Oxford
2. Compute Node Properties (NZA) → SDP Memo w/ASTRON
3. SDP System Sizing (NZA/Oxford) → SDP Memo
4. Negative Scaling (Tim Cornwell) → NZA
5. Data Movement (Cambridge) → NZA w/Oxford

1. Compute Efficiency



Roofline Model from Berkeley

- **Composite Operational Intensity of Pipeline**

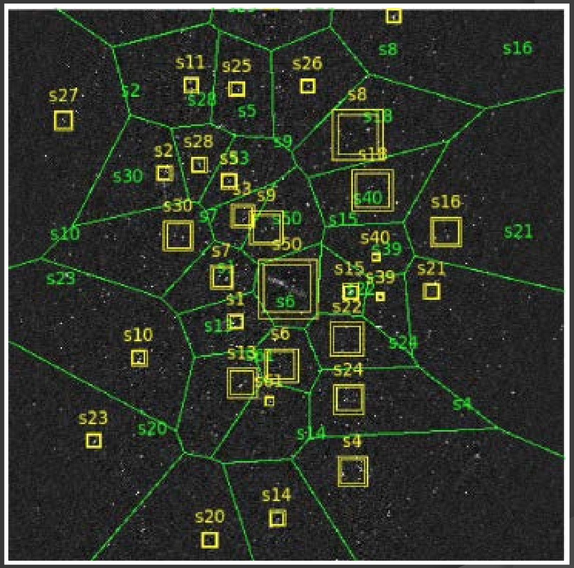
  (OI)cip = Σ [Wi * (OI)i] if (OI)i < (OI)r + Σ [Wi * (OI)r] if (OI)i >= (OI)r

- **Compute Efficiency, Roofline Efficiency & Programming Efficiency**

  CE = RE * PE

2. Compute Node

- Assume **1500 nodes** per SDP for pre-defined HPSO including CIP
  - High Priority Science Objectives
  - Continuum Imaging Pipeline
- Properties
  - Grid size limited by memory footprint (64k x 64k no can do)
    - Facet is desirable
  - ARL code 40% sequential
    - Amdahl Law
  - Time weighting of FFT : Gridding : MSMFS
  - Abstraction of Python, numpy, numba, pyCUDA, DASK

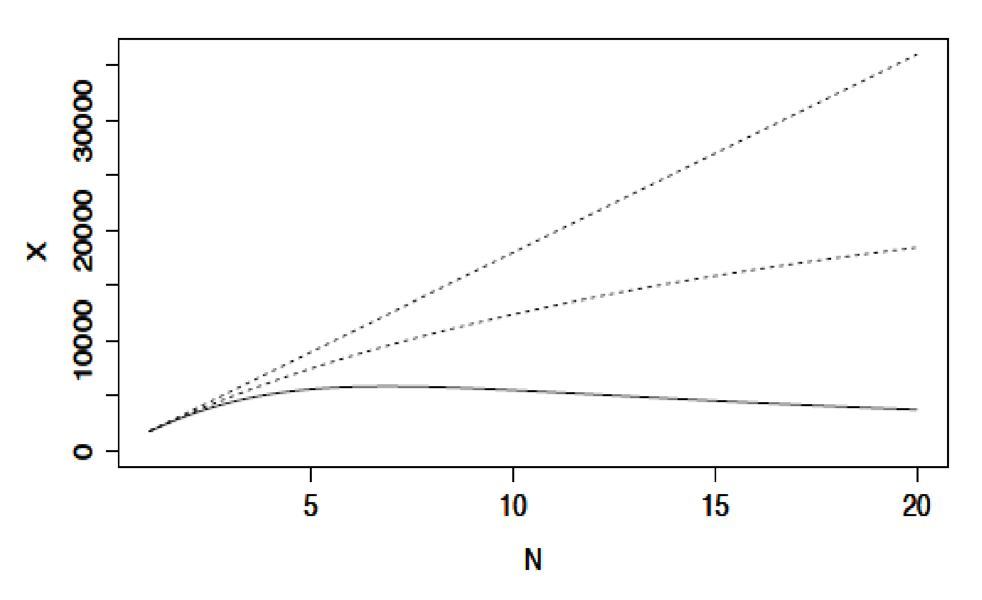


3. SDP System Sizing

How many compute nodes are needed to run science pipelines?

- **NZA Mission**
  - Review previous assumption of SDP system sizing based on Compute Efficiency of 10% for CIP
- **Outcome of Investigation with CIP**
  - Invalidated previous assumption
  - Could reach 75% for GPU
- **Next Steps**
  - Investigate ICAL (more complex)
  - Oxford optimisation skills essential

4. Negative Scaling



Compute Efficiency

Scaling Efficiency

- Speed Up = N / (1+a*(N-1)+b*N*(N-1))

  N is number of compute nodes or workers
  a = 0 in above for kernels to be executed in GPU
  b is unknown for sublinear scaling

5. Big Data Movement

13PB/s

- Last investigated by Cambridge in 2016-05
  - Several crises of data flow bottleneck
  - Global synchronization blocks
- Splits- multiply them together!
  - Time slices, baselines, frequencies, facets, polarisations, Taylor terms, and PSF
- Scenarios:
  - Storage access, Network transfer, Memory, transfer, Streams

Summary

- **SKA Pre-Construction Phase ends 2018-12**
  - **Huge learning curve for all involved due to an unprecedented scale**
  - **NZ team has done well in CSP and SDP as Andrew explained- how do we translate our NZA successes into NZ benefits? Q&A for 4:40pm**