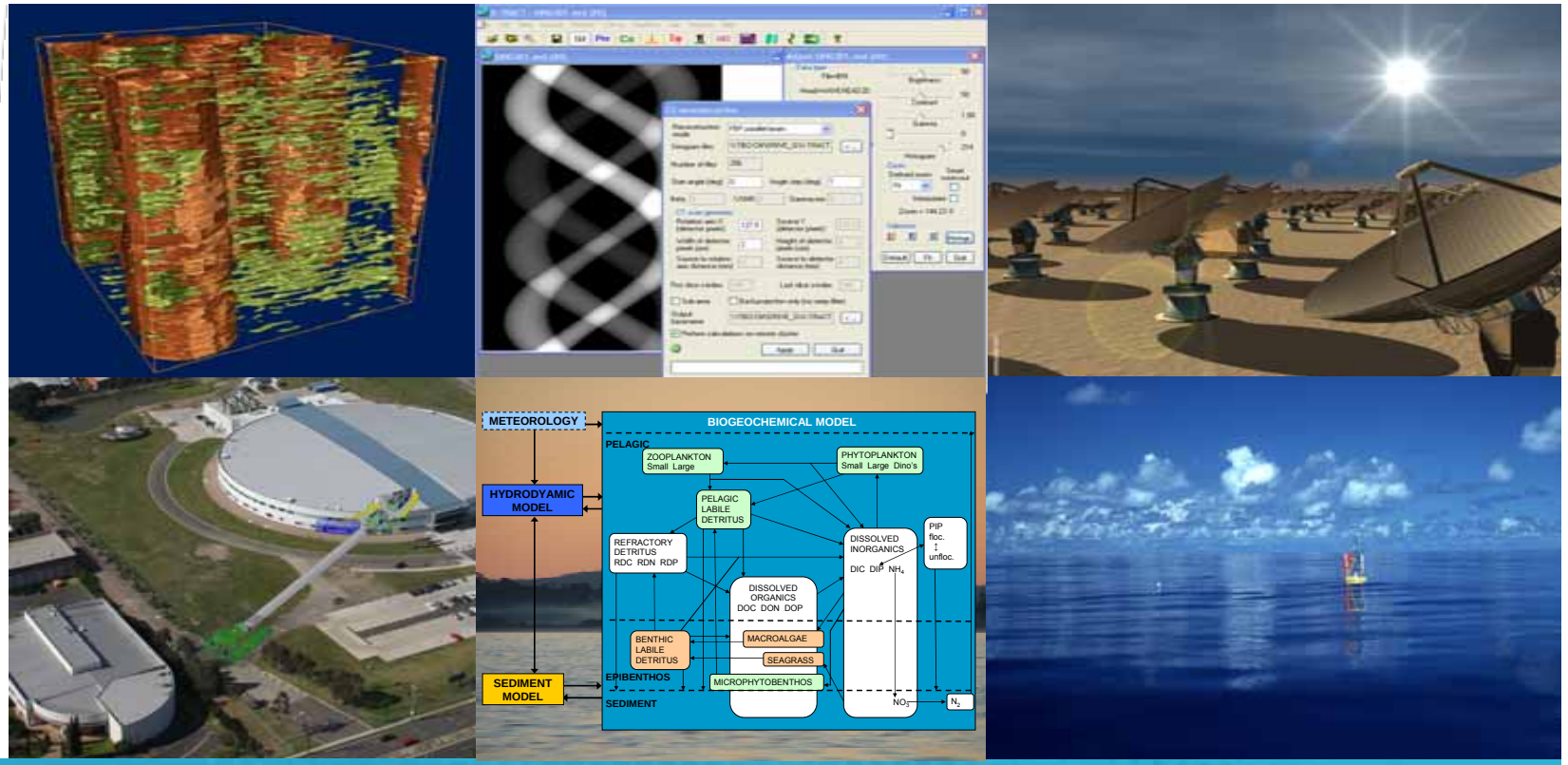


www.csiro.au

CSIRO Computational and Simulation Sciences

Data + Algorithms + Visualisation
= Scientific Discovery

Dr John A Taylor
Science and Business Leader
CSIRO Computational and Simulation Sciences

Presentation to NVIDIA GPU Computing Seminar January 2010

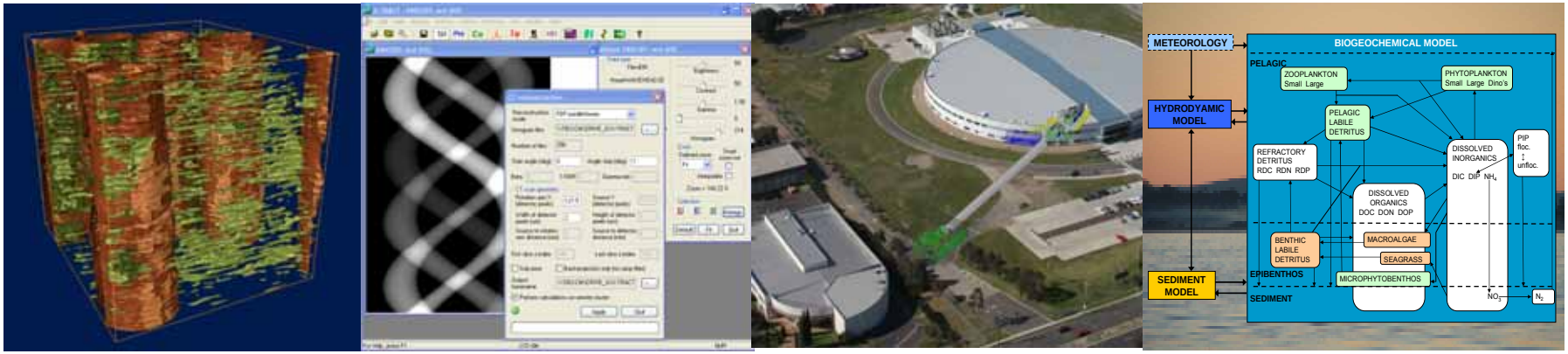
CSIRO

Summary of this presentation

- **Context**
  - An overview of CSIRO
  - CSIRO Computational and Simulation Science
  - CSIRO GPU Cluster
- **Applications**
  - Solving ODEs on GPUs
  - Imaging
  - High performance CT Reconstruction
  - Square Kilometre Array Radio Telescope

CSIRO

- An overview



CSIRO today: a snapshot

Australia's national science agency

One of the largest & most diverse in the world

6500+ staff over 55 locations

Ranked in top 1% in 14 research fields

20+ spin-off companies in six years

160+ active licences of CSIRO innovation

Building national prosperity and wellbeing

Revenue

89% of funding is directed to Australia's National Research Priorities

CSIRO in 2008-09:

- Total revenue of AUD$1.3 billion
- Federal funding of $668.1 million
- $634.8 million external revenue
    - With $229.6 million from IP revenue

National Research Flagships

Climate Adaptation

Light Metals

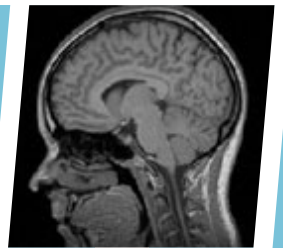
Sustainable Agriculture

Energy Transformed

Minerals Down Under

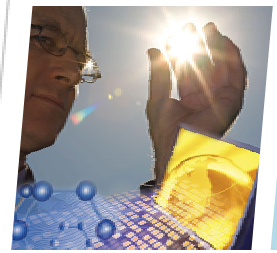
Water for a Healthy Country

Food Futures



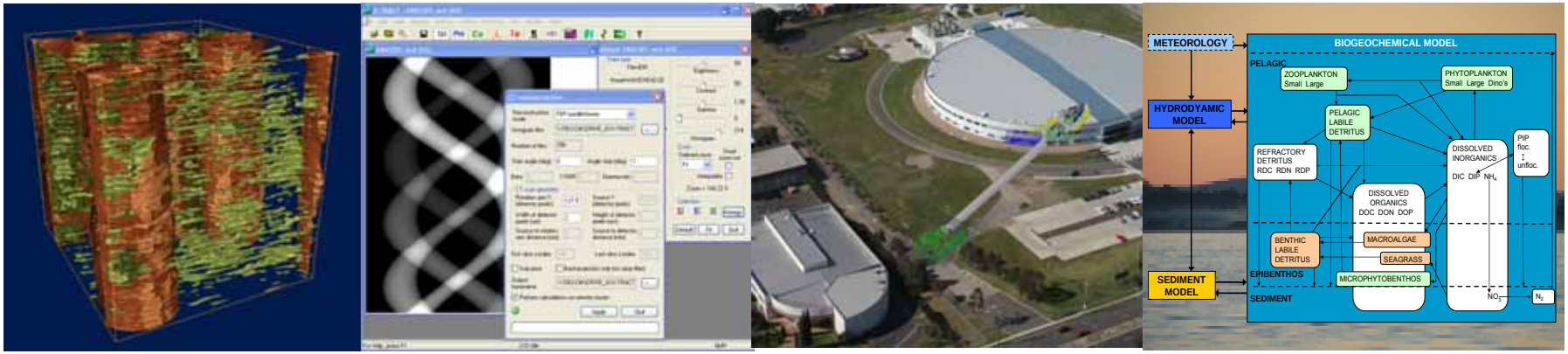
Preventative Health

Wealth from Oceans



Future Manufacturing

CSIRO Computational and Simulation Science (CSS)

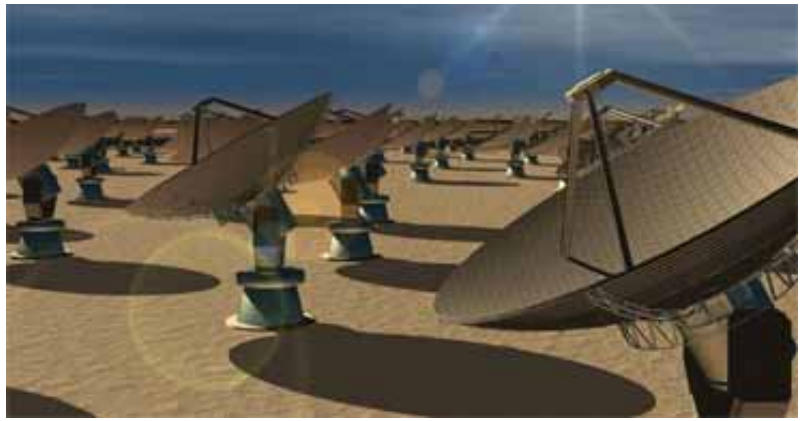


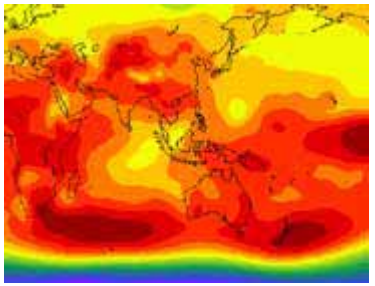
The need for CSS capability

In one week, ASKAP will generate more information than is contained on the entire world wide web

Climate change is here today… and will be tomorrow

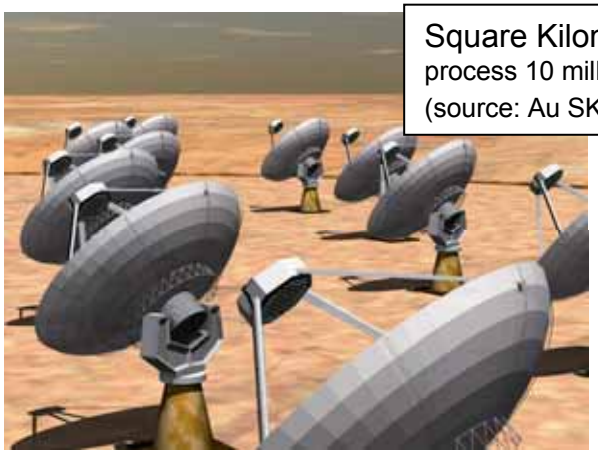
The genomic revolution is dawning



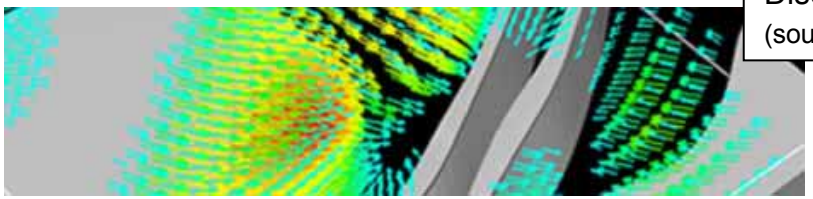


CSS extends our reach beyond the limits of theory and experiment

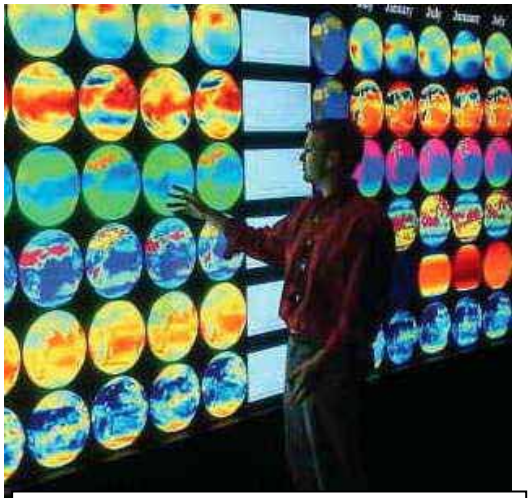
Science: Emerging Science Challenges



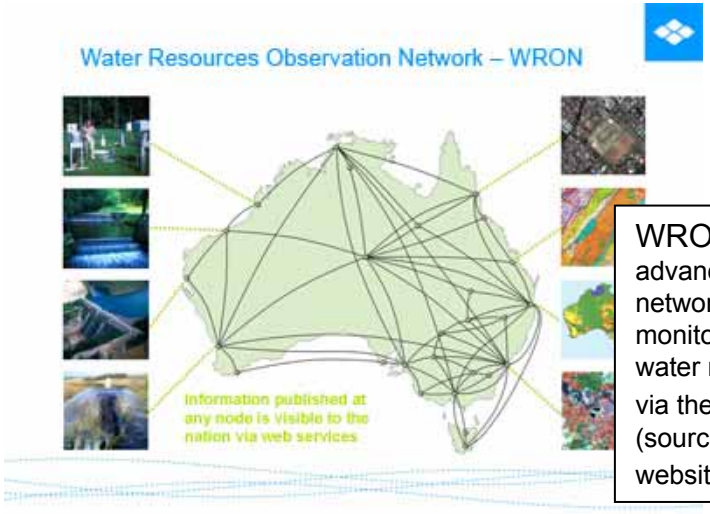
Square Kilometre Array – Potential to process 10 million Gb of data per hour (source: Au SKA website)



Predictive Mineral Discovery – (source:CSIRO E&M)

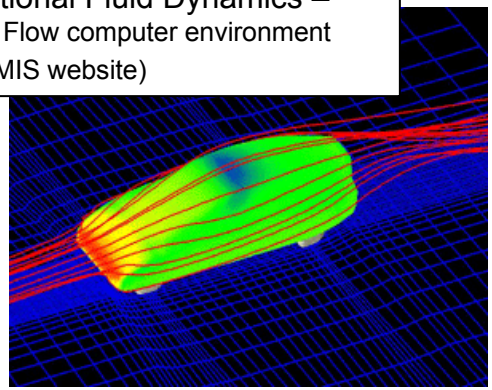


Visualization capabilities – To illuminate high-performance computing data from large-scale climate simulations. (source: US Department of Energy)

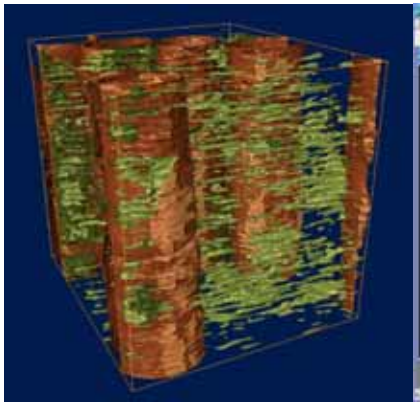


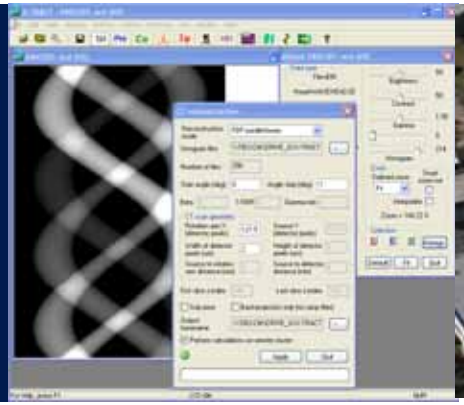
WRON – Using advanced sensor networks to monitor scarce water resources via the internet (source: WRON website)

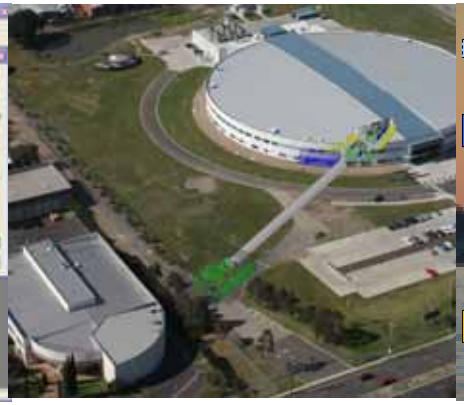
Computational Fluid Dynamics – CMIS Fast Flow computer environment (source: CMIS website)

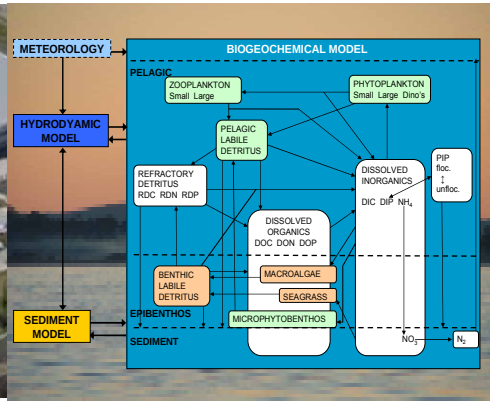


CSIRO GPU Cluster

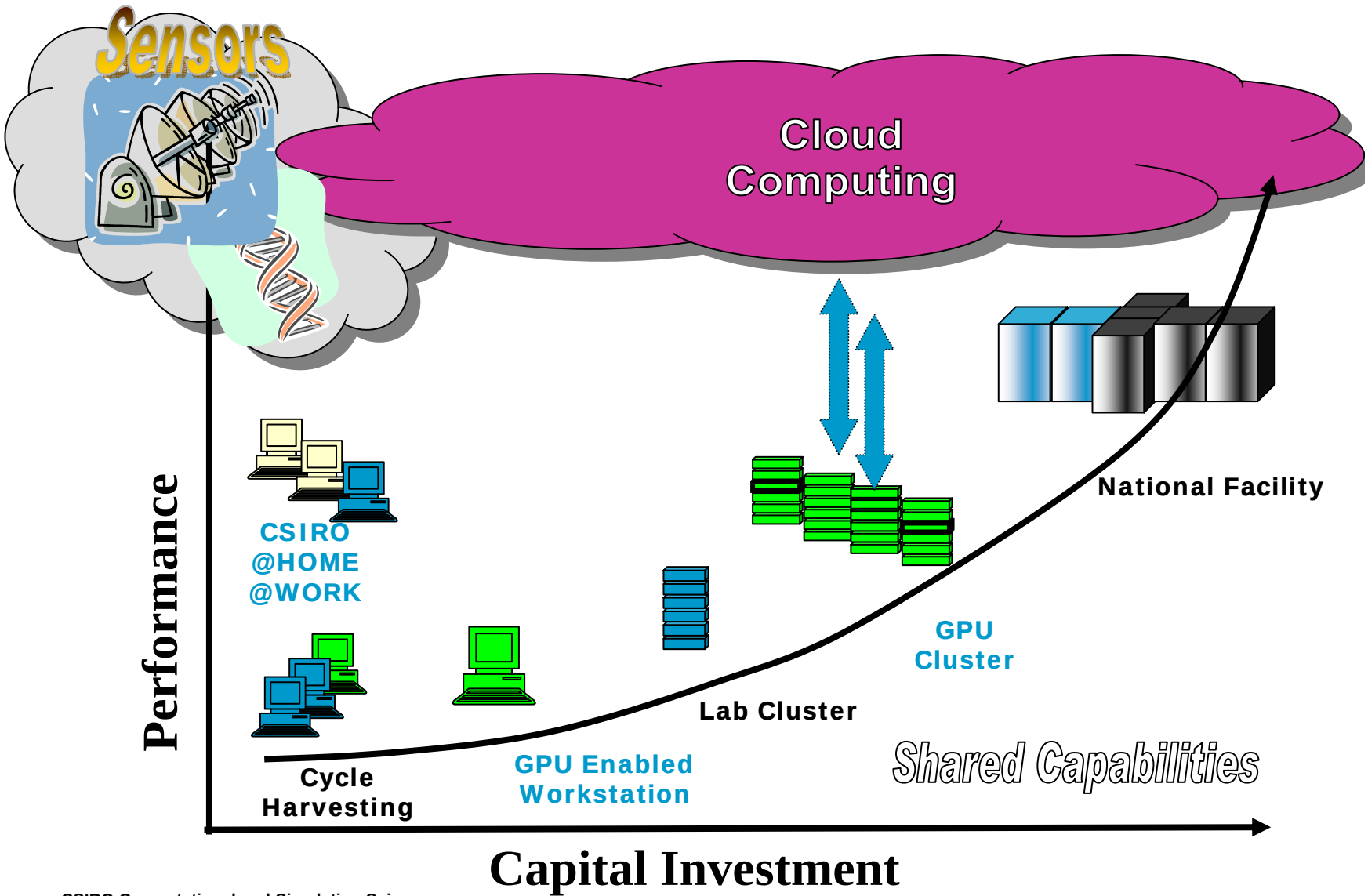




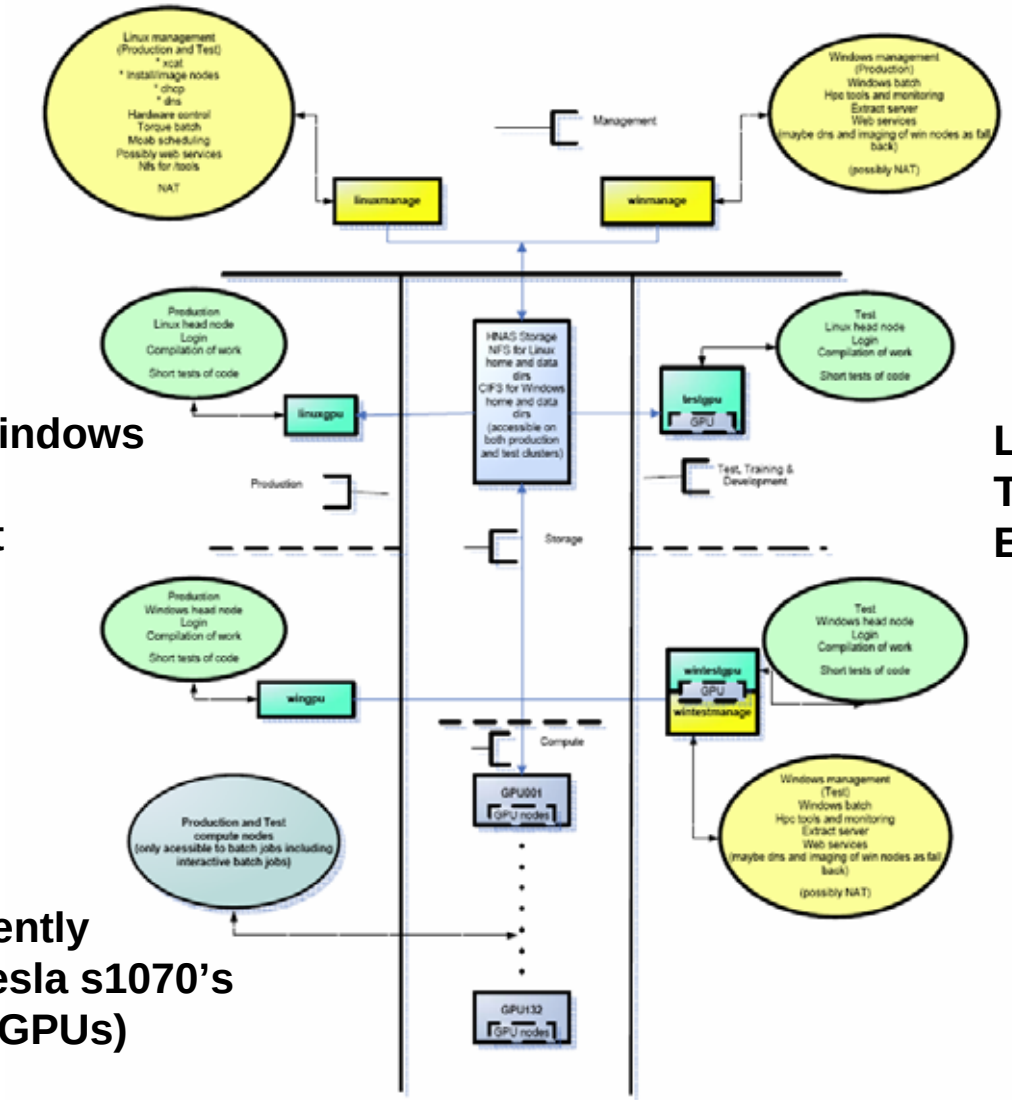




CSS TCP - Desktop to Global Resources



CSIRO GPU CLUSTER



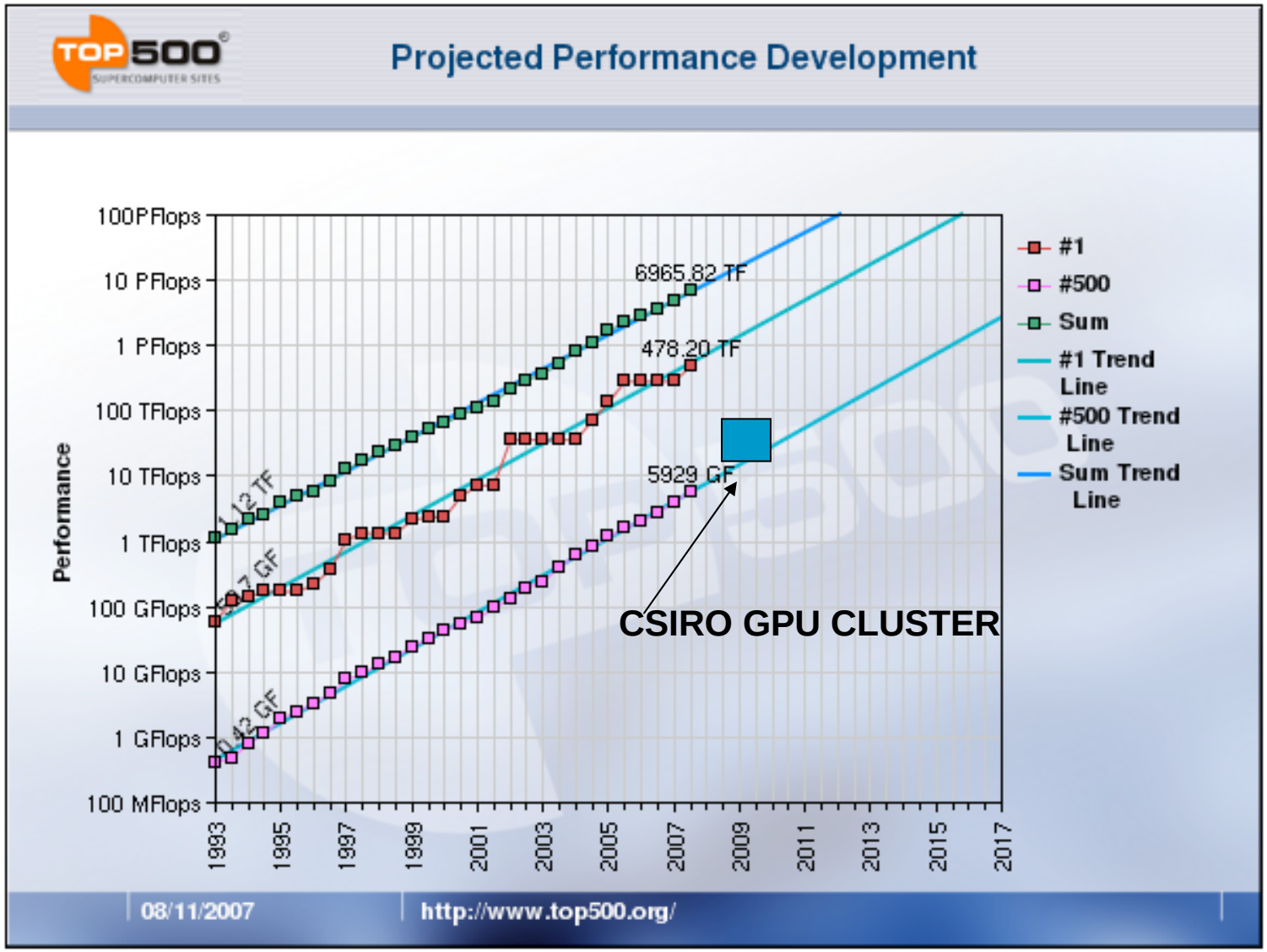
Linux and Windows Management Nodes

Linux and Windows Production Environment

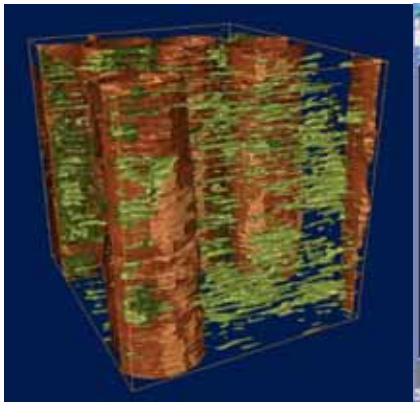
Linux and Windows Test & Training Environment

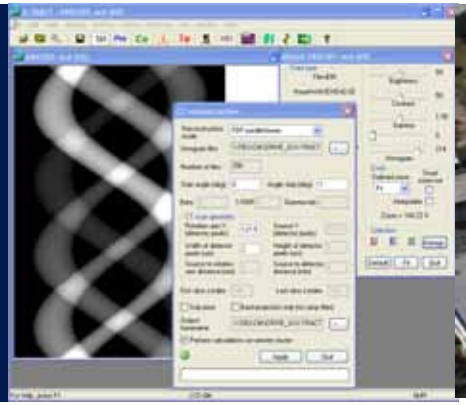
Currently
64 Tesla s1070's
(256 GPUs)

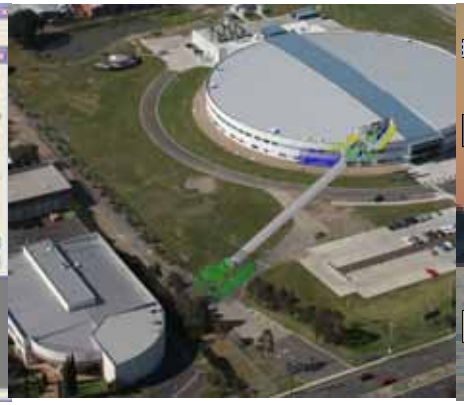
CSIRO GPU Cluster

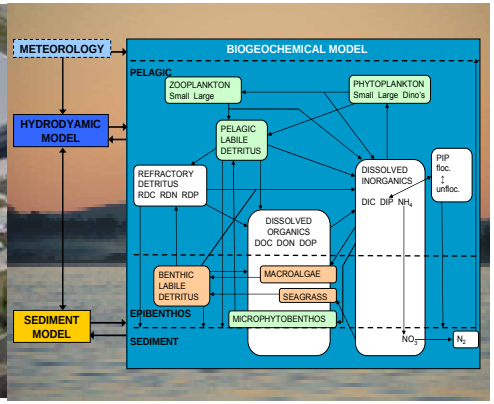


Solving ODEs on GPUs







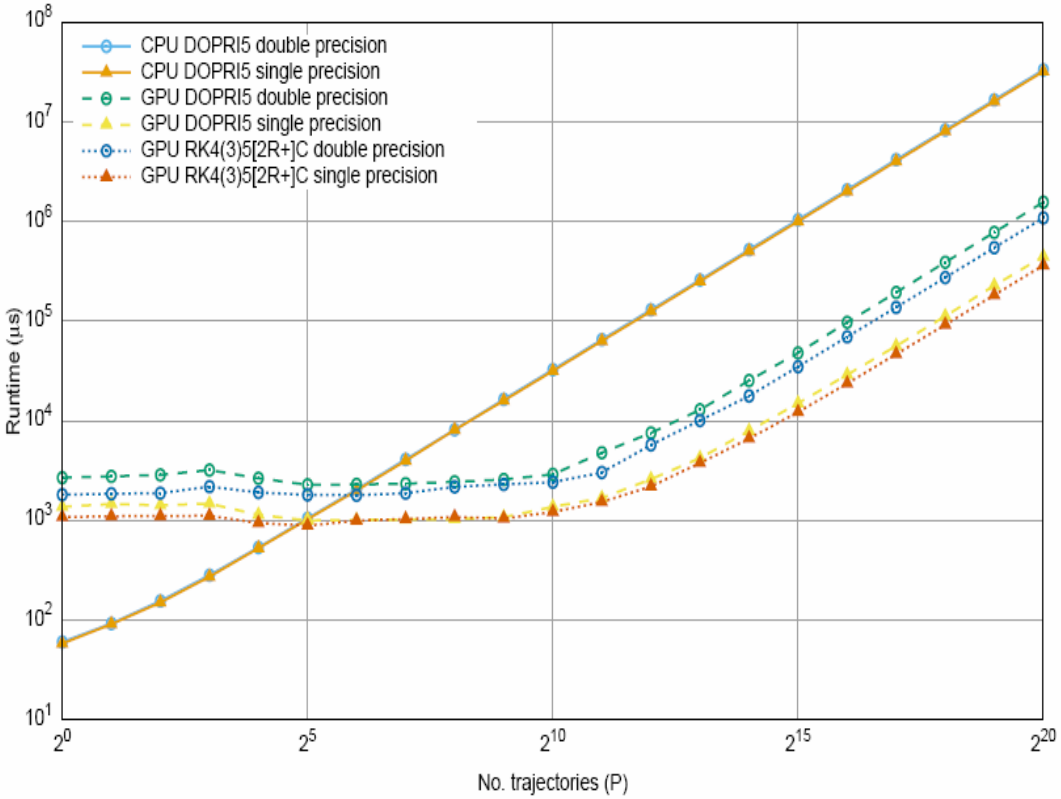


Ordinary Differential Equations

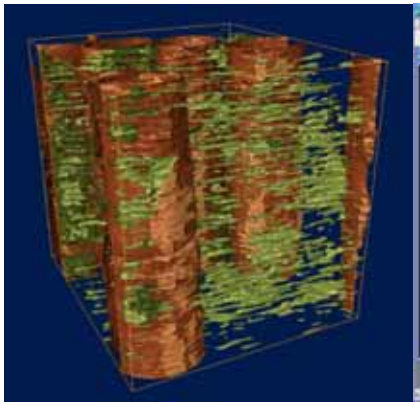
- **Ordinary differential equations (ODEs) are a ubiquitous means of modelling various physical, biological, ecological and other phenomena in the scientific community.**
- **Integration schemes for ODEs work through several data-dependent stages over many steps, and so have relatively large working memory requirements.**
- **Even so, they can be adapted to the GPU by a two-dimensional parallelisation over both variables and trajectories, and suitable use of shared memory.**
- **Additional gains can be made by considering integration schemes with low storage requirements, that minimise register use to maximise the number of concurrently running threads.**

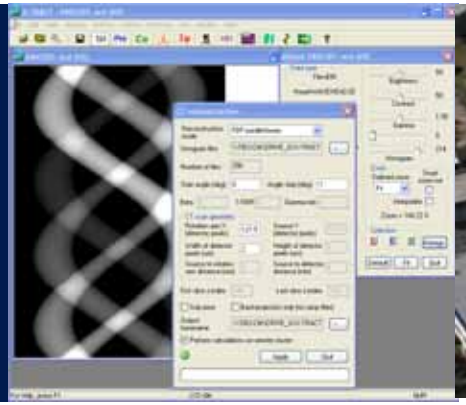
Ordinary Differential Equations

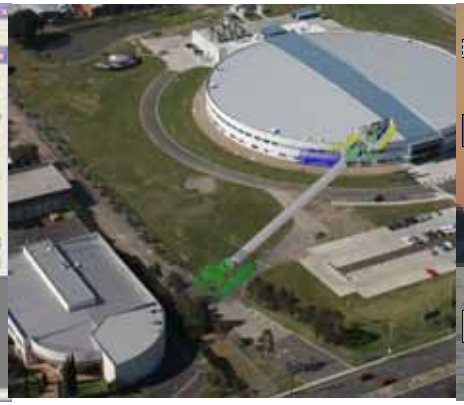
- **DOPRI5 integrator on a GPU performs up to 23 times faster in double and 76 times faster in single precision.**
- **A low-storage method, RK4(3)5[2R+]C does even better -- 33 times faster in double and 93 times faster in single precision.**
- **Note that the gain from this register usage optimisation is particularly evident in double-precision, perhaps of most interest to scientists for these types of problems.**
- **The take home message: minimise working memory requirements at the level of algorithm design.**

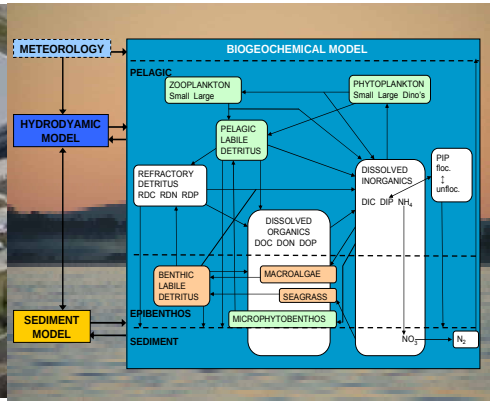


Imaging



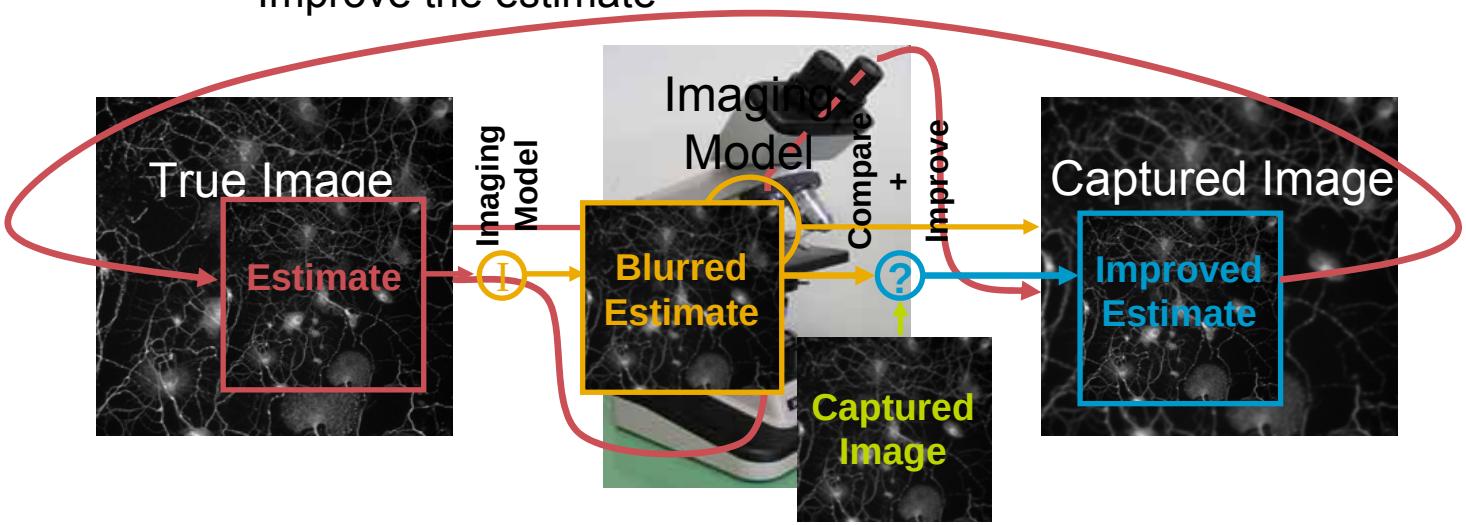





METEOROLOGY
BIOGEOCHEMICAL MODEL
PELAGIC
ZOOPLANKTON
Small Large
PHYTOPLANKTON
Small Large Dino's
HYDRODYAMIC MODEL
PELAGIC LABILE DETRITUS
REFRACTORY DETRITUS
RDC RDN RDP
DISSOLVED INORGANICS
DIC DIP NH_4
PIP
floc.
↕
unfloc.
DISSOLVED ORGANICS
DOC DON DOP
BENTHIC LABILE DETRITUS
MACROALGAE
SEAGRASS
EPIBENTHOS
MICROPHYTOBENTHOS
SEDIMENT MODEL
SEDIMENT
NO_3
N_2

Reversing the blur

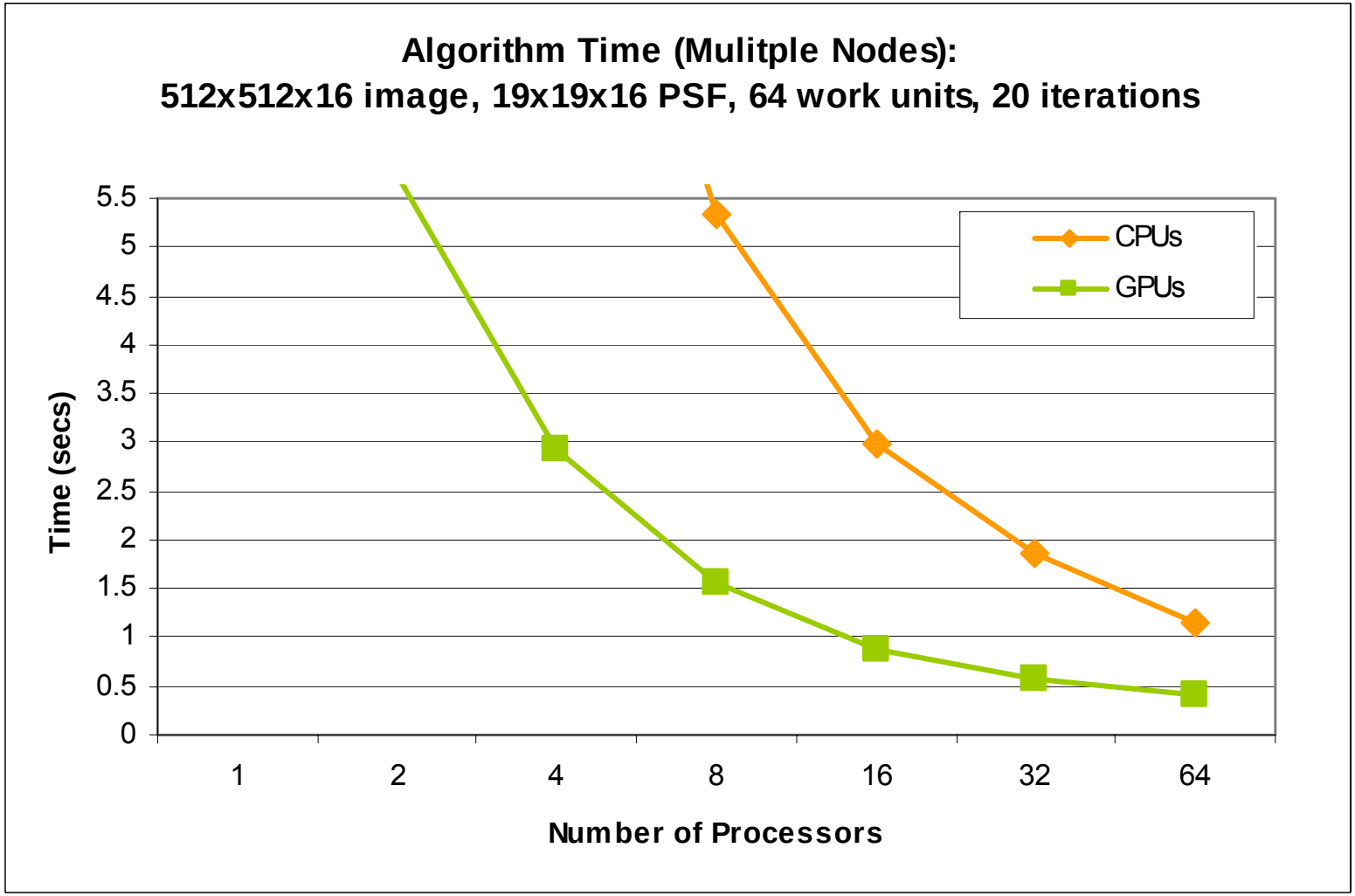
- **Imaging Model**
  - Mathematical function that describes how true image is blurred
- **Deconvolving**
  - Achieved by repetitively:
    - Applying imaging model to an estimate of true image
    - Comparing results to the captured image
    - Improve the estimate



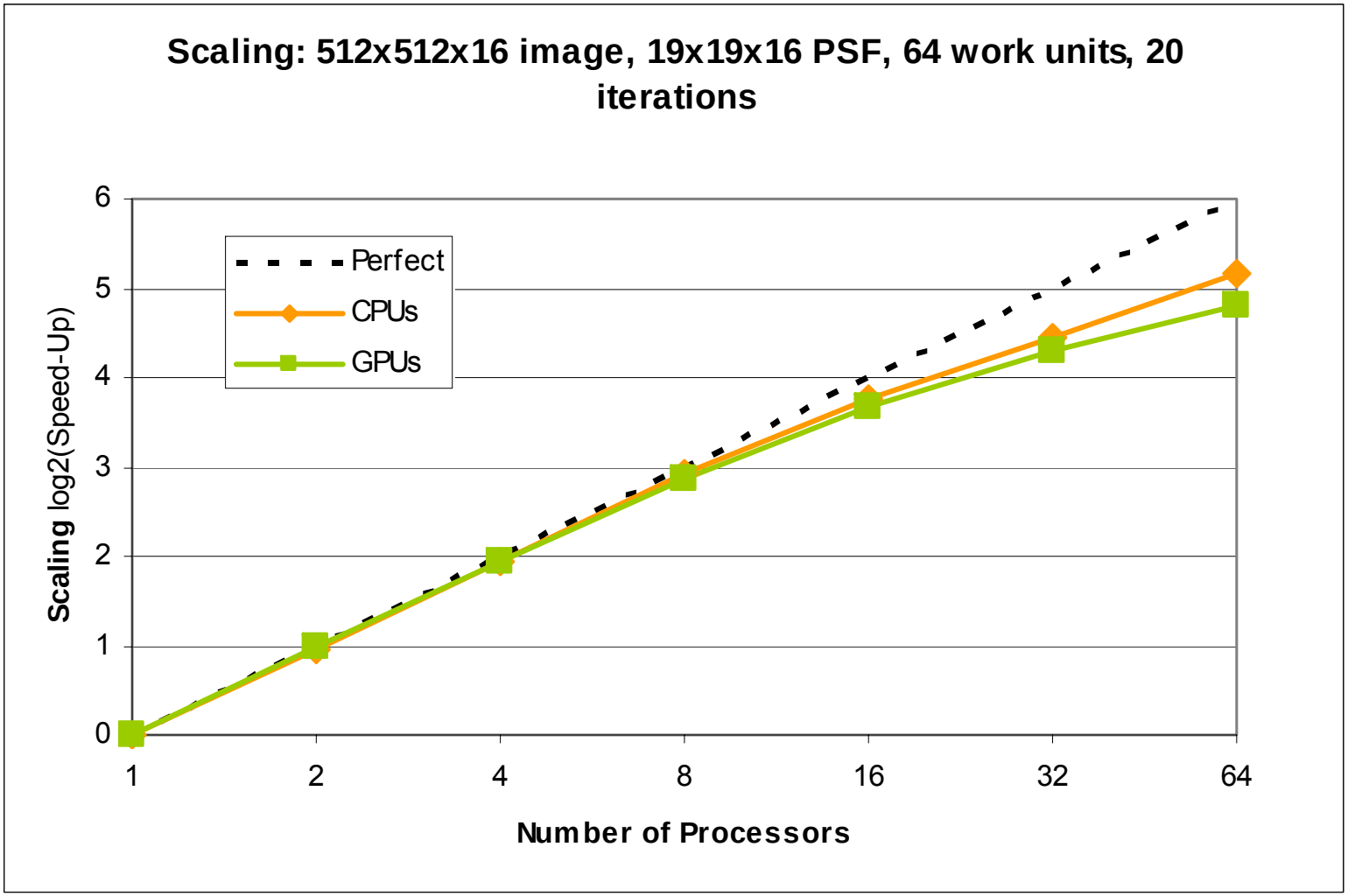
3D Spatially-Variant Image Deconvolution

- **Remove blurring introduced by limitations of imaging device**
  - Diffraction, defocus, motion, environmental factors
- **Increase resolving power and image quality**
  - Examine specimens in greater detail, at smaller scale, with higher accuracy
- **Implementation on heterogeneous compute cluster**
  - Use GPUs and CPUs to squeeze all the processing power from a node
- **Biologist's see microscope images deconvolved on the fly**

Results

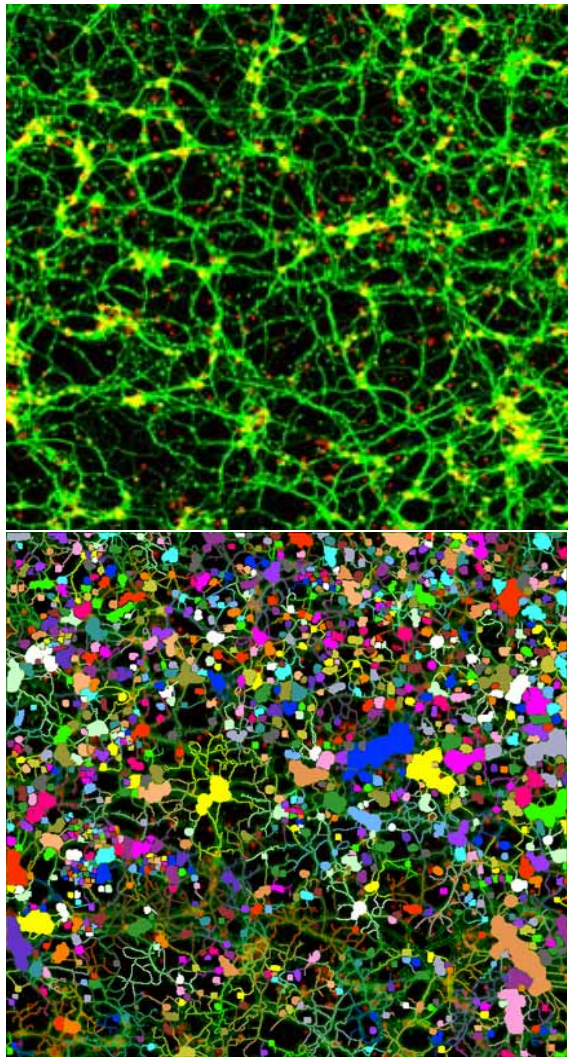

Algorithm Time (Mulitple Nodes):
512x512x16 image, 19x19x16 PSF, 64 work units, 20 iterations
Time (secs)
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
1
2
4
8
16
32
64
Number of Processors
CPUs
GPUs

Results


Scaling: 512x512x16 image, 19x19x16 PSF, 64 work units, 20 iterations
Perfect
CPUs
GPUs
Scaling log2(Speed-Up)
0
1
2
3
4
5
6
1
2
4
8
16
32
64
Number of Processors

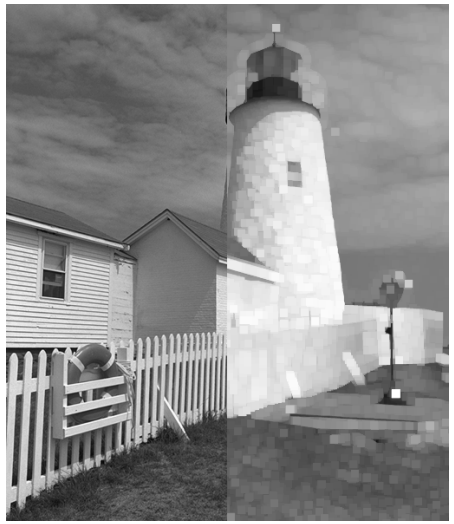
High-Content Analysis and High-Throughput Imaging

- **Applications in drug discovery**
  - Pharmaceutical companies produce thousands of images during automated assays/experiments
  - Image size and complexity (1280 x 1280; 100 – 1000s of object)
  - Millions of parameters to measure
- **Computational Bottlenecks**
  - Manual analysis is near impossible
  - Automated analysis can take several hours
  - Throughput time unacceptable for the workflows in laboratories
- **Solution**
  - Utilise GPU to reduce processing times from hours to minutes
  - Focus on both high-level and low-level algorithms



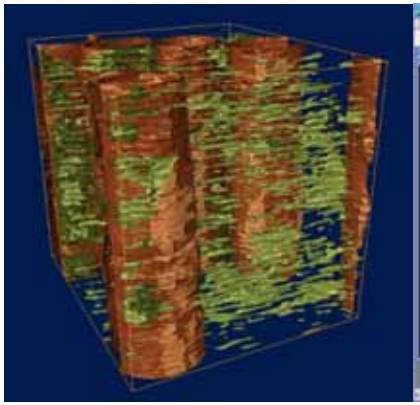
Other activities

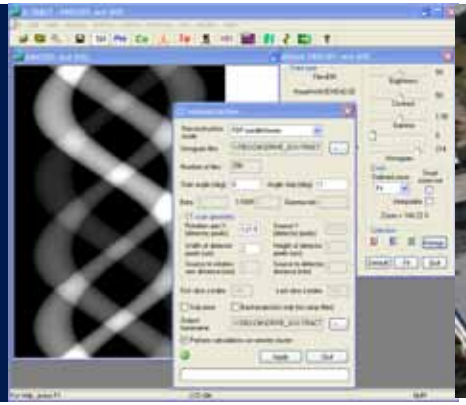
- **Low-level image analysis operations**
  - Highly optimised, low level image operations
  - Well known techniques used commonly in image analysis
  - Need to implement MANY different operations to build up a useful pipeline of GPU accelerated functions

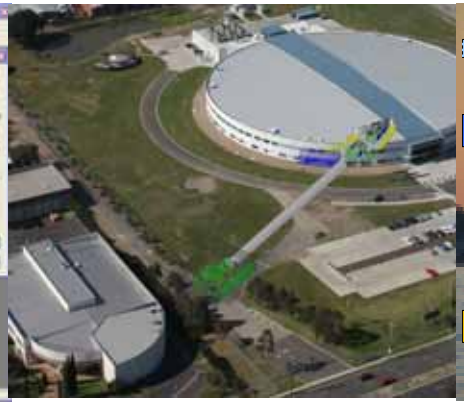


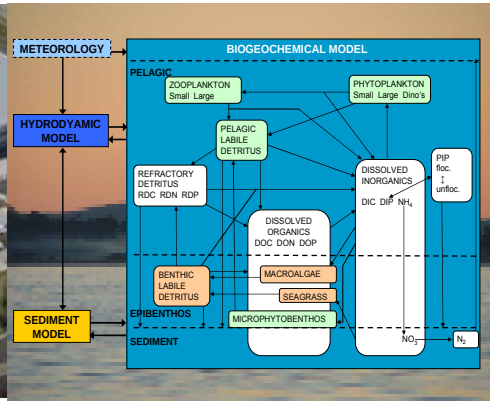
- **GPU accelerated statistics with R**
  - Use publicly available GPU libraries to accelerate R
  - BLAS and LAPACK are important libraries used heavily by R
  - Reach a wide array of application areas and problems by enhancing a generic language/tool

High-performance CT reconstruction using GPUs









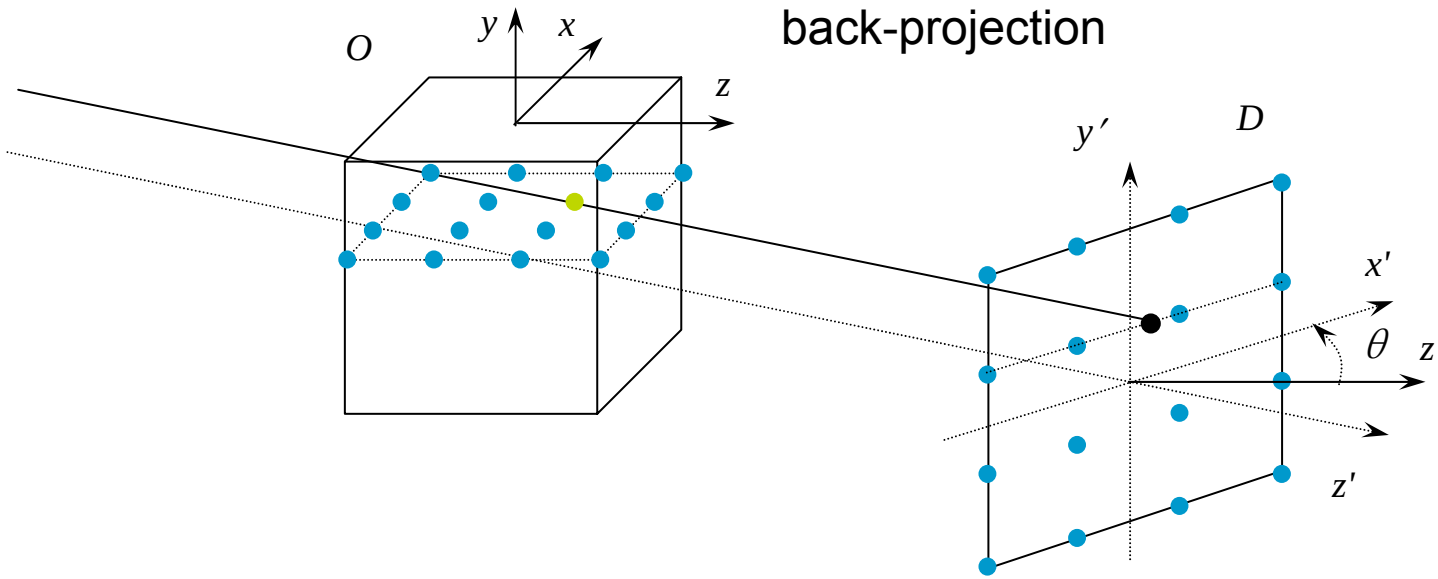
Filtered back-projection (FBP) algorithm

If the source-to-object distance is much larger than the object size then the incident beam can be considered as parallel

ramp-filtering

$$f(x,y,z)=\int_0^{\pi}\Big[\mathbf{F}_1^{-1}\big(|\xi'|\,\mathbf{F}_1\big[(\mathbf{P}_\theta f)(x',y')\big]\big)\Big]_{\substack{x'=x\sin\theta+z\cos\theta\\ y'=y}}\,d\theta$$

back-projection



Typical data sizes in CT reconstruction

- **CT reconstruction is data Input/Output intensive**

N / M*	N^2 float (projection / slice)	NM float (sinogram)	N^2M float (all sinograms)	N^3 float (all slices)
1k / 720	4 MB	2.8 MB	2.8 GB	4 GB
2k / 1,440	16 MB	11¼ MB	22½ GB	32 GB
4k / 2,880	64 MB	45 MB	180 GB	256 GB
8k / 5,760	256 MB	180 MB	1.4 TB	2 TB
16k / 11,520	1 GB	720 MB	11¼ TB	16 TB

* N is the linear size of a projection/slice
M is the number of projections

Implementation model for the FBP and FDK algorithms

• Reconstruction of the object is done in a slice-by-slice manner (at any time only a single axial slice is reconstructed by each host process/thread)

• Sinograms rather than projections are used as the input data

Pros:

• Allows one to minimize the amount of utilized host RAM and GPU memory (e.g., in the FBP implementation, memory for only a single reconstructed slice and a single sinogram are allocated in the host RAM and on the GPU)

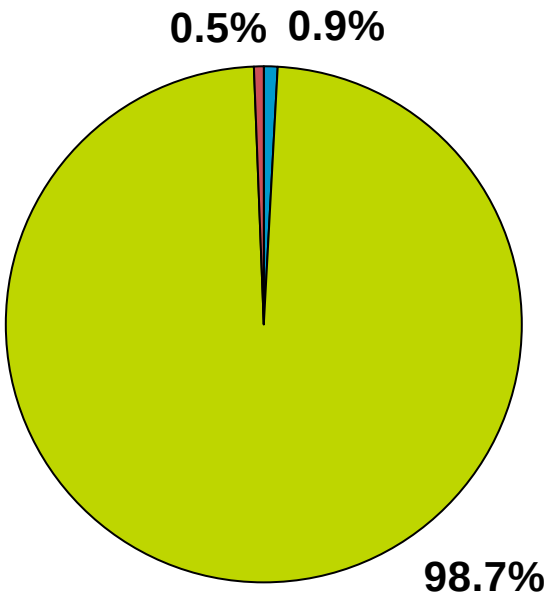
•Allows one to reconstruct axial slices of up to 16k×16k pixels (32-bit each) using top-end GPUs (with more than 1GB memory onboard)

Cons:

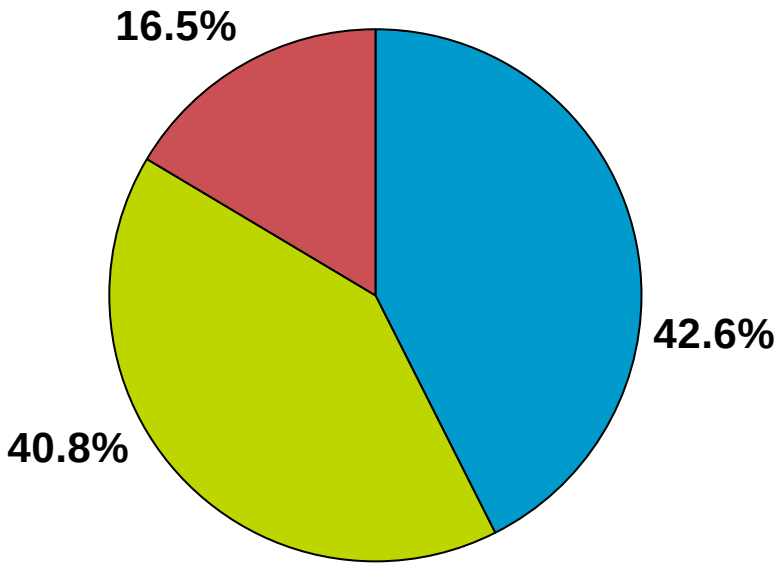
• All projections should be acquired before the reconstruction starts and converted to sinograms (e.g. during a pre-processing step)

FBP CT reconstruction, 1024x(1024x360) -> 1024^3

CPU
(Xeon E5420 @ 2.5GHz)



CPU+GPU
(GeForce GTX260)



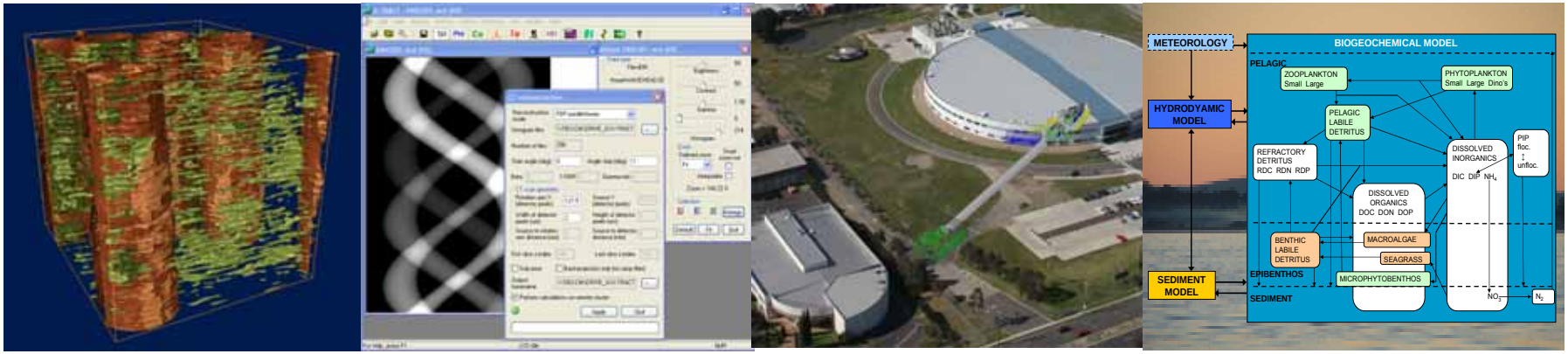
R/W
BP
Other

CT reconstruction time

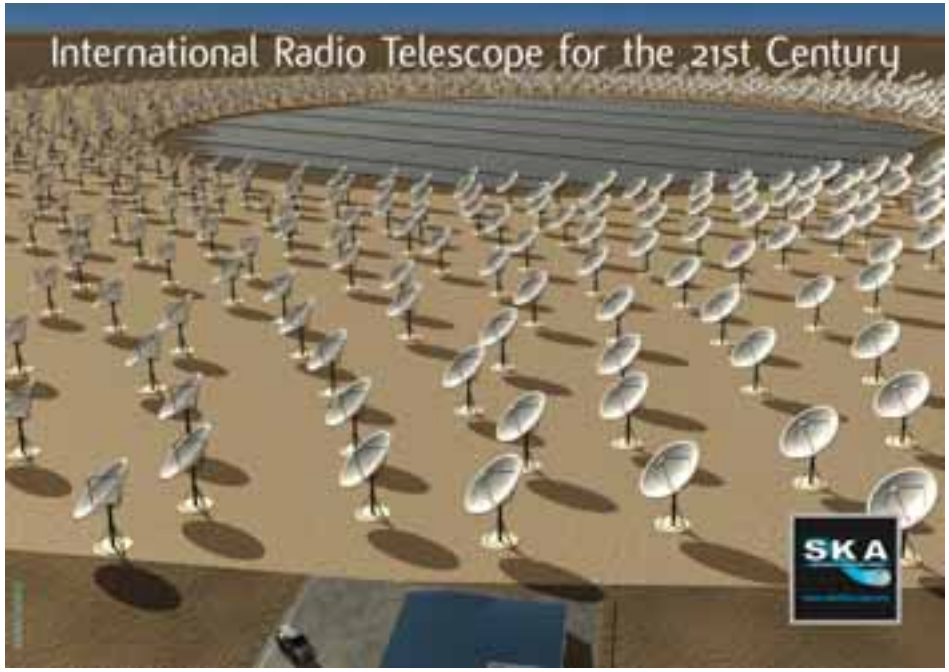
Volume	CPU(1)	CPU(4)	CPU+ GPU(1	CPU+ GPU(4)
512^3	**32' 17"**	**8' 10" (3.95×)**	**41.7" (46.4×)**	**20.8" (93.1×)**
1024^3	**9h 6' 2"**	**2h 25' 9" (3.76×)**	**5' 42" (95.8×)**	**2' 56" (186×)**
2048^3	**~161.7h**	**~40.5h (3.995×)**	**1h 18' 14" (124×)**	**41' 53" (232×)**

Square Kilometre Array

Radio Telescope



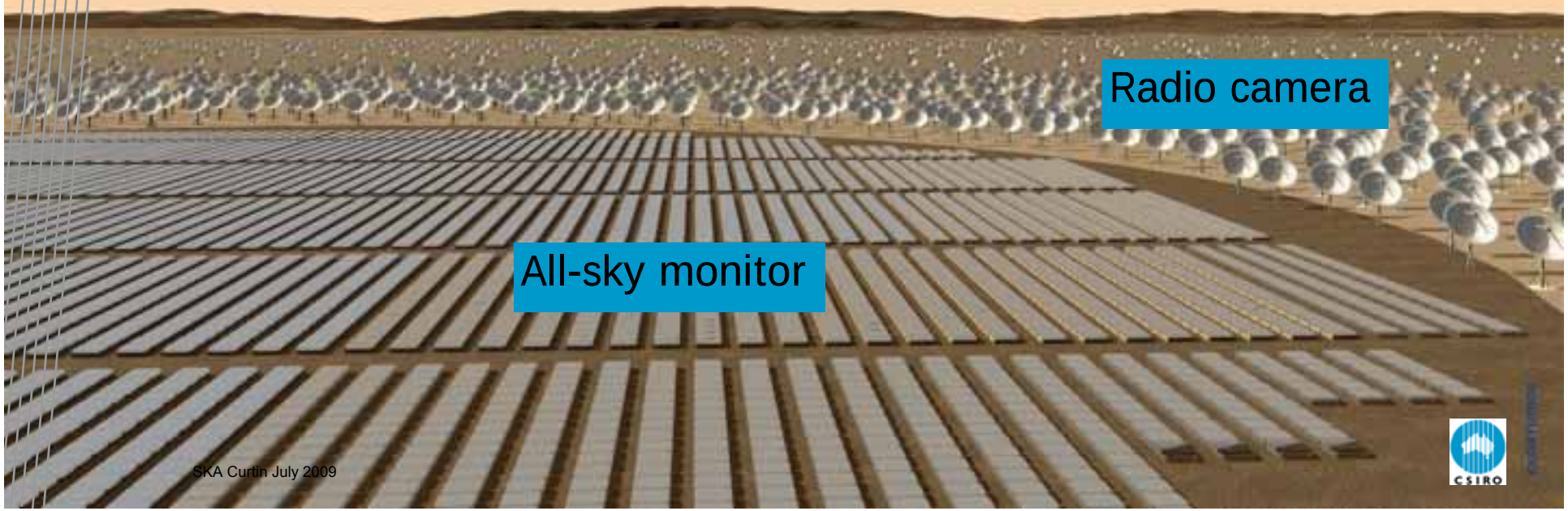
The Square Kilometre Array



- **2020 era radio telescope**
- **Very large collecting area (km^2)**
- **Very large field of view**
- **Wide frequency range (70MHz - 25 GHz)**
- **Large physical extent (3000+ km)**

SKA design

- Up to 1500 antennas (15m diameter) in the central 5 km
- Another 1500 from 5 km to 3000+ km
- Aperture arrays for all sky monitor
- Antennas for surveys



Australian SKA Pathfinder = 1% SKA

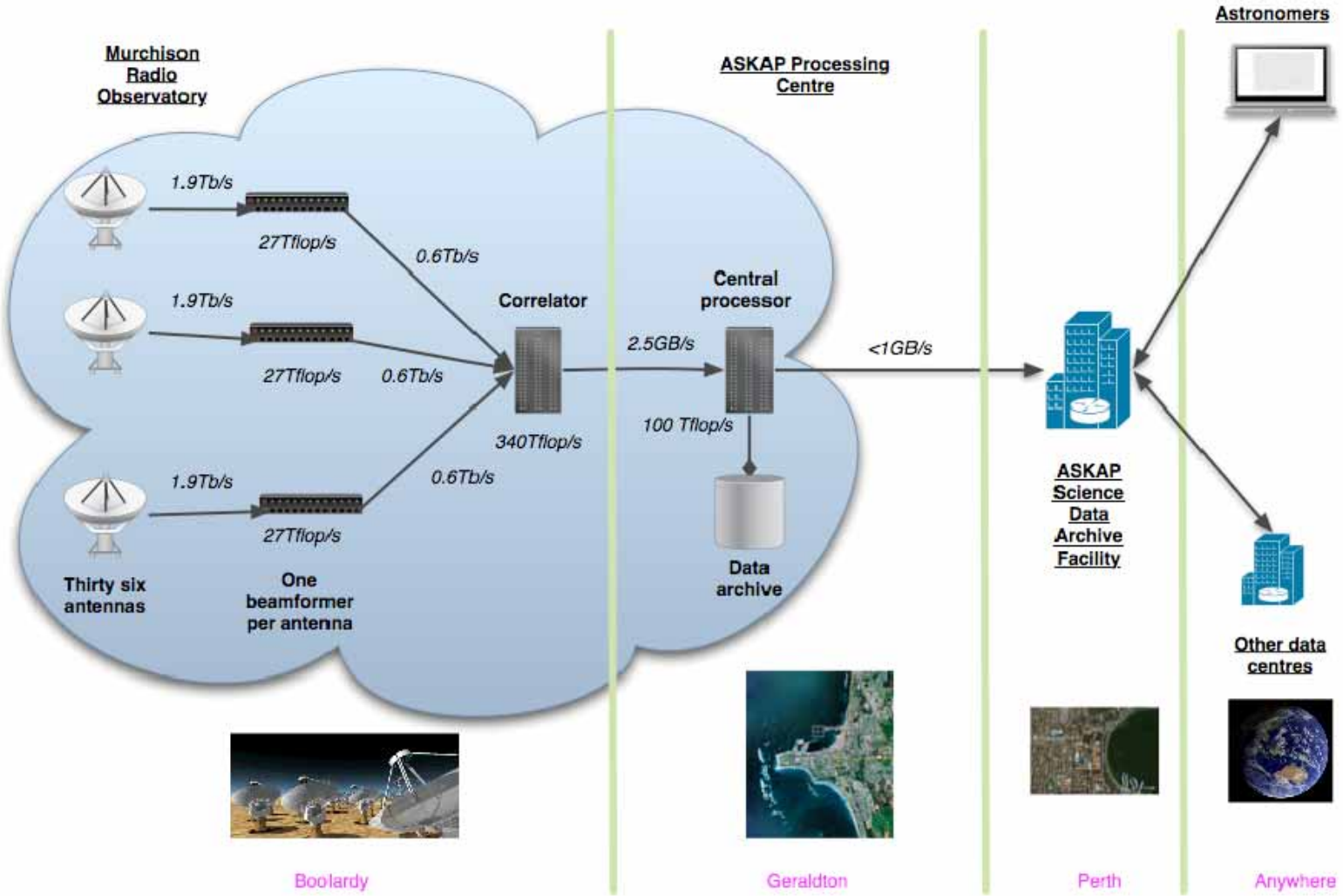
- **Wide field of view telescope (30 square degrees)**
- **Sited at Boolardy, Western Australia**
- **36 antennas compared to ~ 3600 for SKA**

An illustration of the speed of the Pathfinder

- **ATCA image of Centaurus**
- Required 1200 hours observing on the Australia Telescope Compact Array in Narrabri

- The Pathfinder will take about 10 minutes

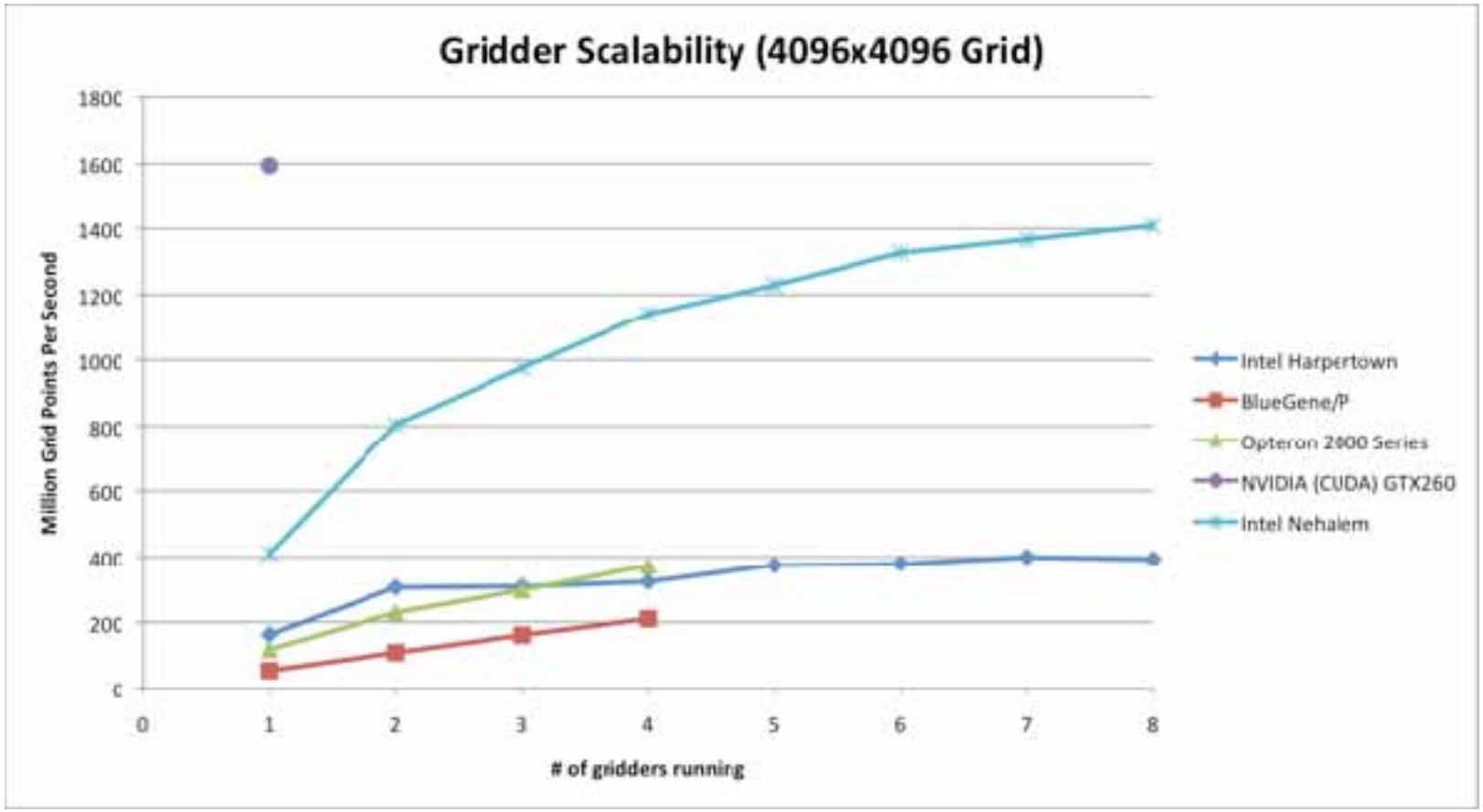
ASKAP data flow, processing, and storage



Evaluation of Hardware for gridding/degridding

- **Built a small benchmark from the core gridding/degridding algorithm.**
  - About 90% of our computing requirements relate to this algorithm
- **Distributed benchmark very widely**
- **Benchmarked on systems from:**
  - Intel (Harpertown and Nehalem CPUs)
  - AMD (Opteron 2000 series CPUs)
  - NEC (SX-8R & SX-9R)
  - SGI (SGI Altix 4700 Itanium & SGI Altix XE)
  - IBM (BlueGene/P)
  - Cray (XT5 & X2)
- **Also investigated special purpose hardware**
  - GPGPU, FPGA, Cell

Performance on multiple gridders on one node

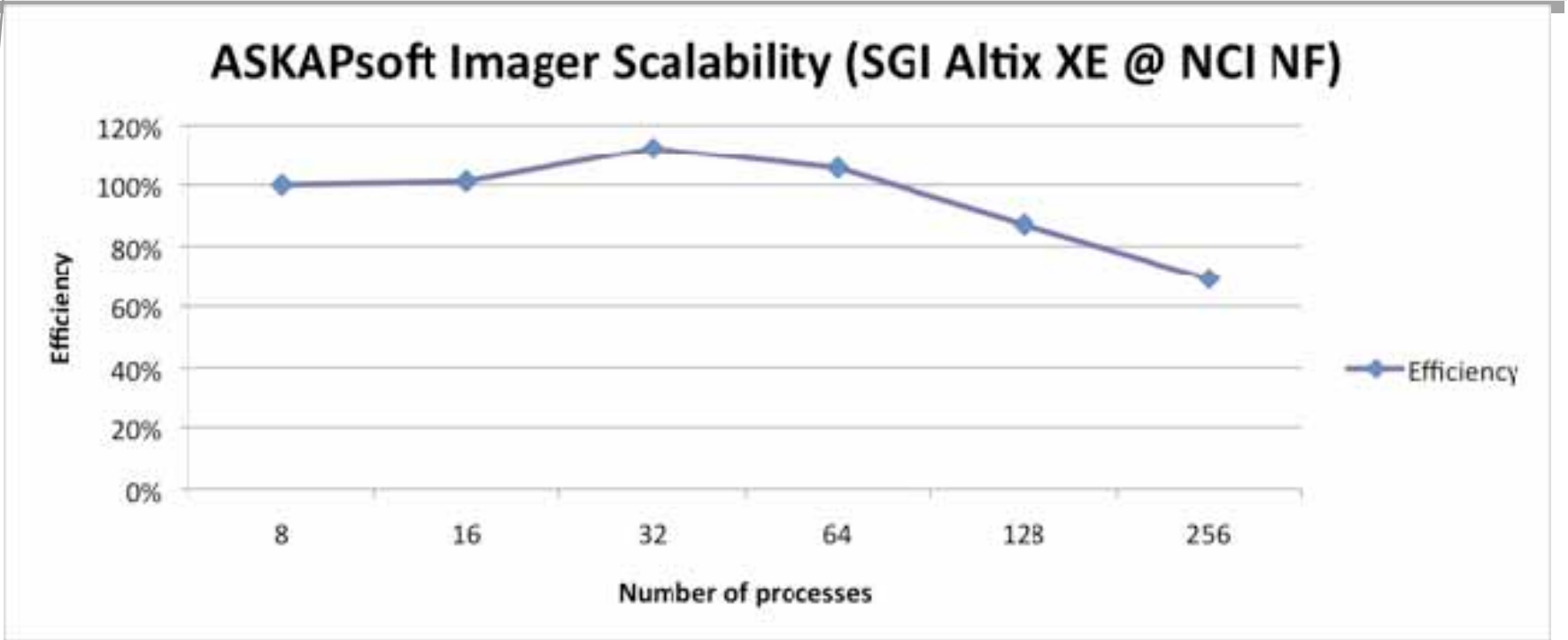


Other numbers are confidential

Scaling to ~ 10,000 cores

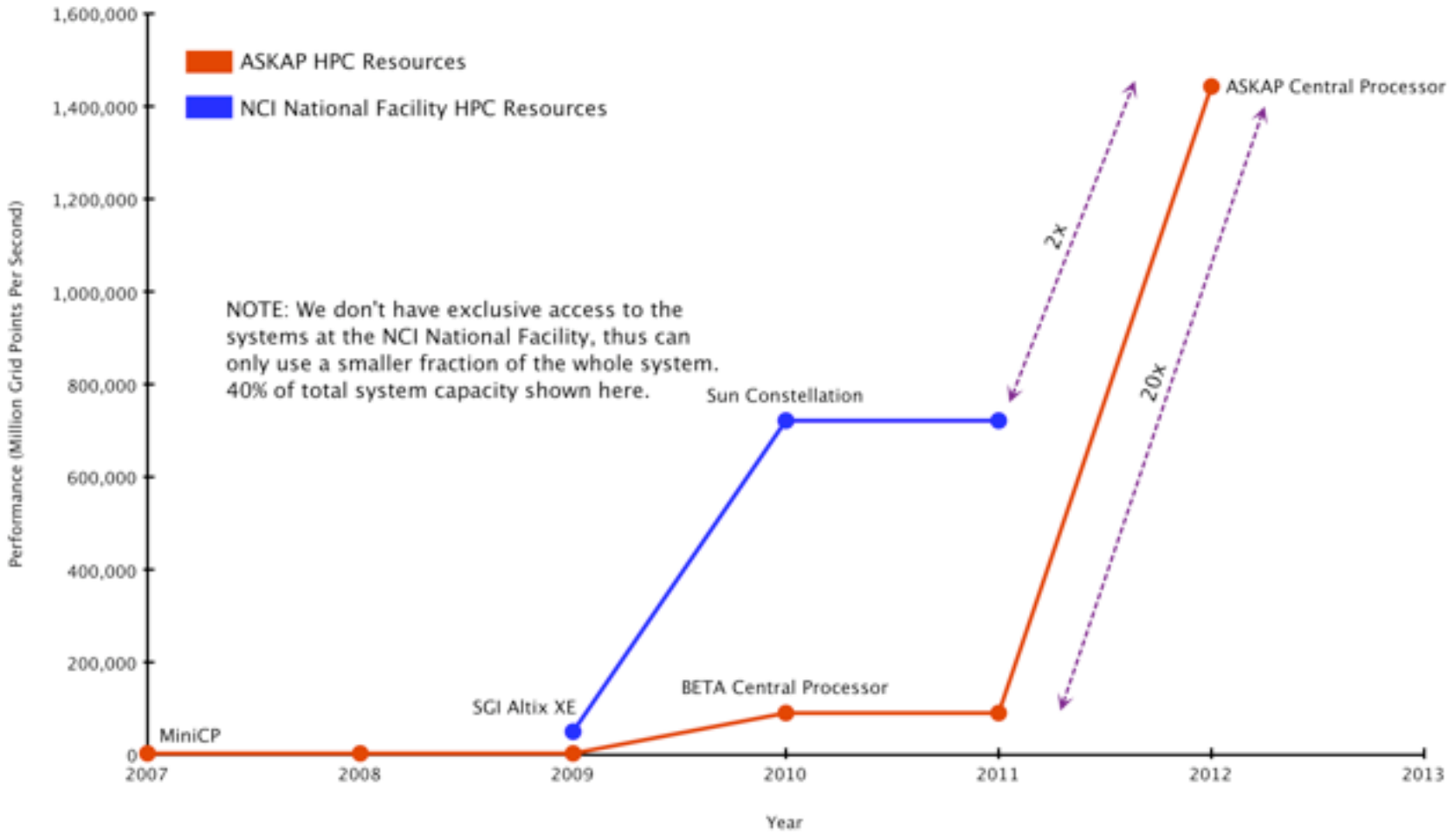
- **Goal is to get 80% efficiency at full scale**
- **Necessary to continuously measure and improve scaling**
- **Currently performing scalability testing at the National Computational Infrastructure (NCI) National Facility**
- **SGI XE Cluster System**
  - 156 x SGI Altix XE 320 nodes
  - 1248 cores (312 x 3.0GHz Intel Harpertown CPUs)
  - DDR InfiniBand interconnect
  - 18 x Quad-core SGI Altix XE 210 servers for Lustre filesystem
- **Migrating to the new Sun Constellation system late 2009**
  - Hosted by the NCI National Facility @ ANU
  - 1500 Sun Blade modules (12,000 cores)
  - 500TB Lustre Filesystem

Imager scalability on NCI SGI Altix



- **Obtain ~ 70% efficiency at 256 cores**
- **Thought to be due to Lustre configuration**

Central Processor Scaling Timeline

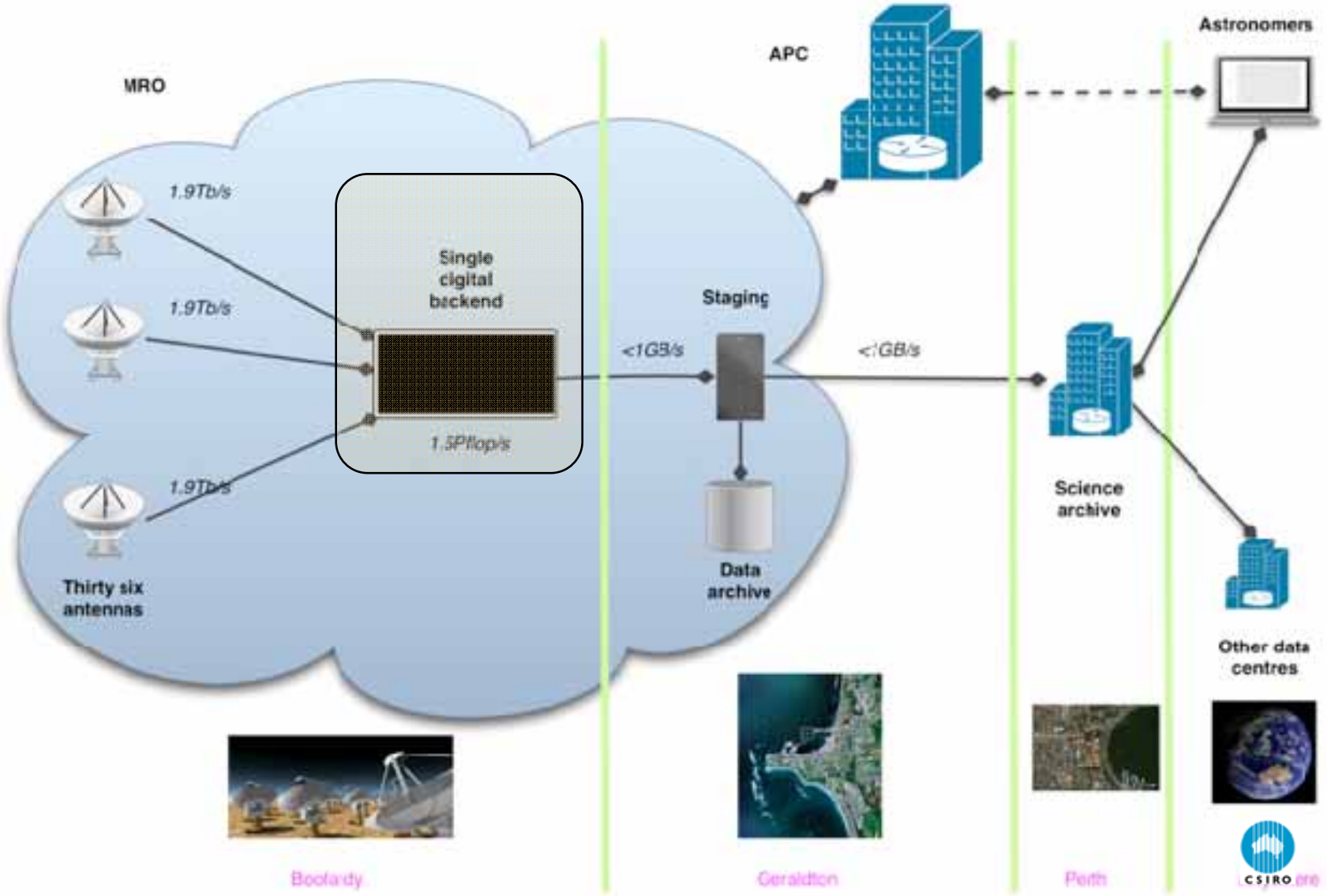


BETA and ASKAP computing needs

- **BETA hardware requirements:**
  - 3-6 TFlop/s
    - 256-512 cores (as of late 2008 / early 2009)
  - 1-2 TB memory
  - Good memory bandwidth (>15 GB/s per socket)
  - 50 TB persistent storage (1 GB/s I/O rate)
  - Modest network interconnect
    - Single 1GbE for compute nodes
    - Single 10GbE for the ingest and output nodes
- **ASKAP**
  - 100 TFlop/s
    - ~8000 cores (as of late 2008 / early 2009)
    - ~10000 if we assume a more realistic 80% efficiency
  - 16-150 TB memory (depending on processing model)
  - Good memory bandwidth (>15 GB/s per socket)
  - 1 PB persistent storage (8-10 GB/s I/O rate)
  - Modest network interconnect
    - 1GbE for compute nodes
    - 2-4 x 10GbE for the ingest and output nodes

Memory bandwidth more important than flops

ASKAP Single Digital Backend



ASKAP SDB beam forming and correlation load

- **Computing load**
  - 2PFlop/s
  - Enormous input data rate ~ 72Tb/s
  - Moderate output data rate ~ 25Gb/s
  - Lots of Complex Multiply ACcumulates (CMACs)
  - Moderate length FFTs
  - Heavily streamed - not much memory needed
  - Two major data reshuffles
    - Beamforming - elements to elements for each antenna
    - Correlation - antennas to antennas for each beam
- **Derived requirements**
  - High input capability
  - Low computational intensity
  - Interconnect flexibility
  - High interconnection bandwidth
  - Moderate memory
  - Power efficiency

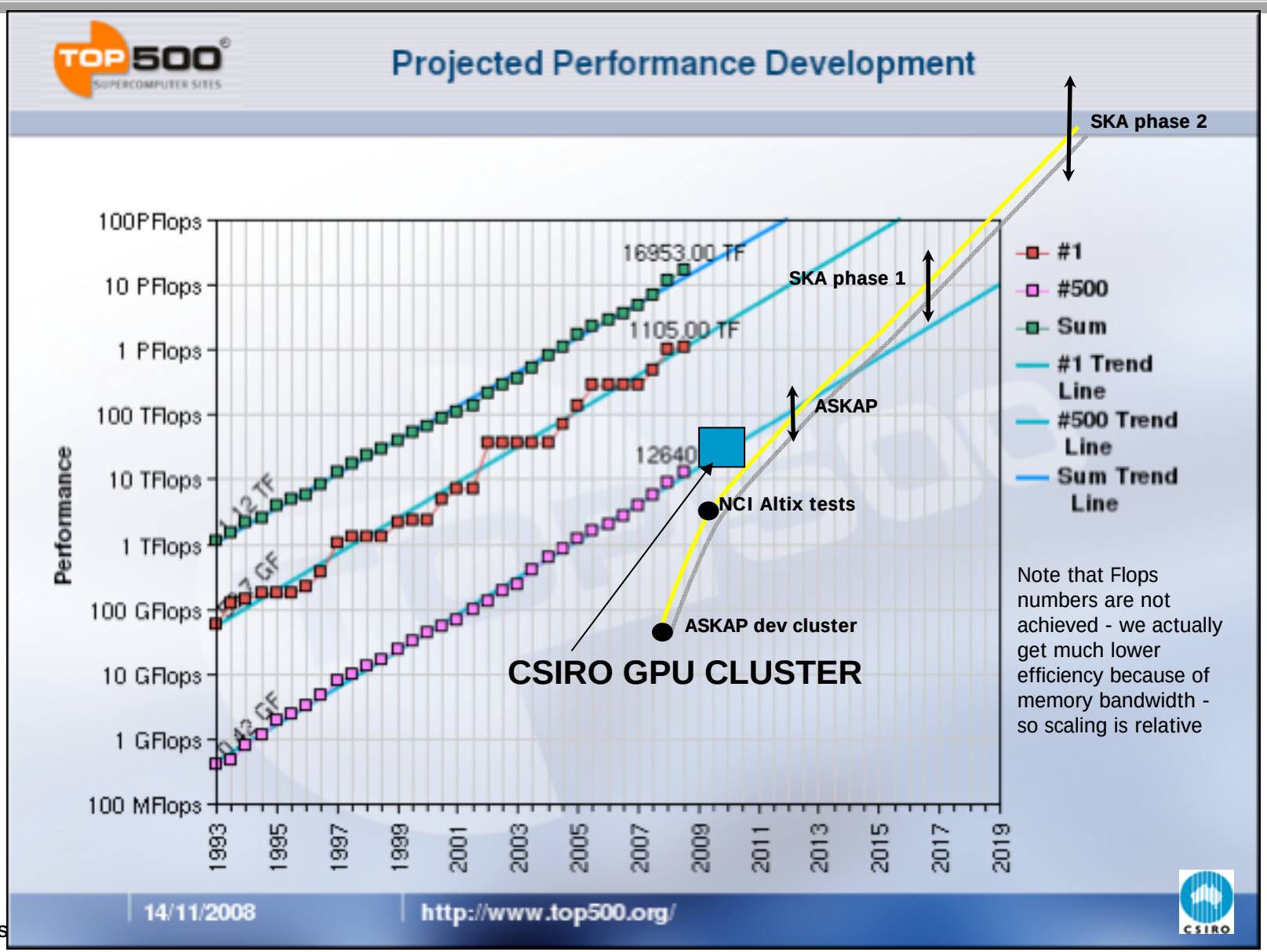
SDB = A toolkit for innovation

- **Telescope = device for imaging from measurements of E-field**
- **Standard approach to ASKAP**
  - Antenna+PAF->Beamformers->Correlator->Inverse FFT Box->Science result
- **Another approach**
  - Antenna+PAF->Massive solver of linear equations->Science result
- **Or**
  - Antenna+PAF->2 Petaflop programmable box->Science result
- **And the next step is**
  - SKA->few Exaflop programmable box->Science result

Scaling ASKAP computing to SKA

- **Roughly 100 times more antennas**
  - Assume PAF wide field option
- **Processing scales as square of antennas**
  - SKA processing > 10,000 ASKAP processing for same field of view
  - > 1 Exaflop/s processing rate
  - 1 Exabyte/day data rate input to imaging machine
- **Flops becoming irrelevant**
- **Data movement will determine architecture, cost, and power**
  - Disk
  - IO subsystem
  - Processing subsystem
  - Off chip memory
  - On chip caches
- **Algorithms will have to change**

Climbing Mount Exaflop



Note that Flops numbers are not achieved - we actually get much lower efficiency because of memory bandwidth - so scaling is relative

Open questions at SKA level

- **Computing**
  - Scale of processing?
  - Processing model?
  - Data storage and distribution?
- **Software**
  - Models for development?
  - Buy vs build vs reuse?
  - Dealing with complexity?
- **Data processing**
  - Scalable algorithms?
  - Calibration of specific telescope design?
  - Simulation capabilities?

Summary

- **Progressing towards real-time pipelines for ASKAP data**
- **Solving multiple technical problems on the way**
  - Distributed, scalable synthesis processing code
  - Algorithmic changes and innovations
- **Conducting end-to-end testing on simulated and real data**
  - Simulation capabilities will be available to Science Survey Teams
- **Progressing to first real tests on BETA in late 2010**
- **Developing some understanding of related SKA processing**
  - SKA Phase 2 PAF option requires ~ 10^8 cores
  - CPU power ~ 1EFlop/s
  - Memory bandwidth ~ 1EB/s
- **SKA computing has unknown architecture, cost, and power!**

Acknowledgements

- **Tim Cornell, ATNF**
- **Luke Domanski, CMIS**
- **Tim Gureyev, CMSE**
- **Nick Markwart, IM&T**
- **Tad Matuszkiewicz, IM&T**
- **Lawrence Murray, CMIS**
- **Yakov Nesterets, CMSE**
- **Pascal Vallotton, CMIS**
- **Dadong Wang, CMIS**

Mathematical and Information Sciences
Dr John A Taylor
Science and Business Leader
Computational and Simulation Sciences

Phone: +61 2 6216 7077
Email: John.A.Taylor@csiro.au
Web: http://www.csiro.au/science/CSS.html

Questions……

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