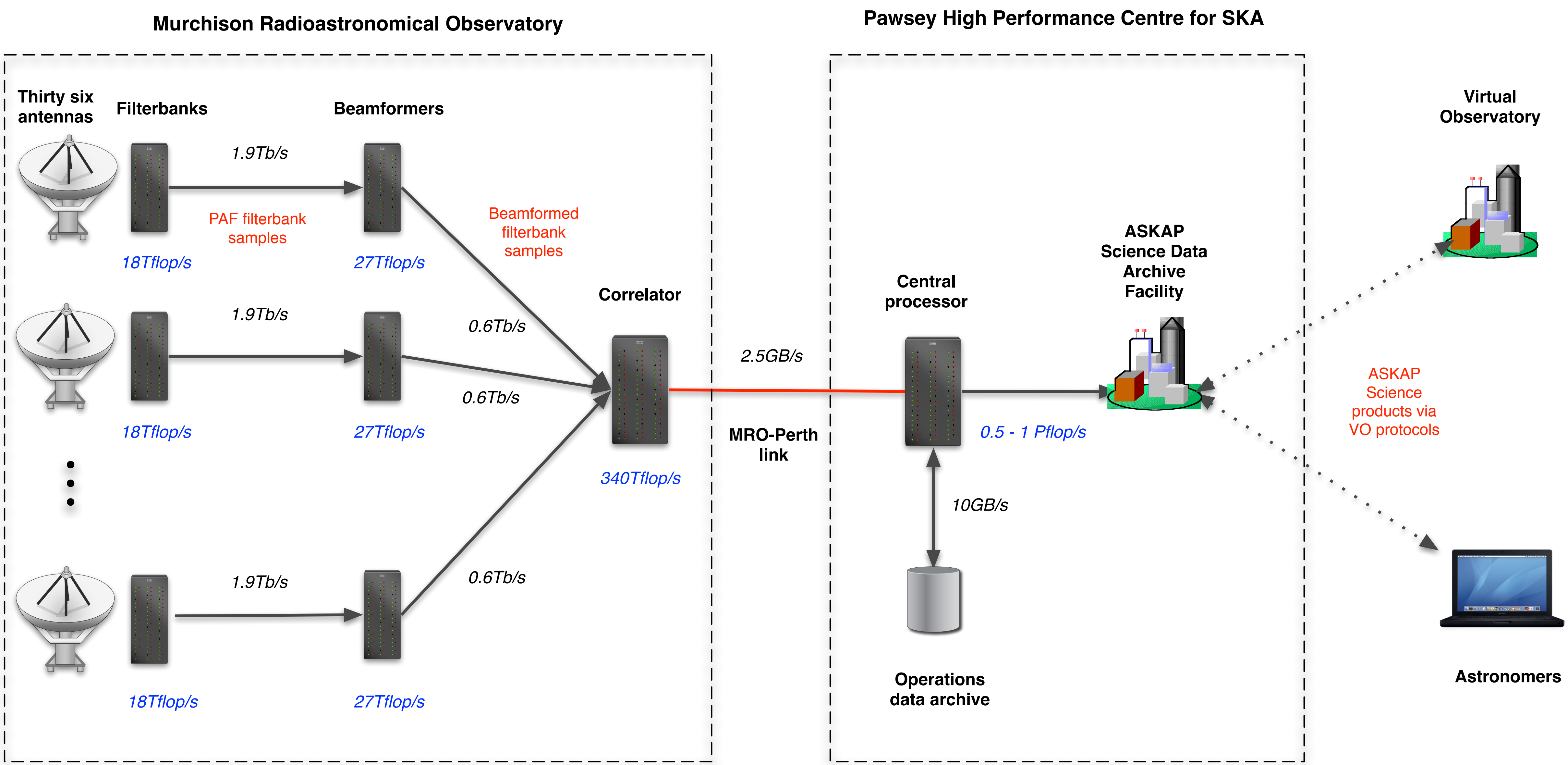
www.csiro.au

Scaling for ASKAP and SKA1

Tim Cornwell
Ben Humphreys
Maxim Voronkov
Australian Square Kilometre Array Pathfinder

CSIRO

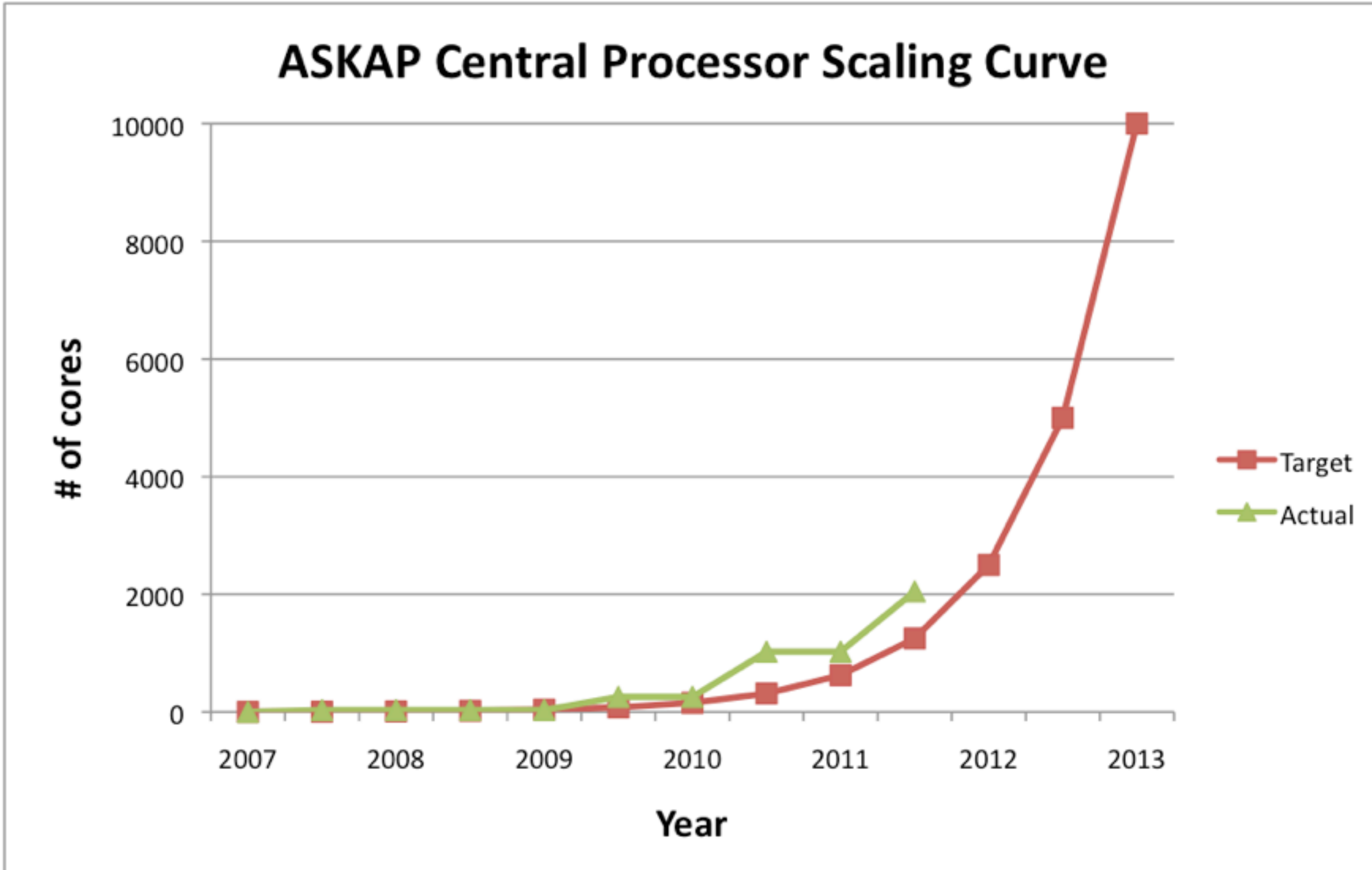
ASKAP data flow



- From observing to archive with no human decision making
  - Calibrate automatically
  - Image automatically ~ 80 TB per 8 hour observation
  - Form science oriented catalogues automatically

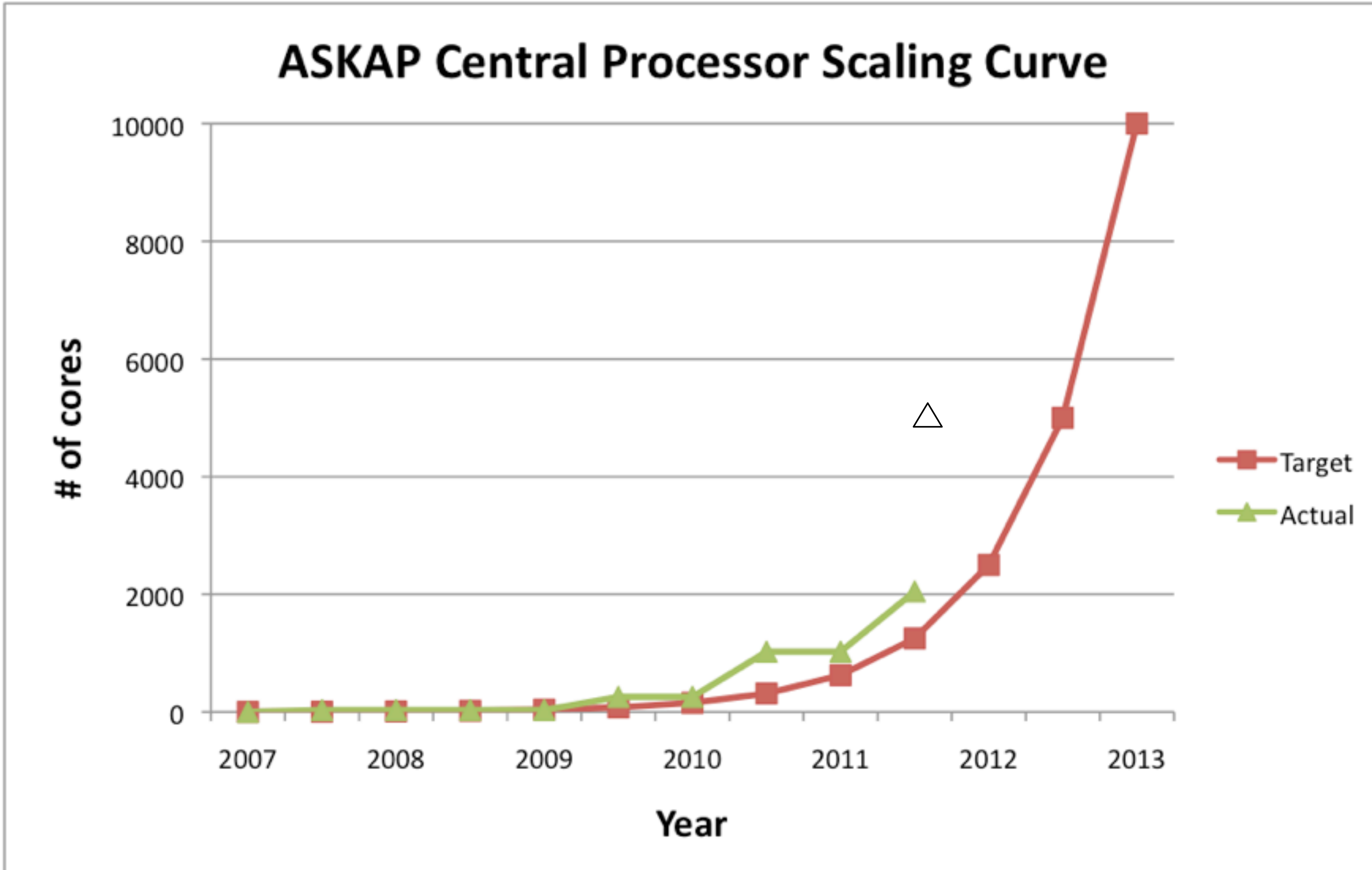
T. Cornwell, July 9 2010

Improvement of scaling since project initiation



Calim 2011

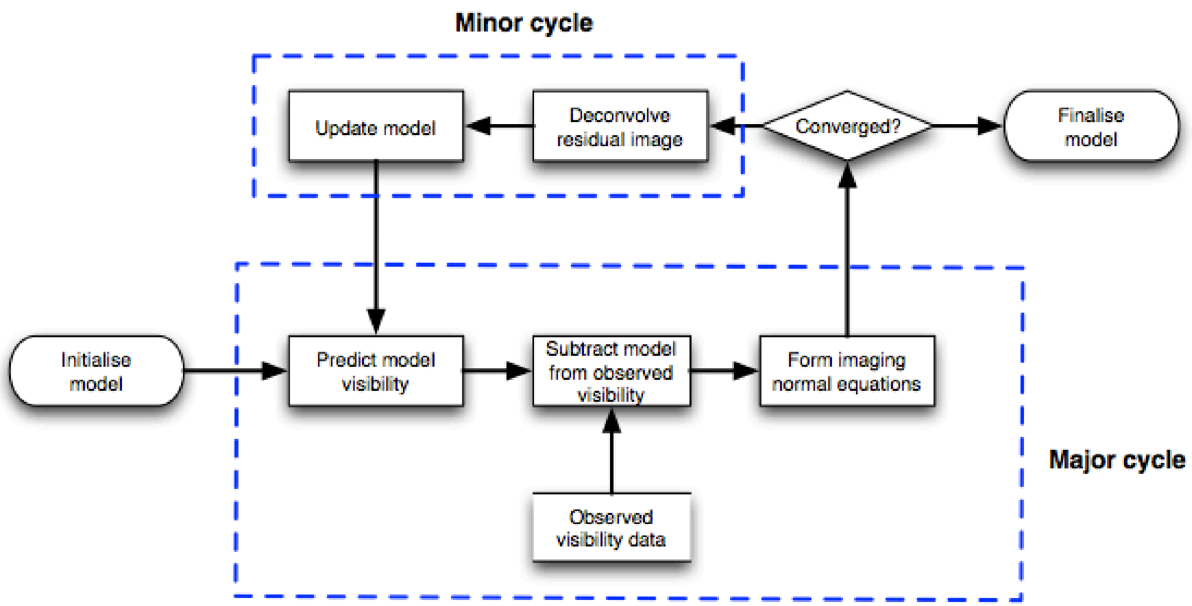
Improvement of scaling since project initiation



Calim 2011

Overview of ASKAP imaging

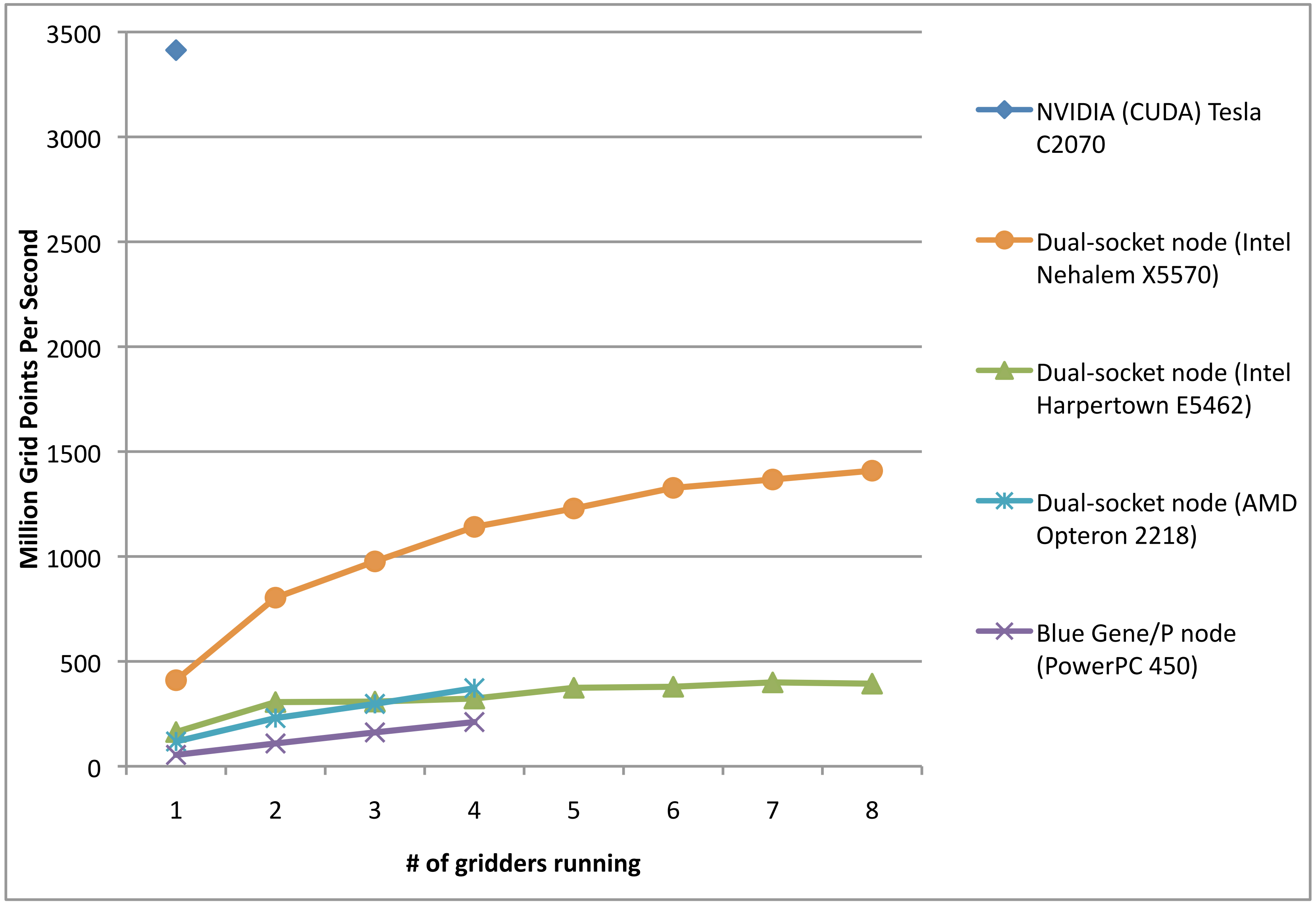
- Necessary to develop calibration and imaging code from scratch
- 1 MS per chunk of frequencies
- All MS's stored on common Lustre file system
- All images in memory
- Master/worker pattern
- Master distributes model image, accumulates normal equations, deconvolves
- Workers accept model and calculate normal equations
- 1 master, many workers



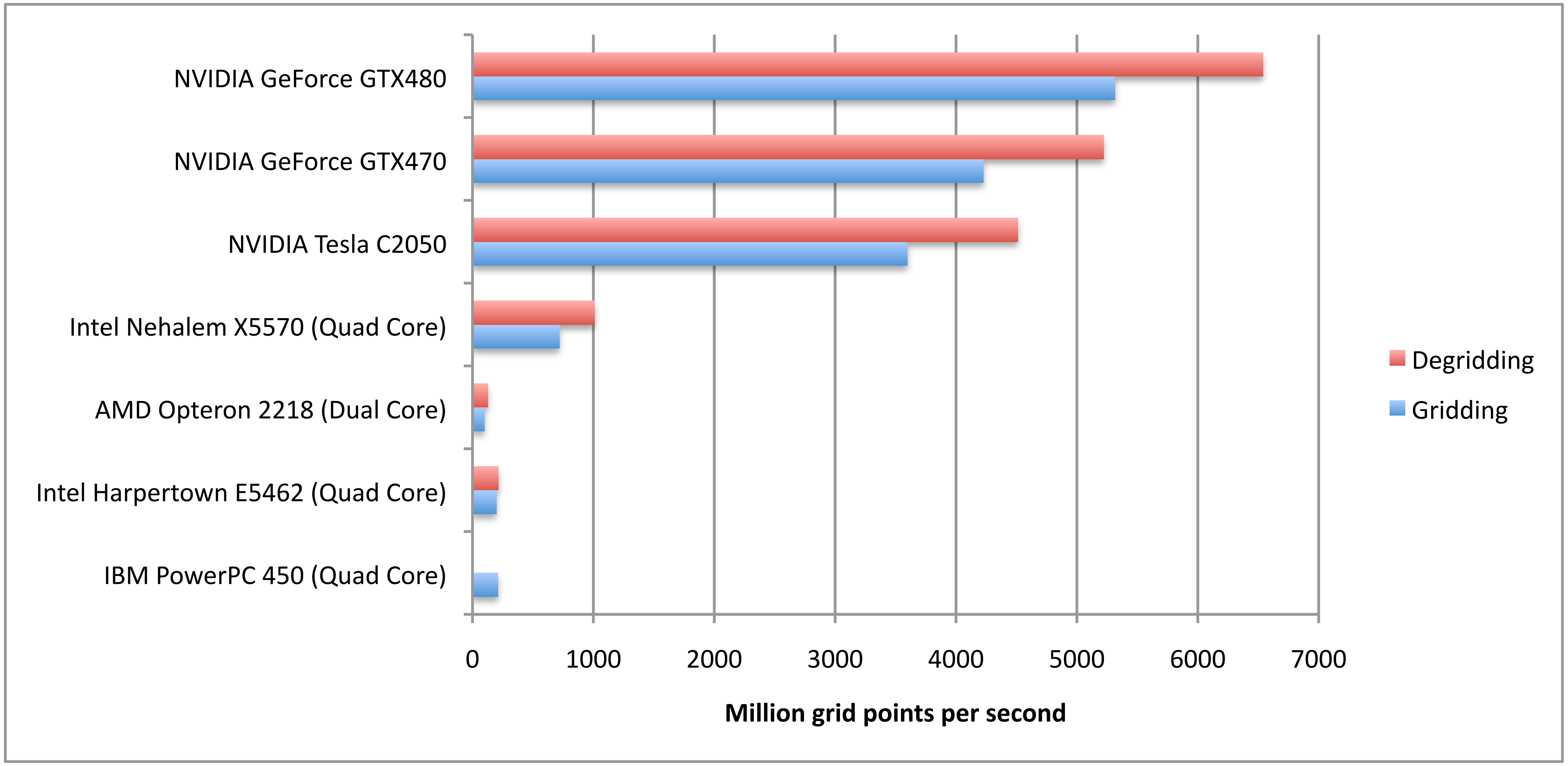
- One MPI process per core
- Eventually will move to one MPI process per node
  - And OpenMP for processing within a node

Scaling of gridding inside one node

- Humphreys and Cornwell, SKA memo 128

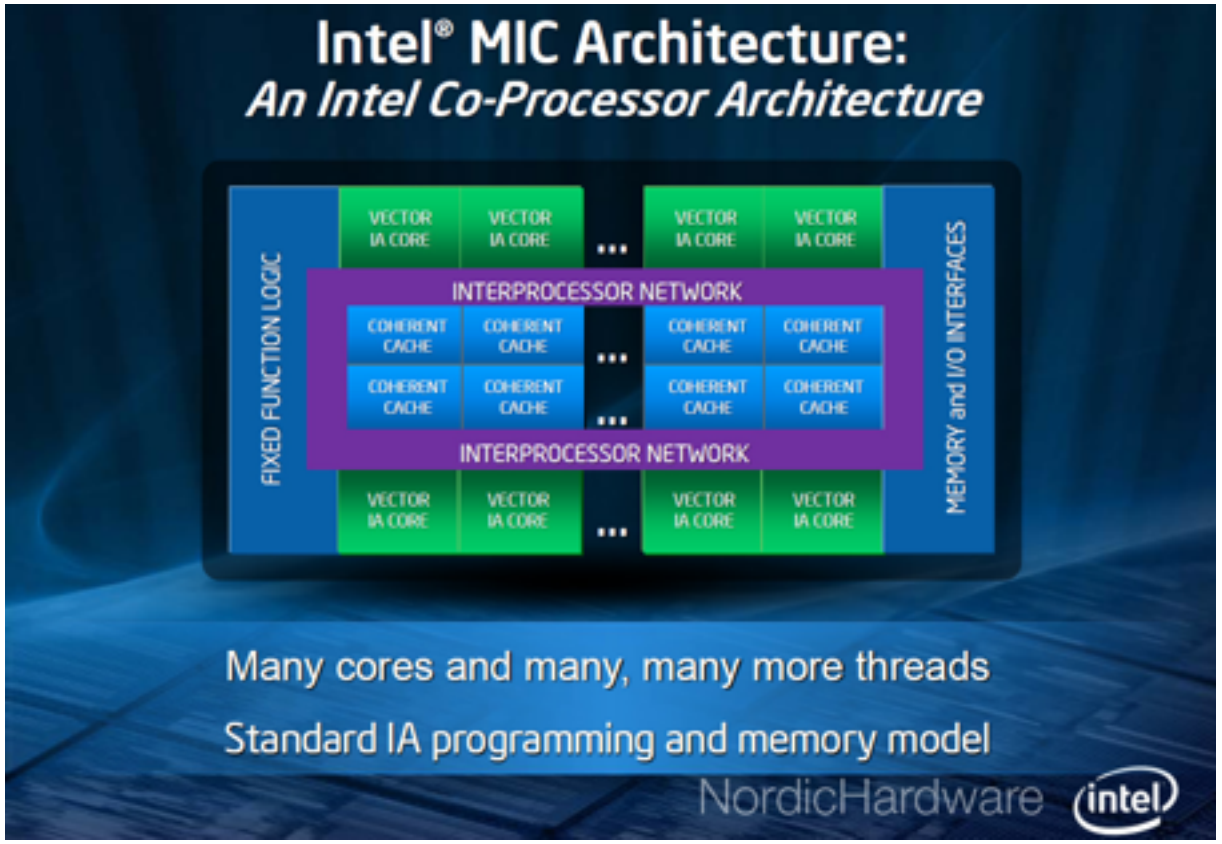


Overall gridding performance



Intel Many Integrated Core system

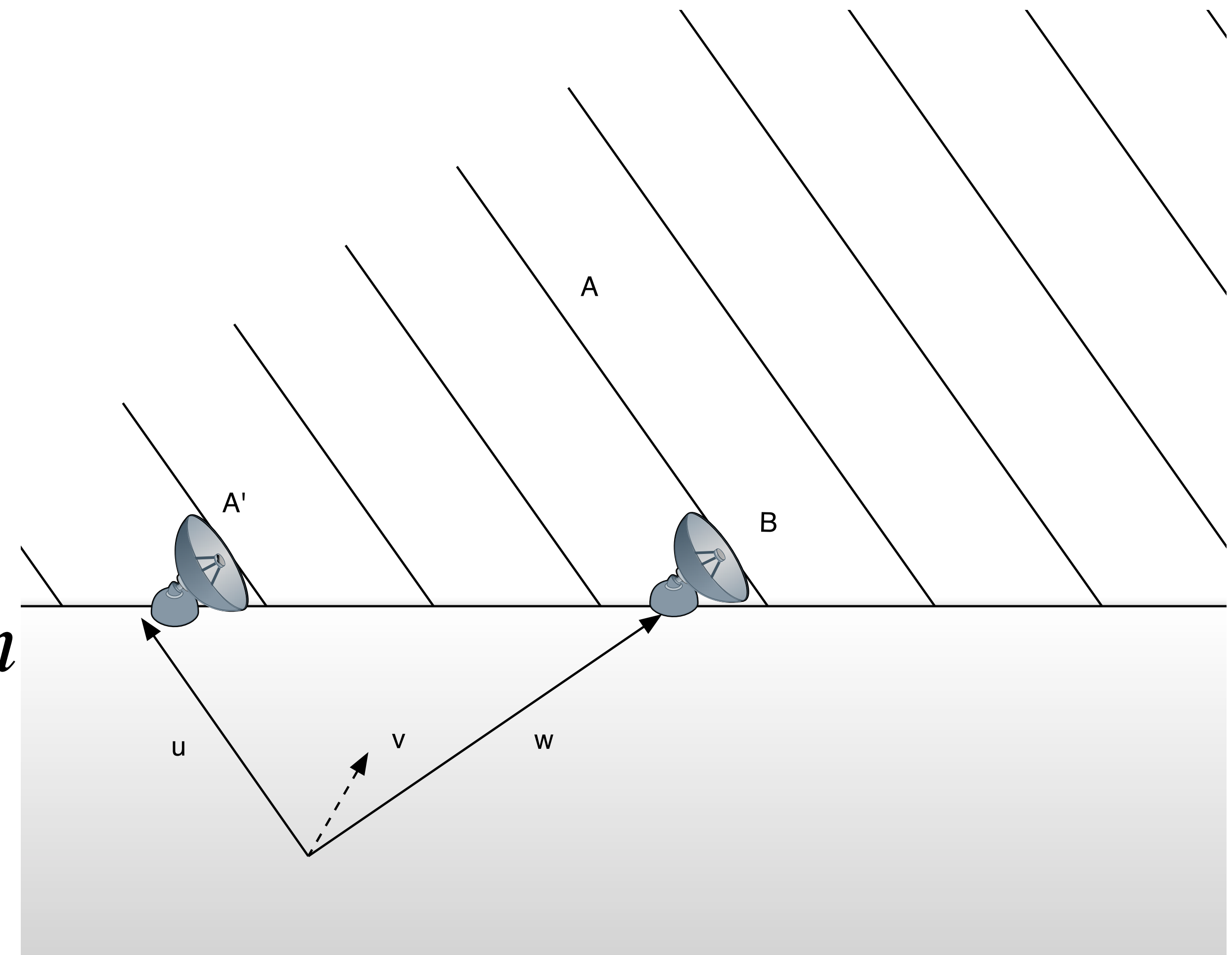
- Intel Many Integrated Core (MIC) beta program
- '*Knights Ferry*' development platform.
  - http://www.intel.com/technology/architecture-silicon/mic/index.htm
- 22nm, up to 50 cores
- The highlight of the product is the software development stack
  - Allows simple OpenMP directives to be added to existing code to offload processing to the MIC card.
- The software stack lived up to expectations, with a gridding kernel/benchmark port taking about 30 minutes to port.
- A simple implementation of a Hogbom clean was also developed for the MIC.
- Results under NDA



Radio telescope imaging

- Spatial coherence of electric field (visibility) is Fourier transform of sky brightness

$$V_{A'B} = \left\langle E_{A'} E_B^* \right\rangle_t$$
$$= e^{-2\pi jw} \int I(l,m) e^{-2\pi j(ul+vm)} dldm$$

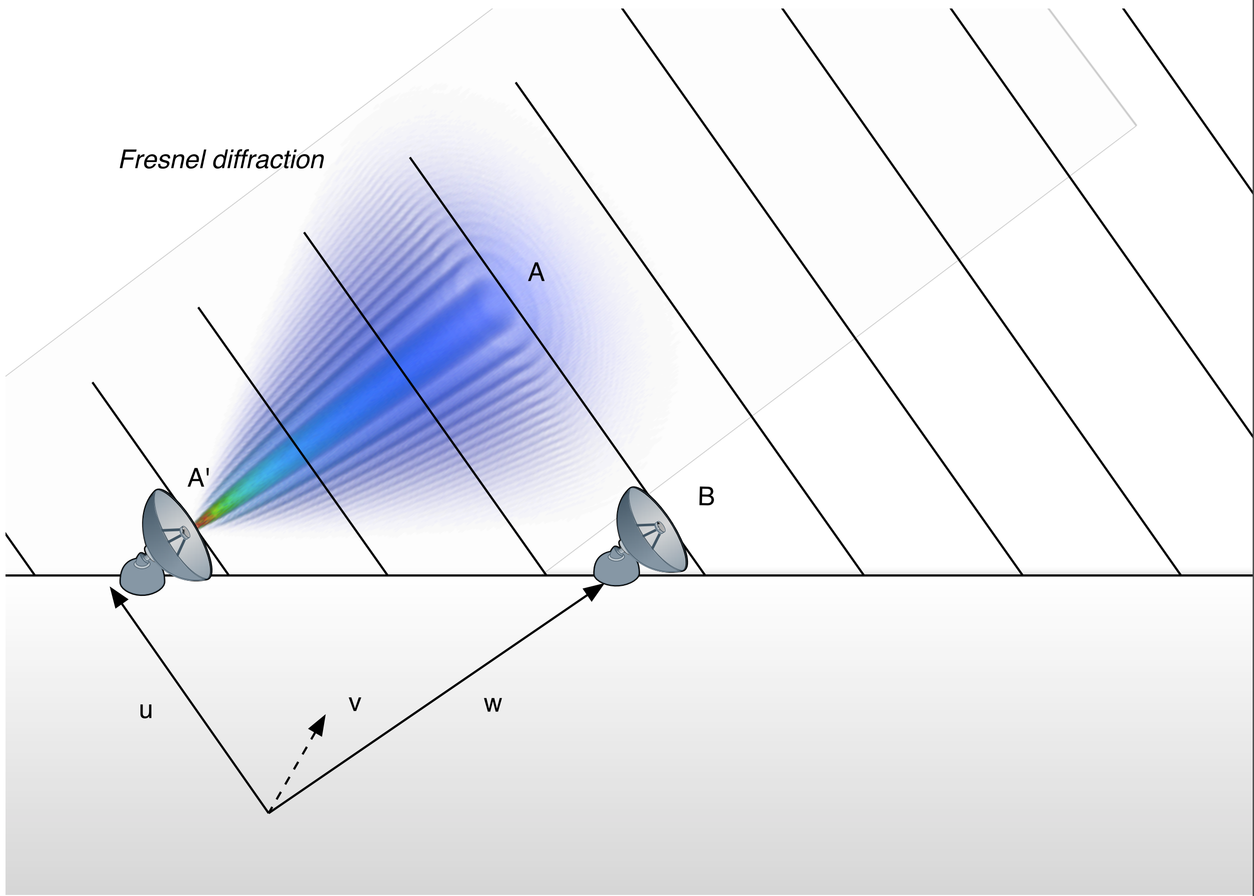


- Measures for many values of the Fourier components u,v
- Invert Fourier relationship to get image of sky brightness
- Typical problems
  - Incomplete u,v, sampling
  - Calibration
  - Wide field of view: no longer Fourier transform

Wide field imaging

- Visibility on plane A'B is Fourier transform of sky

$$V_{A'B} = e^{-2\pi jw} \int I(l,m) e^{-2\pi j(ul+vm)} dl dm$$



- Antenna A' receives radiation by Fresnel diffraction from AB plane
- Visibility between A' and B is Fresnel/ Fourier transform

$$V_{A'B} = \int I(l,m) e^{-2\pi j\left(ul+vm+w\sqrt{1-l^2-m^2}\right)} \frac{dl dm}{\sqrt{1-l^2-m^2}}$$

- Not invertible
- Use iterative algorithm
  - Predict forward with high accuracy
  - Reverse calculation is approximate
  - Apply prior information *e.g.* sparse

Calim 2011

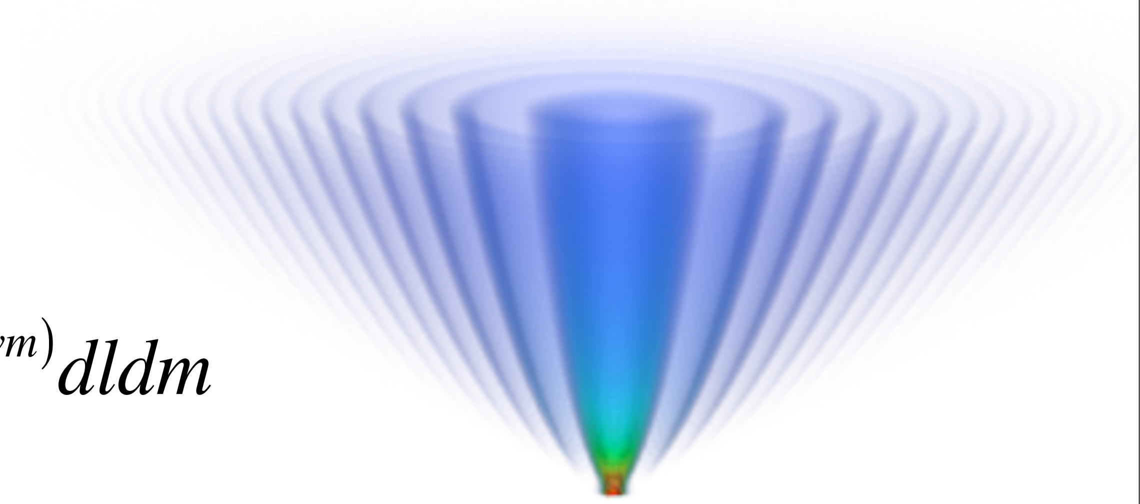
CSIRO

W projection algorithm

- Re-arrange imaging equation

$$V(u,v,w) = \tilde{G}(u,v,w) \otimes V(u,v)$$

$$\tilde{G}(u,v,w) = \int \frac{e^{j2\pi w\left(\sqrt{1-l^2-m^2}-1\right)}}{\sqrt{1-l^2-m^2}} e^{j2\pi(ul+vm)} dldm$$



- Algorithm
    - Given a trial image *I(l,m),* Fourier transform to single grid
    - Tabulate *w*
    - Calculate convolution function for each w
    - For each visibility, calculate trial value using anti-aliasing filter
- Problems
    - Large number of flops
    - Memory for convolution function can be very large

Snapshot imaging

- Plane varies with zenith angle and parallactic angle

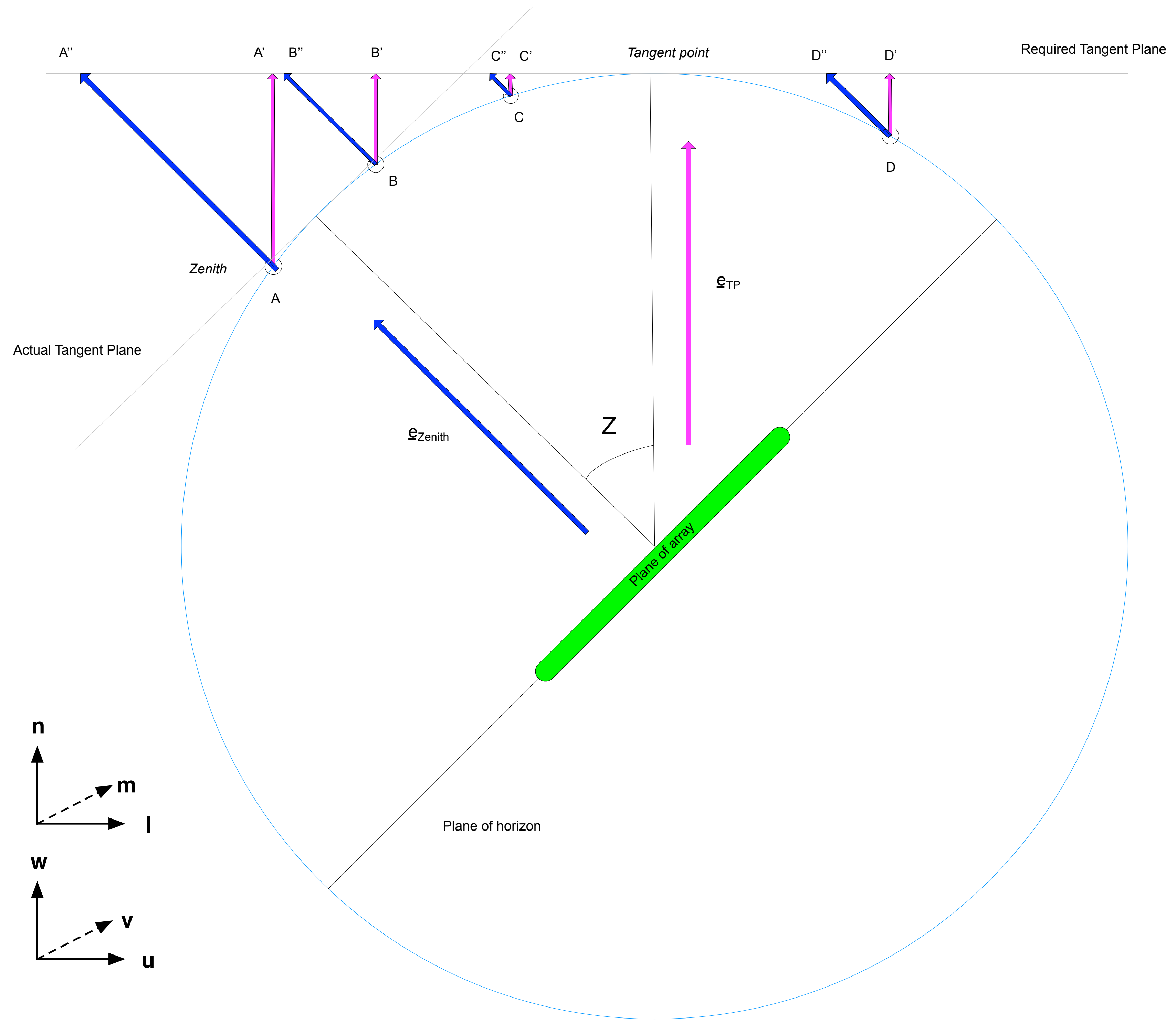
$$w = au + bv$$

$$a = \tan(Z)\sin(\chi)$$

$$b = -\tan(Z)\cos(\chi)$$

$$l \rightarrow l + a\sqrt{1 - l^2 - m^2}$$

$$m \rightarrow m + b\sqrt{1 - l^2 - m^2}$$



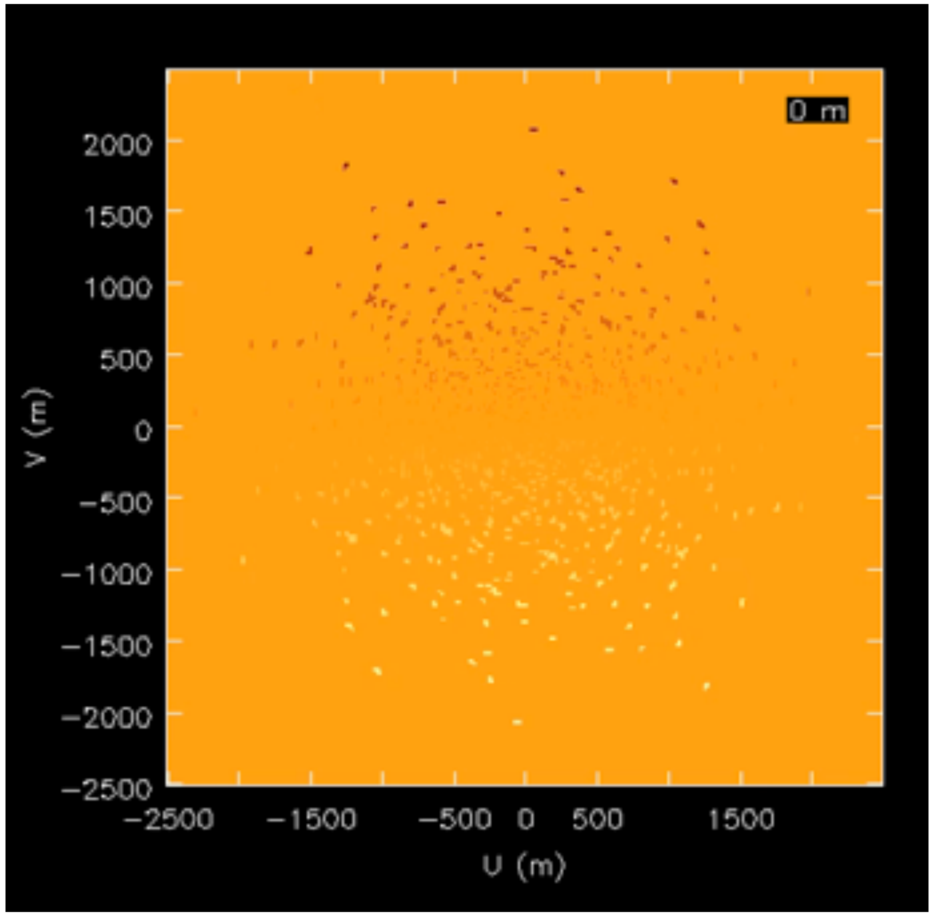
Calim 2011

Snapshot imaging

- Instantaneously arrays are mostly coplanar

$$V_{A'B} = \int I(l,m) e^{-2\pi j\left(ul+vm+w\sqrt{1-l^2-m^2}\right)} \frac{dldm}{\sqrt{1-l^2-m^2}}$$

$$V_{A'B} = \int I(l,m) e^{-2\pi j\left(u\left(l-a\sqrt{1-l^2-m^2}\right)+v\left(m-b\sqrt{1-l^2-m^2}\right)\right)} \frac{dldm}{\sqrt{1-l^2-m^2}}$$



w as a function of u,v and hour angle +/- 4 hours

- Grid on two dimensional plane, FFT, correct for coordinate distortion
- Gridding cost much lower, but need to correct image every integration
- Combine snapshot imaging with AWProjection
  - Fit plane to instantaneous u,v,w sampling
  - Use AWProjection to project down onto fitted u,v plane
  - Set threshold in w *e.g.* 30% of maximum w
  - Refit plane when error in fit exceeds that threshold

$$w_{max} \sim \sqrt[3]{\frac{t_{1snapshot}}{t_{1point}} \frac{N_{pixel}}{N_{vis}} \frac{1}{\Theta_{Res}\Theta^2_{FOV}}}$$

Changes in imaging during scaling work

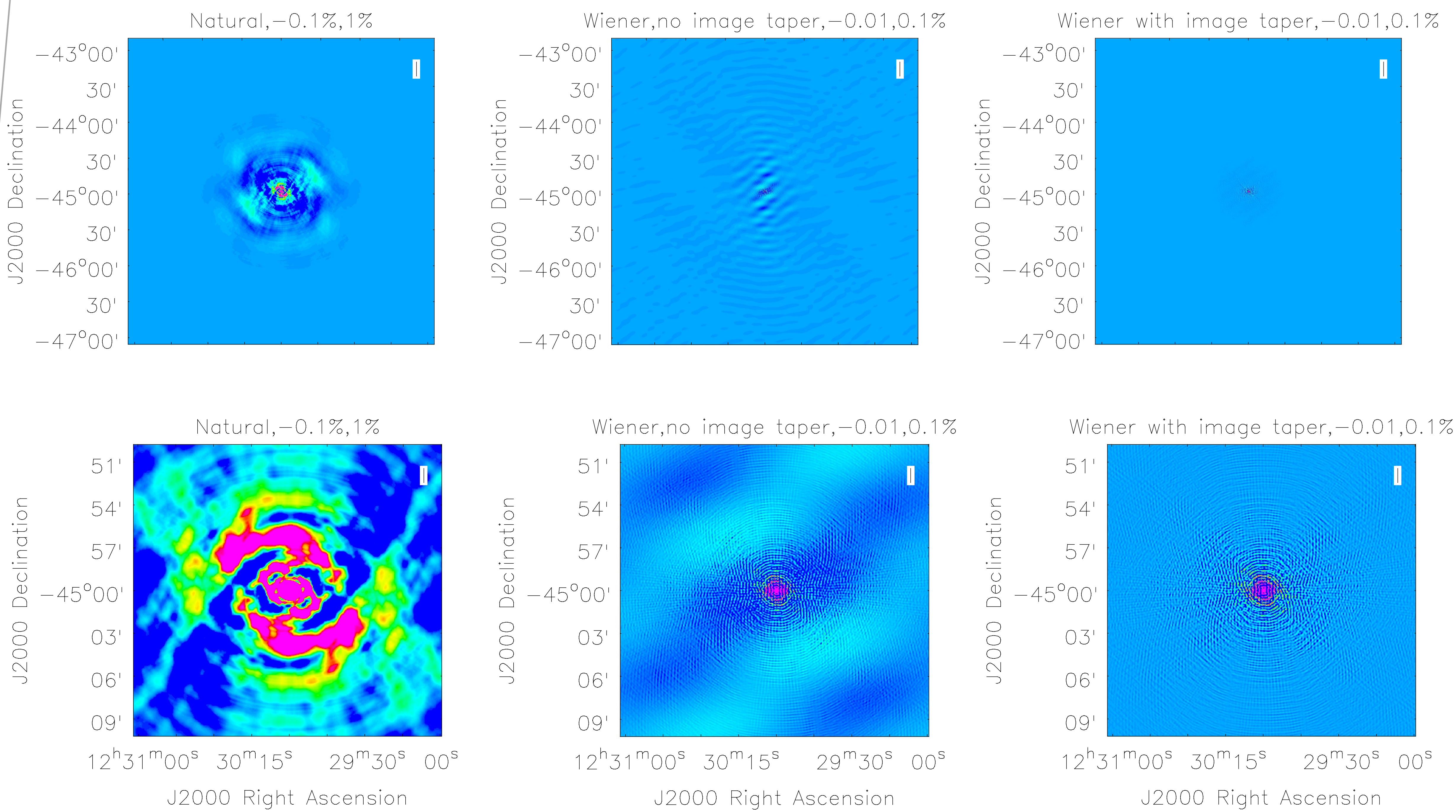
- **AWProject** (2007)
  - W projection + A projection (for primary beam)
  - Too much CPU
  - Too much memory for convolution function
- **AProjectWStack** (2008)
  - Apply W term in image space
  - Much less CPU
  - Too much memory for w stack
- **AWProject + trimmed convolution function** (2009)
  - Only apply and keep non-zero part of convolution function
  - Still too much memory for convolution function
- **AWProject + trimmed convolution function + multiple snapshot planes** (2011)
  - Fit and remove w=au+bv plane every 30 - 60 min
  - Small memory for convolution function
- **Serialise normal equations piece-by-piece for MPI** (2011)
  - Cuts down short bump in memory use

- No current algorithm will scale as-is to full field longer baselines

Preconditioning

- Want to avoid pass through data just for reweighting
- Replaces visibility weighting as a way to control sidelobes
- Calculate dirty image, point spread function with natural weighting
- Calculate Wiener filter to minimise sidelobes
  - Similar to robust weighting
- Can be done after imaging with multiple different parameters
  - Avoids expensive communication pattern

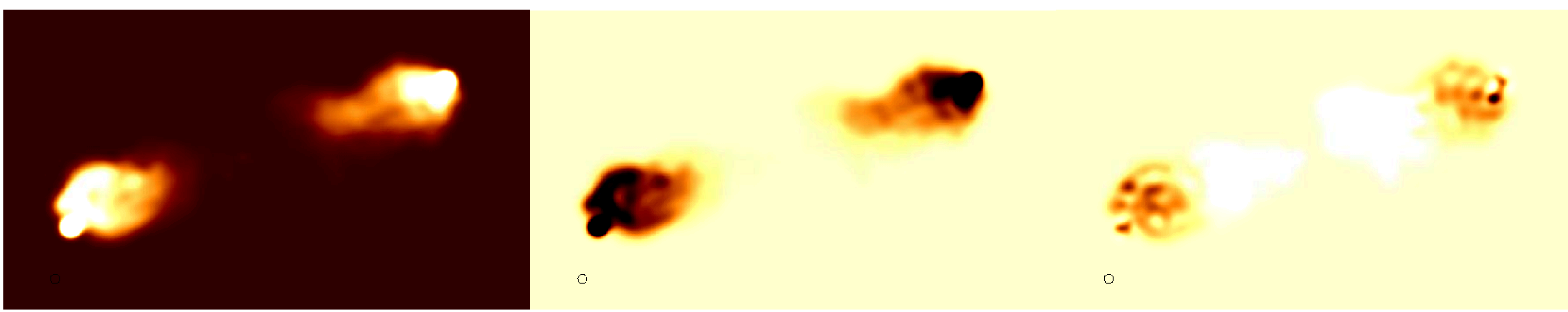
Improvement in preconditioning



Calim 2011

Multiscale Multifrequency Synthesis

- Urvashi's algorithm now implemented in ASKAPsoft
  - Independent of casapy version
- Tested extensively for Urvashi's PhD
- Now in press
- Demonstration here on EVLA simulated data 1.4 - 5.4 GHz
- Parallelises easily
  - Master knows how to do deconvolution minor cycle
  - Workers know how to calculate appropriate normal equations
- But memory use is too large
  - Distribute across multiple MPI processes
  - Investigating Compressive Sampling algorithm



Moment 0 Moment 1 Moment 2

Calim 2011

Summary of imaging capabilities

- Highly parallel code
  - Aiming for ~ 9000 cores
  - Necessary for 10TB/hour throughput
- AWProjection + snapshot imaging
  - Wide field imaging with low memory costs
  - AProjection allows frequency dependent primary beams
- Post-gridding preconditioning
  - To avoid multiple passes through the data

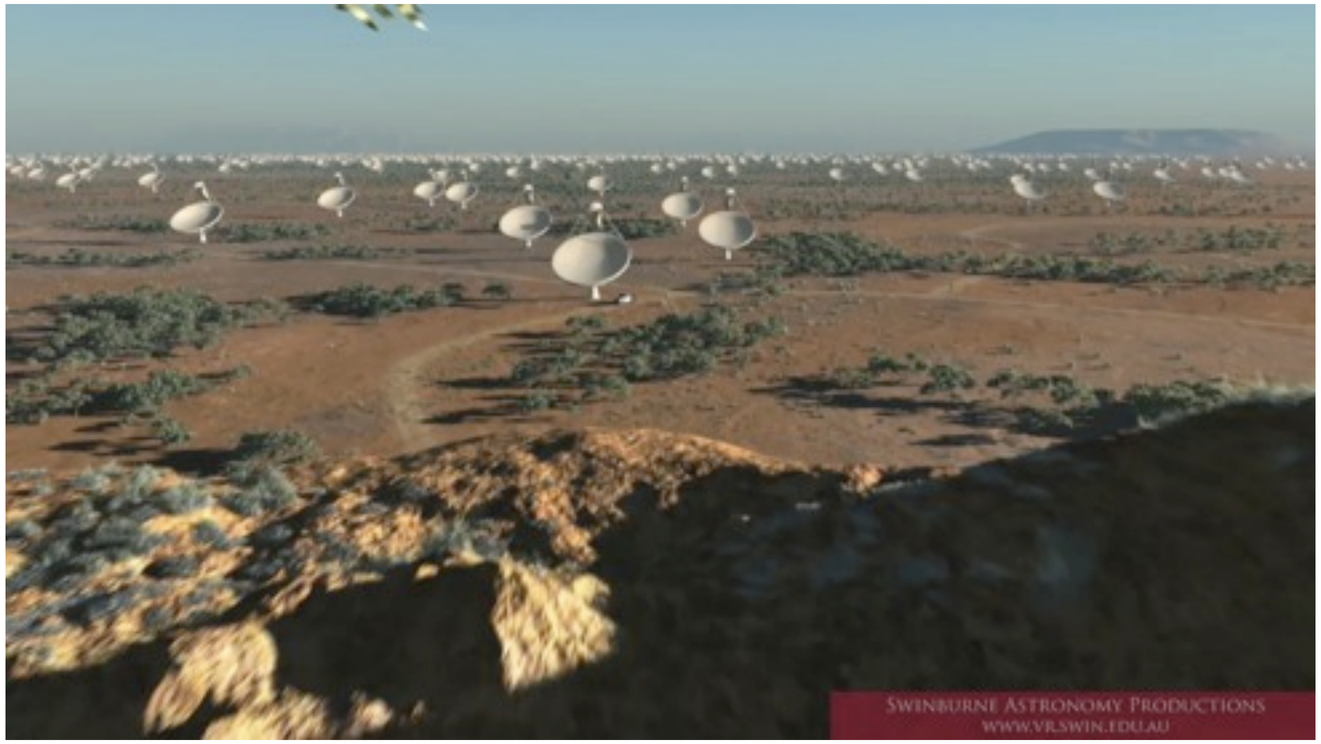
- Urvashi's Multi-Frequency Multi-Scale deconvolution algorithm
  - For wide-band (300MHz) imaging
- SNR-based CLEAN
  - To avoid cleaning low sensitivity regions

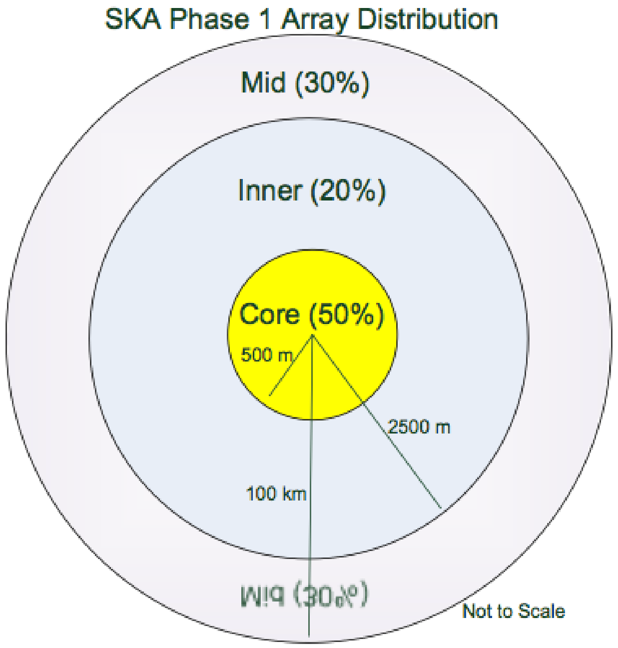
- Designed and optimised for massively parallel processing

- Roughly 7 FTE-years invested

Using ASKAPsoft for SKA1 imaging

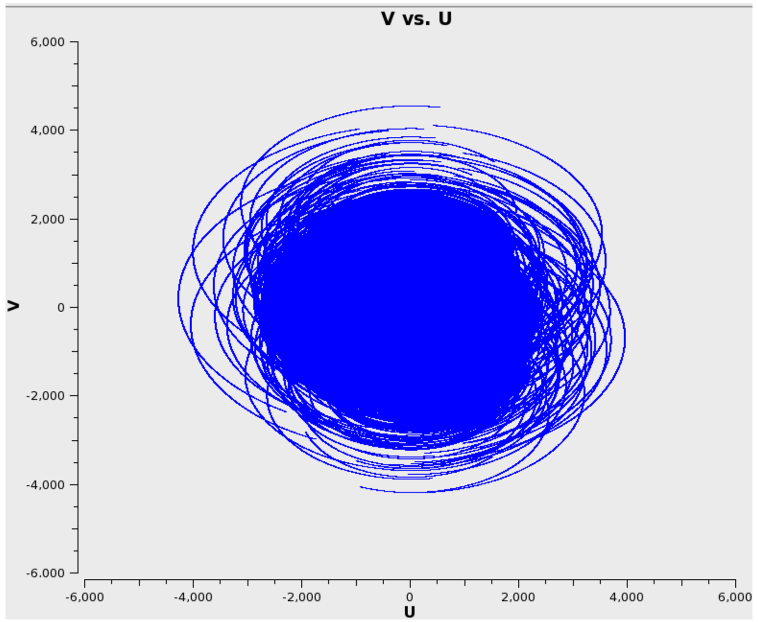
- Dish array
  - Single Pixel Feed
  - Core + inner, 1-2GHz straightforward
    - 170 antennas
    - 8 hour observation = 234 GB
    - 1600 cores
  - Longer baselines still to be tested
- Easy compared to ASKAP!



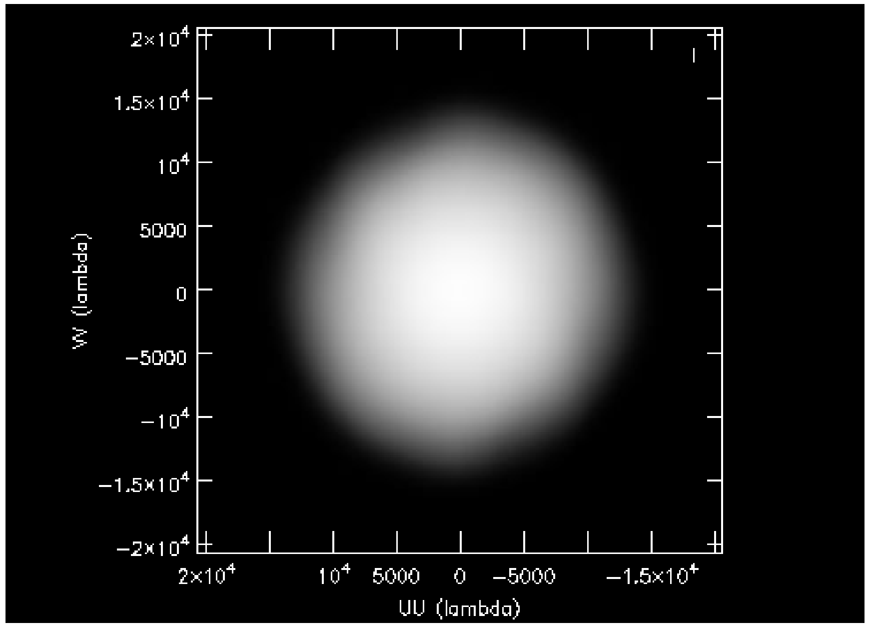


Distribution of antennas

Calim 2011



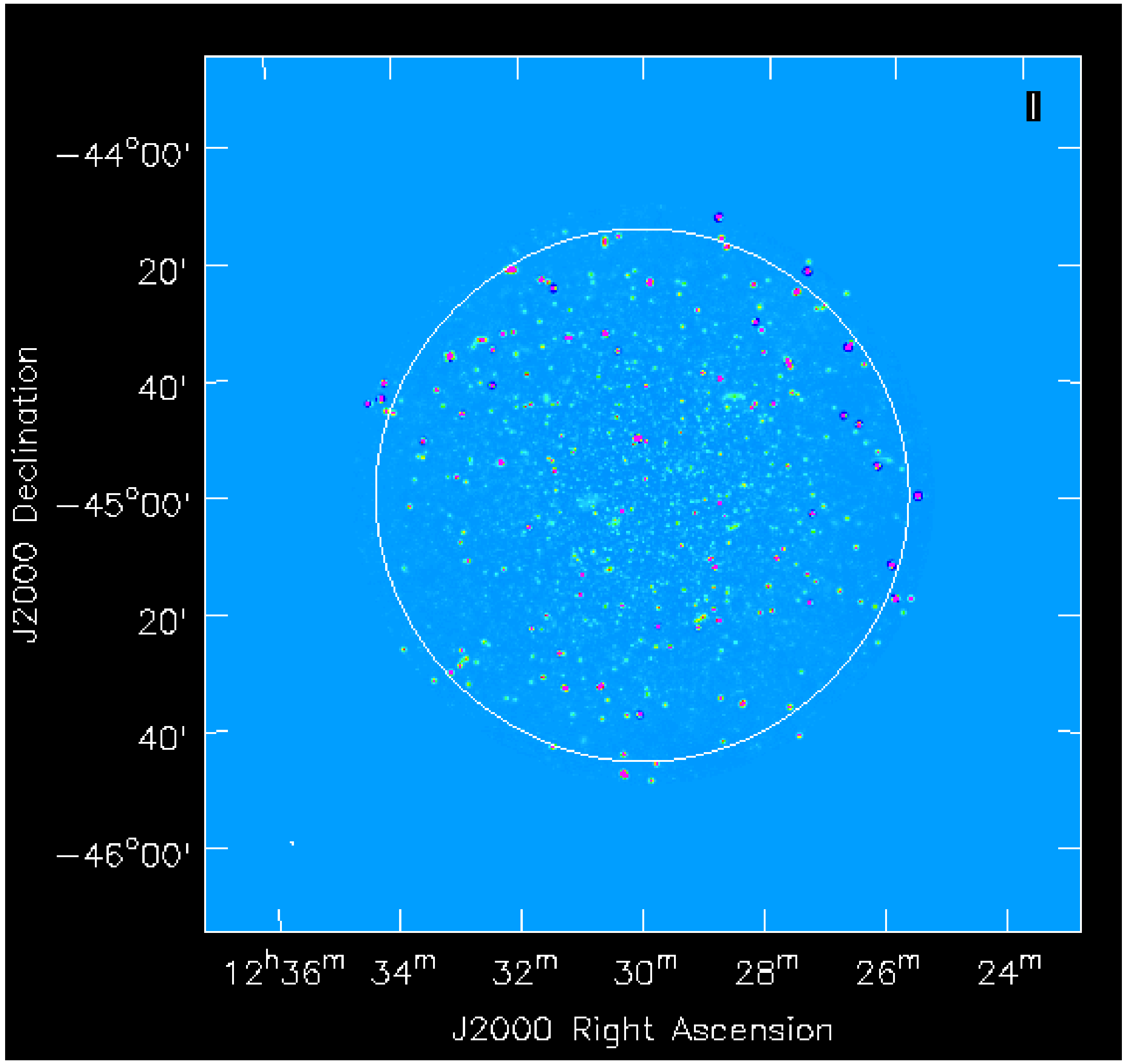
Single frequency uv coverage

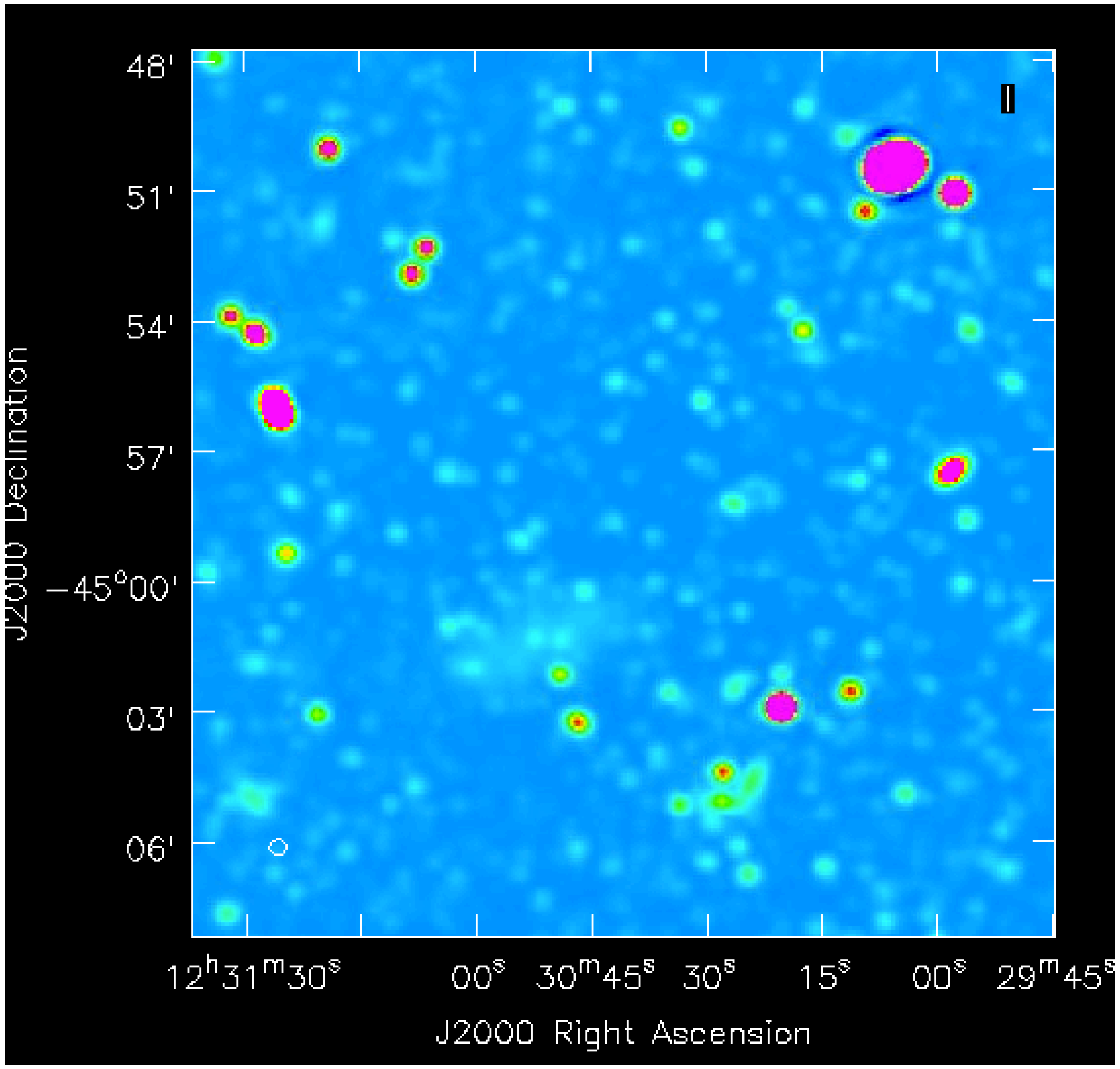


All frequencies uv coverage

Continuum simulations

- 8 hour observation - core and inner
- Confusion limited
- -100uJy to +1000uJy

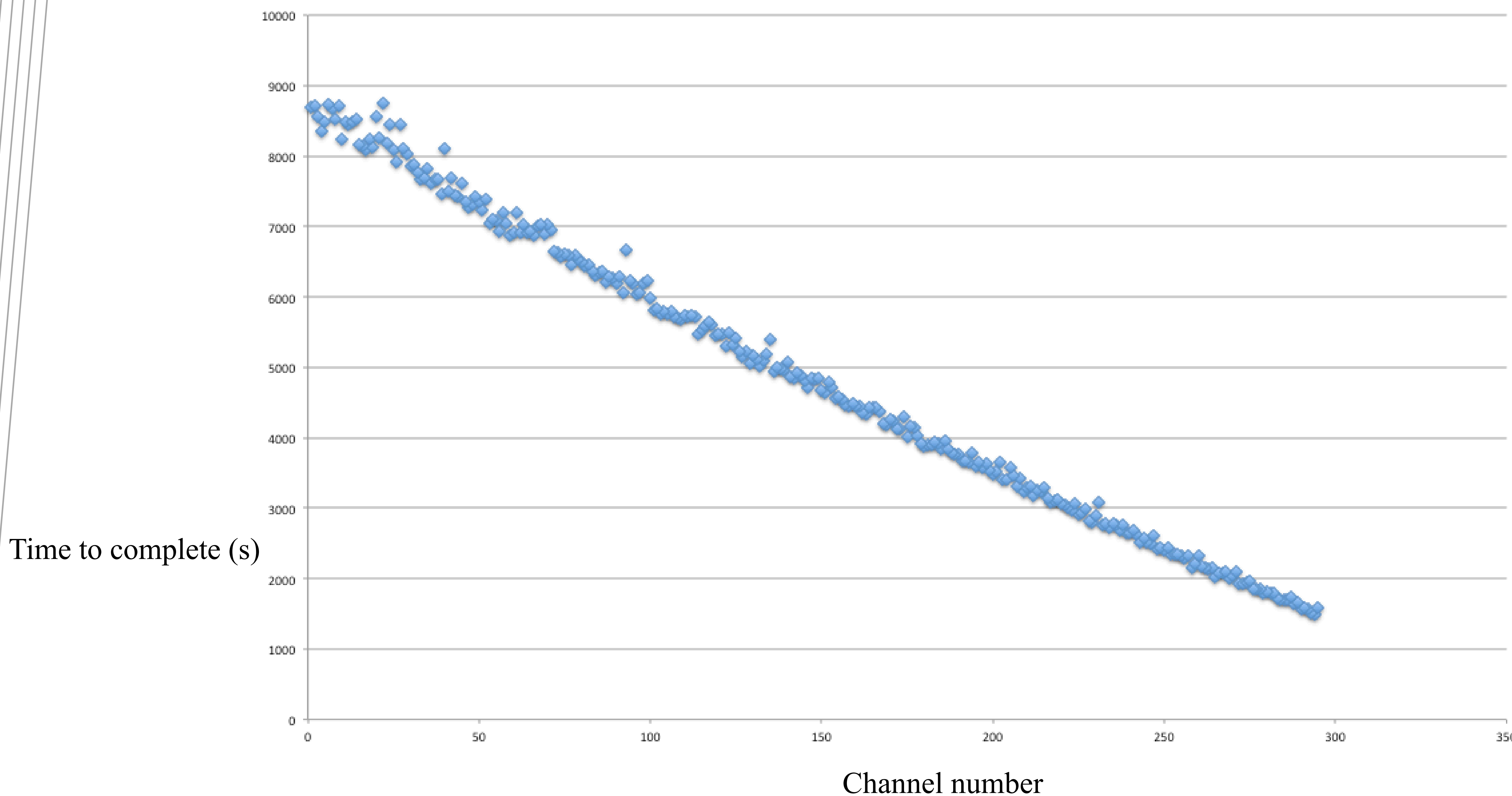




Contour at 1% sensitivity

Calim 2011

Importance of load balancing



Calim 2011

Scaling with baseline length

- Data volume
  - Spectral line ~ B
  - Continuum ~ B^2
- W term ~ B

- Example
  - SKA SPF

Configuration	Data size GB	Major cycle min	Total memory GB	Processes required
1km	88	18	111	7
5km	297	80	115	32
10km	1100	3000	192	240

Calim 2011

Scaling from ASKAP numbers

- 10,000 cores to process
  - 16K channels, 36 antennas, 30 beams, **2km** baselines in real time
- 12,500,000 cores
  - 32K channels, 180 antennas, 30 beams, **10km** baselines in real time

- Roughly 125 PFlops

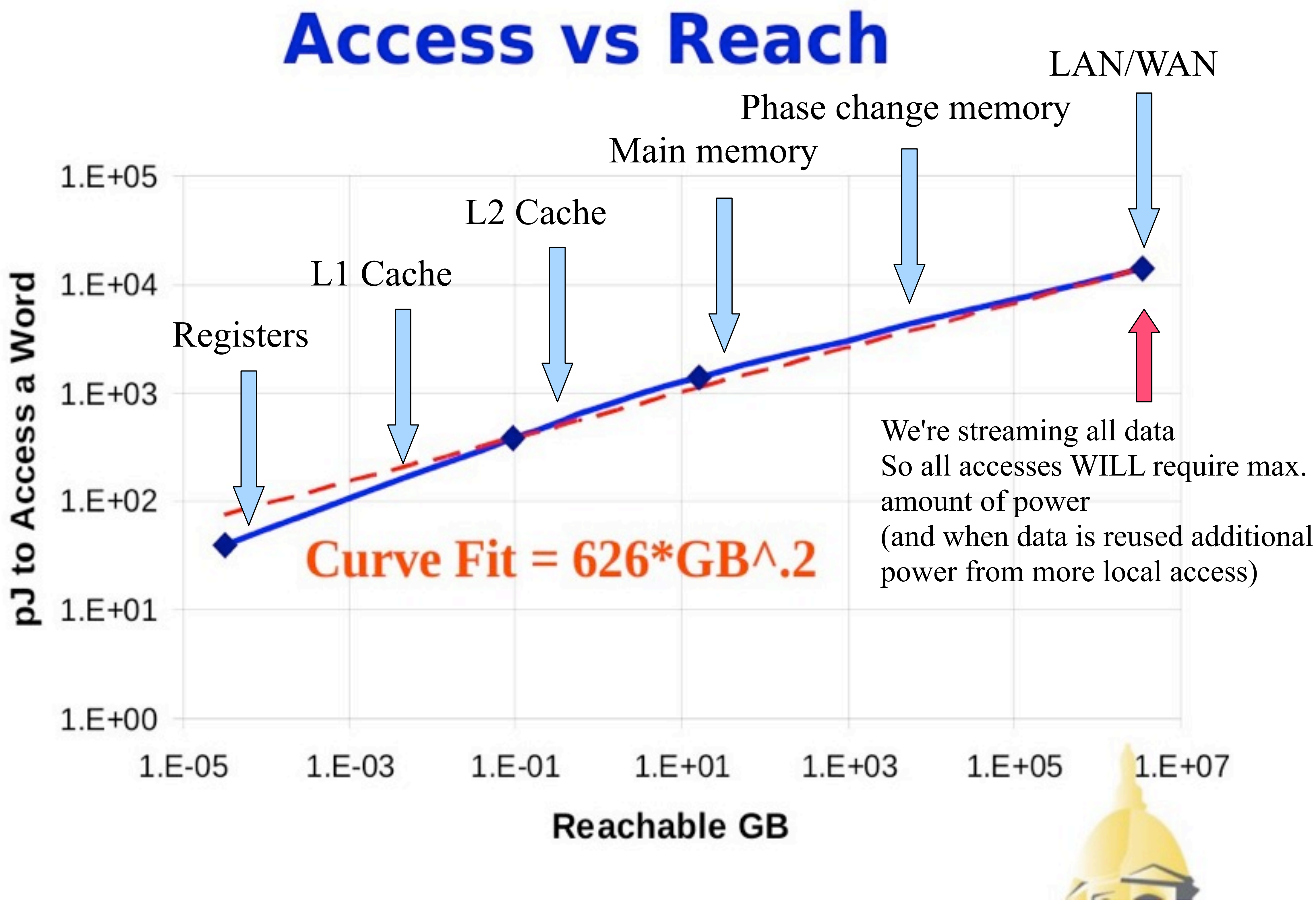
- Cost estimate
  - 1PFlop ~ $10M in 2012
  - 125 PFlops ~ $20M in 2018

- Steepness of scaling curves can be problem and an aid

- Need scientific use cases to firm up these numbers

Exascale=GigaHz KiloCore MegaNode

Systems	2009	2018	Difference Today & 2018
System peak	**2 Pflop/s**	**1 Eflop/s**	**O(1000)**
Power	**6 MW**	**~20 MW**	
System memory	0.3 PB	32 - 64 PB [**.03 Bytes/Flop**]	O(100)
Node performance	125 GF	1,2 or 15TF	O(10) – O(100)
Node memory BW	25 GB/s	2 - 4TB/s [**.002 Bytes/Flop**]	O(100)
Node concurrency	12	O(1k) or 10k	O(100) – O(1000)
Total Node Interconnect BW	3.5 GB/s	200-400GB/s (1:4 or 1:8 from memory BW)	O(100)
System size (nodes)	18,700	O(100,000) or O(1M)	O(10) – O(100)
Total concurrency	225,000	O(billion) [O(10) to O(100) for latency hiding]	O(10,000)
Storage	15 PB	500-1000 PB (>10x system memory is min)	O(10) – O(100)
IO	0.2 TB	60 TB/s (how long to drain the machine)	O(100)
MTTI	days	O(1 day)	- O(10)



Source: Energy at ExaFlops, Peter M. Kogge, SC09 Exa Panel

How to get to 100km?

- Image size ~ 30k by 30k by number of channels

- Will need to go massively multi-threaded (~1000)
- Inter MPI process communication likely to be more important
- More explicit memory management
- Existing third party software will be poorly matched
  - e.g. casacore object copies
- Move away from MeasurementSet

- Between 2012 and 2018 have to expect ~ 5 - 6 substantial changes in algorithm or implementation

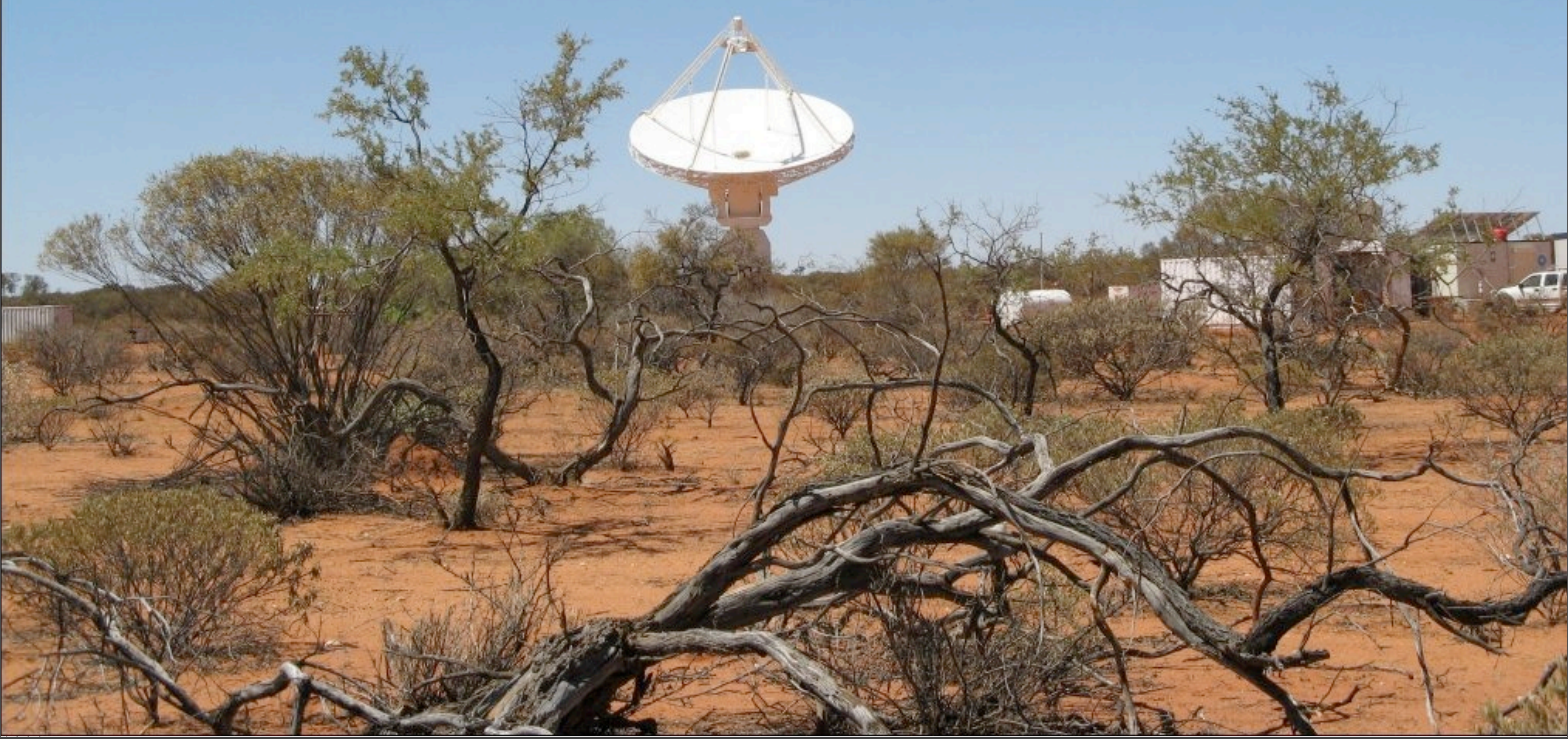
- Incremental development probably most effective
- Goal is highly tuned and specialised software rather than general purpose package

Summary

- Scaling to large numbers of core is slow process
- Roughly 1 substantial change in algorithm/implementation per year
- Resulting code is part of a facility rather than a general purpose package

CSIRO

We acknowledge the Wajarri Yamatji people as the traditional owners of the Observatory site.



ATNF/ASKAP
Tim Cornwell
ASKAP Computing Project Lead

Phone: +61 2 9372 4261
Email: tim.cornwell@csiro.au
Web: www.atnf.csiro.au

www.csiro.au

Thank you

Contact Us
Phone: 1300 363 400 or +61 3 9545 2176
Email: enquiries@csiro.au Web: www.csiro.au

CSIRO