Software, Computing and Data Storage for the Square Kilometre Array

Duncan Hall
XLDB 2009
2009 August 28
Outline

SKA: What are the drivers?
How does radio astronomy work?
What are the SKA’s prime characteristics?
Project Phases: Where are we at?
What are the possible operating modes?
Can large software really be that hard?
Real-time data: pushing the HPC envelope
How much data do we need to store?
Summary
A global project
55 institutes in 19 countries

Similarities to CERN?
Science drivers

- **Origins**
  - Cosmology and galaxy evolution
  - Galaxies, dark matter and dark energy
  - Probing the “Dark Ages”
  - Formation of the first stars
  - Cradle of life
  - Search for signs of life

- **Fundamental Forces**
  - Strong-field tests of general relativity
  - Was Einstein correct?
  - Origin and evolution of cosmic magnetism
  - Where does magnetism come from?

- **Exploration of the Unknown**

Adapted from R. Schilizzi
Answering the questions:
The journalists’ view?

New Electronics

A better view of the skies

The Square Kilometre Array is set to provide astronomers with unprecedented views of what’s out there – and opportunities for UK electronics.
How does radio astronomy work?
76 m: 3,000+ tonnes
How telescopes work

Conventional telescopes reflect the light of a distant object from a parabolic surface to a focus
A partially filled aperture ...

But the reflecting surfaces do not need to be part of the same surface.

Suppose we cover up most of the surface of the mirror.

M. Longair via R. Schilizzi
... can produce images

We can still combine the radiation from the uncovered sections to create an image of the distant object, if we arrange the path lengths to the focus to be the same.

M. Longair via R. Schilizzi
To make sure that the waves travel the same distance when they are combined, we need to add this path difference to the waves arriving at the other telescope.

The incoming waves are in phase.
Radio interferometry

- Each pair of antennas is called a baseline
- More different baselines → more detailed the image
- Short baselines - antennas are close to each other - provide coarse structure
- Long baselines provide the fine detail: the longer → the finer the detail

A 6 antenna interferometer has 15 baselines
1974: removing artefacts

Figure 1 Example of a baseline coverage obtained with the Green Bank Interferometer. The measured points have been mirrored through the origin to give a better impression of the overall structure.

Resultant point spread function

Figure 2  The “dirty beam” corresponding to the u,v coverage shown in figure 1. Contours are drawn at 5, 10, 15, 20, 30 etc. % of the beam maximum (top center). No distinction has been made in the figure between positive and negative contours; half of the sidelobes are in fact negative.

Figure 3 Illustrating the effect of the CLEAN procedure on measurements at 2695 MHz of the radio source 3C 244.1 taken with the u-v coverage shown in figure 1. Contours are drawn at the same intensity ratios as in figure 2. a) the “dirty map” b) cleaned map after one iteration with the loop gain γ=1 and subsequent removal of the clean beam, c) same, but after two iterations and the removal of the two fainter sources. The next preceding component is extended and there are some weaker components present. d) 6 iterations. The further improvement here is due to the cleaning of the side lobes from the less intense features that remain after the two first iterations.
What are the SKA’s prime characteristics?
EVLA 25 m dish
Inside an EVLA dish
SKA: prime characteristics

1. More collecting area: \( \sim 1 \text{km}^2 \)
   - Detect and image hydrogen in the early universe
   - Sensitivity \( \sim 50 \times \) EVLA, LOFAR

2. Bigger field of view
   - Fast surveying capability over the whole sky
   - Survey speed \( \sim 1,000,000 \times \) EVLA

3. Wide ranges of frequencies
   - Low: 70-300 MHz
   - Mid: 300 MHz-10 GHz
   - High: 10-25+ GHz

4. Large physical extent: \( \sim 3,000+ \) km
   - Detailed imaging of compact objects
   - Astrometry with \( \sim 0.001 \) arc second angular resolution

Adapted from R. Schilizzi
• 1,500 dishes (~15m diameter) in a 5 km core
• Additional 1,500 dishes from 5 km to ~3,000+ km
• Aperture arrays (AA) in a core
• Signal processing: (1) beam forming
• Optical fibre connection to (2) correlator
• Optical fibre to remote (3) High Performance Computer
One possible configuration

Comms links + power

Dishes in stations along spiral arms

Correlator

40 stations 10-200 km

40 remote stations 200 to >3,000 km

Station

Max. Distance for Dense AAs 200 km

Dishes
Dense AA
Sparse AA

Adapted from A. Faulkner
Where are we at?
Phase 1:
- 2013 – 2018 construction
- 10-20% of the collecting area

Phase 2:
- 2018 – 2022 construction
- Full array at low and mid frequencies

Phase 3:
- 2023+ construction
- High frequencies
Current development

PrepSKA 2008-2011

The Preparatory Phase for the SKA is being funded by the European Commission’s 7th Framework Program

€5.5M EC funding for 3 years + €17M contributed funding from partners (still growing)
€150M SKA-related R&D around the world

Coordinated by the Science and Technology Facilities Council (UK)
WP2: Design + Cost

Coordinated by the SKA Program Development Office in Manchester

- System Definition
- Dishes, feeds, receivers
- Aperture arrays
- Signal transport
- Signal processing
- Software
- High performance computers
- Data storage
- Power requirements
What are the possible operating modes?
One possible operational mode:

# Grammar for Python

# Note: Changing the grammar specified in this file will most likely require corresponding changes in the parser module
# (.*/Modules/parsermodule.c). If you can’t make the changes to that module yourself, please co-ordinate the required changes with someone who can; ask around on python-dev or help fred
# Drake <fdrake@acm.org> will probably be listening there.

# NOTE WELL: You should also follow all the steps listed in PEP 306, "How to Change Python’s Grammar"

# Start symbols for the grammar:
# single_input is a single interactive statement;
# file_input is a module or sequence of commands read from an input file;
# eval_input is the input for the eval() and input() functions.
# NB: compound_stmt in single_input is followed by extra NEWLINE!

single_input: NEWLINE | simple_stmt | compound_stmt NEWLINE

file_input: (NEWLINE | stmt)* ENDMARKER

eval_input: testlist NEWLINE* ENDMARKER

decorator: '@' dotted_name [ '(' [arglist] ')' ] NEWLINE
decorators: decorator+
decorated: decorators (classdef | funcdef)
funcdef: 'def' NAME parameters ['->' test] ':' suite
parameters: '(' ['typedargslist'] ')' typedargslist: (tfpdef ['=' test] ',')*

(')' [tfpdef] (',', tfpdef ['=' test])* [',', '**' tfpdef] | '**' tfpdef)

| tfpdef ['=' test] (',', tfpdef ['=' test])* [',',]
tfpdef: NAME [':=' test]
vararglist: ((vpdef ['=' test] ',')*
Another possible mode:

Group 3: Button-Pushing Astronomers

The ideal astronomer UI (Tony Willis):

- **GO**
- **GO FASTER**
- **DO WHAT I MEAN!**

NB: this picture is meant to be purely representative

Yet another possible mode
Can large software really be that hard?
Yes!

- First-order model for estimating effort
- Diseconomies of scale
- Confirmation from the literature
Sound familiar?

- Over-commitment
  - Frequent difficulty in making commitments that staff can meet with an orderly engineering process
- Often resulting in a series of crises
  - During crises projects typically abandon planned procedures and revert to coding and testing
- In spite of ad hoc or chaotic processes, can develop products that work
  - However typically cost and time budgets are not met
- Success depends on individual competencies and/or “death march” heroics
  - Can’t be repeated unless the same individuals work on the next project
- Capability is a characteristic of individuals, not the organisation

An ill-conditioned non-linear problem:

\[ \text{Effort} (\$) = \prod_{P, S > 1} \begin{bmatrix} \text{Problem space} \\ \text{Solution space} \end{bmatrix} \times \begin{bmatrix} \frac{d}{dt} \text{Problem space} \end{bmatrix} \]

where

\[
\begin{bmatrix}
\text{Problem space} \\
\text{Solution space}
\end{bmatrix} = \begin{bmatrix}
\text{Size (+)} \\
\text{Complexity (+)} \\
\text{Time (-)} \\
\text{Interfaces (+)} \\
\text{Reliability (+)} \\
\end{bmatrix} \times \begin{bmatrix}
\text{Requirements} \\
\text{People skills (-)} \\
\text{Colocation (-)} \\
\text{Processes (-)} \\
\text{Prior experience (-)} \\
\text{Reuse opportunities (-)} \\
\text{Toolsets (-)} \\
\end{bmatrix}
\]
First-order model for estimating effort
McConnell’s data on log-log axes

Estimated Effort for Scientific Systems & Engineering Research Projects

Staff Years = 2.0(-5) x (SLOC)^1.33

Staff Years = 4.7(-6) x (SLOC)^1.36

McConnell diseconomy


Staff Years

Source Lines of Code (SLOC)
2008Nov11 Case study c.f. McConnell data

Estimated Effort for Scientific Systems & Engineering Research Projects

Staff Years = 2.0(-5) x (SLOC)^1.33

Staff Years = 4.7(-6) x (SLOC)^1.36

Staff Years = 4.5(-11) x SLOC^2.4

McConnell diseconomy

Case study 1 diseconomy

Astronomers developed legacy codes

guest editors’ introduction

End-User Software Engineering
How big are the “legacy” codes?

Table I. Sample community codes for radio astronomy imaging.

<table>
<thead>
<tr>
<th>Package name</th>
<th>Development languages (ordered by prevalence)</th>
<th>Size (MSLOC(^a))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Astronomical Image Processing System (AIPS)(^b)</td>
<td>Fortran 77, C</td>
<td>0.6</td>
</tr>
<tr>
<td>Multi-channel Image Reconstruction, Image Analysis, and Display (MIRIAD)(^c)</td>
<td>Fortran 77, C</td>
<td>0.2</td>
</tr>
<tr>
<td>Astronomical Information Processing System (AIPS++)(^d)</td>
<td>C++, Glish [14], Fortran 77</td>
<td>1.0</td>
</tr>
</tbody>
</table>

\(^a\)MSLOC = 10^6 SLOC, as measured by SLOCCount (written by David A. Wheeler).

\(^b\)Modified version of base release 15 OCT 97.

\(^c\)Release v4.

\(^d\)Modified version of code base v1.8 #667.


**MSLOC:**

- Debian 4.0: 283
- Mac OS X 10.4: 86
- Vista: 50
- Linux kernel 2.6.29: 11
- OpenSolaris: 10
Legacy: ~20 to ~700+ staff years effort?

**Estimated Effort for Scientific Systems & Engineering Research Projects**


\[
\text{Staff Years} = 2.0(-5) \times (\text{SLOC})^{1.33}
\]

\[
\text{Staff Years} = 4.7(-6) \times (\text{SLOC})^{1.36}
\]

\[
\text{Staff Years} = 4.5(-11) \times \text{SLOC}^{2.4}
\]

+1 Standard Deviation

Avg. Staff Years (=12 months)

Source Lines of Code (SLOC)

http://www.skatelescope.org/pages/WP2_Meeting_10Nov08/Day2-3_Hall.pdf;
A software intensive system product is much more than the initial algorithm:

One problem the large science experiments face is that software is an out-of-control expense. They budget 25% or so for software and end up paying a lot more. The extra software costs are often hidden in other parts of the project – the instrument control system software may be hidden in the instrument budget.
SKA real-time data: pushing the HPC envelope...
Φ2 real-time data from dishes

SKA Conceptual
High Level
Block Diagram

80 Gbs⁻¹

Outlying
Station

Outlying
Station

Outlying
Station

Outlying
Station

80 Gbs⁻¹

~40

~2,280

80 Gbs⁻¹ per dish

Dish Array
Wide band single pixel feeds (WBSPF)
Phased array feeds (PAF)

Aperture Array

Processing Facility
Digital Signal Processing
Beamforming
Correlation

Computing Facility
High Performance Computing
Data Storage

Regional Science Centre(s)
Regional Engineering Centre(s)

0.1 ~ 1 ExaFlops;
0.1 ~ 1 ExaByte storage

SKA HQ
(Off Site)

Global

On Site

J. Cordes “The Square Kilometre Array – Astro2010 RFI #2 Ground Response” 27 July 2009; Table 1, pp 9 - 10
Pushing the HPC envelope

Projected Performance Development

Performance [TFlops] = 0.055e^{0.622(year-1993)}

SKAΦ2 ~1 EFlop

SKAΦ1 ~10 PFlop ~100 TFlop

ASKAP

Orders of magnitude are required:

World's 500 Most Powerful and Efficient Computers [2007 - 2009 Green500]:
Gigaflops vs kWatts (≡Mflops vs Watts)

Source: http://www.green500.org/
accessed 2009-Jul10

http://www.green500.org accessed July 2009
How much data do we need to store?
Large datasets challenge applications in practice.

Scale up your datasets enough and your apps come undone. What are the typical problems and where do the bottlenecks surface?

BY ADAM JACOBS

The Pathologies of Big Data

A database on the order of 100GB would have been far too expensive, and requiring the operators to manually mount and dismount thousands of 40MB tapes would have slowed progress to a crawl, or at the very least severely limited the kinds of questions that could be asked about the census data.

A database on the order of 100GB would not be considered trivially small even today, although hard drives capable of storing 10 times as much can be had for less than $100 at any computer store. The U.S. Census database included many different datasets of varying sizes, but let's simplify a bit: 100GB is enough to store at least the basic demographic information—age, sex, income, ethnicity, language, religion, housing status, and location, packed in a 128-bit record—for every living human being on the planet. This would create a table of 6.75 billion rows and maybe 10 columns. Should that still be considered “big data”? It depends, of course, on what you're trying to do with it. Certainly, you could store it on $10 worth...
500-GB tapes switched to 1-TB models – an upgrade that took a year of continuous load/read/load/write/discard operations, running in the interstices between the data centre’s higher-priority tasks.

Disk storage: annual 50% cost reduction

1 EB = $1~$10 million

Note: neither RAID, controllers, nor interconnect cables are included in these estimates

Power for EB-size disk looks reasonable


Note: power numbers here are for the drives only; any electronics associated with drive controllers [e.g. ECC, RAID] needs to be counted separately
Summary

It is hard workin' on this stuff.

Yep, son, we have met the enemy and he is us.