NCI
NATIONAL COMPUTATIONAL INFRASTRUCTURE

Scaling Weather Climate and Environmental Science Applications, and experiences with Intel Knights Landing

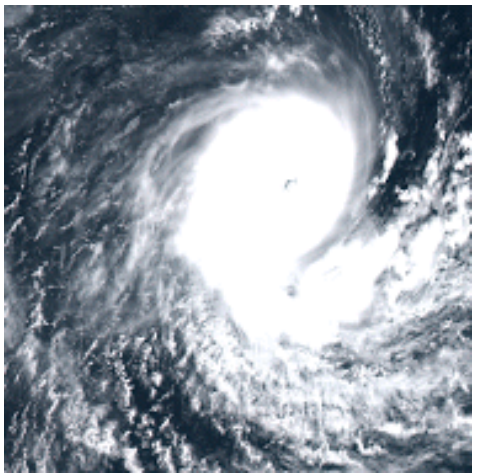
Ben Evans, Dale Roberts

17th Workshop on High Performance Computing in Meteorology

Australian Government Department of Education
Australian National University
Australian Government Bureau of Meteorology
CSIRO
Australian Government Geoscience Australia
Australian Government Australian Research Council

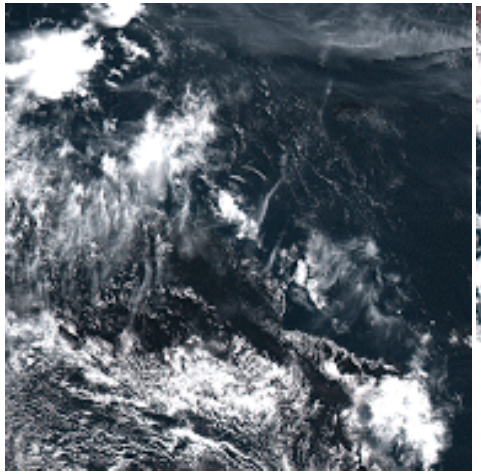
- Modelling Extreme & High Impact events – BoM
- NWP, Climate Coupled Systems & Data Assimilation – BoM, CSIRO, Research Collaboration
- Hazards - Geoscience Australia, BoM, States
- Geophysics, Seismic – Geoscience Australia, Universities
- Monitoring the Environment & Ocean – ANU, BoM, CSIRO, GA, Research, Fed/State
- International research – International agencies and Collaborative Programs

Tropical Cyclones



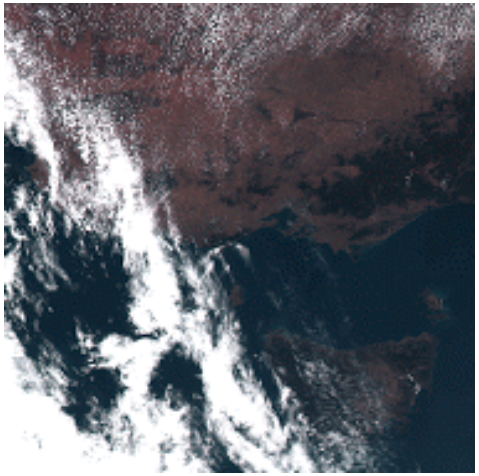
Cyclone Winston
20-21 Feb, 2016

Volcanic Ash



Manam Eruption
31 July, 2015

Bush Fires



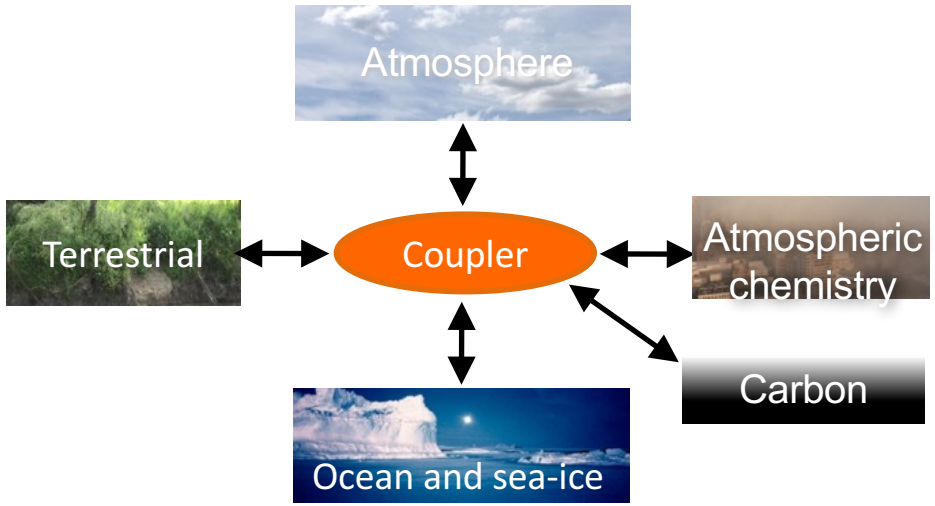
Wye Valley and
Lorne Fires
25-31 Dec, 2015

Flooding

St George, QLD
February, 2011

ACCESS Model: NWP, Climate, Ocean, ESM, Seasonal and Multi-year

ACCESS
The Australian Community Climate and Earth-System Simulator



Core Model

- Atmosphere – UM 10.5+
- Ocean – MOM 5.1 (for most models)
- NEMO 3.6 (for GC3 seasonal-only)
- Sea-Ice – CICE5
- Coupler – OASIS-MCT

Carbon cycle (ACCESS-ESM)

- Terrestrial – CABLE
- Bio-geochemical
- Couple to modified ACCESS1.3

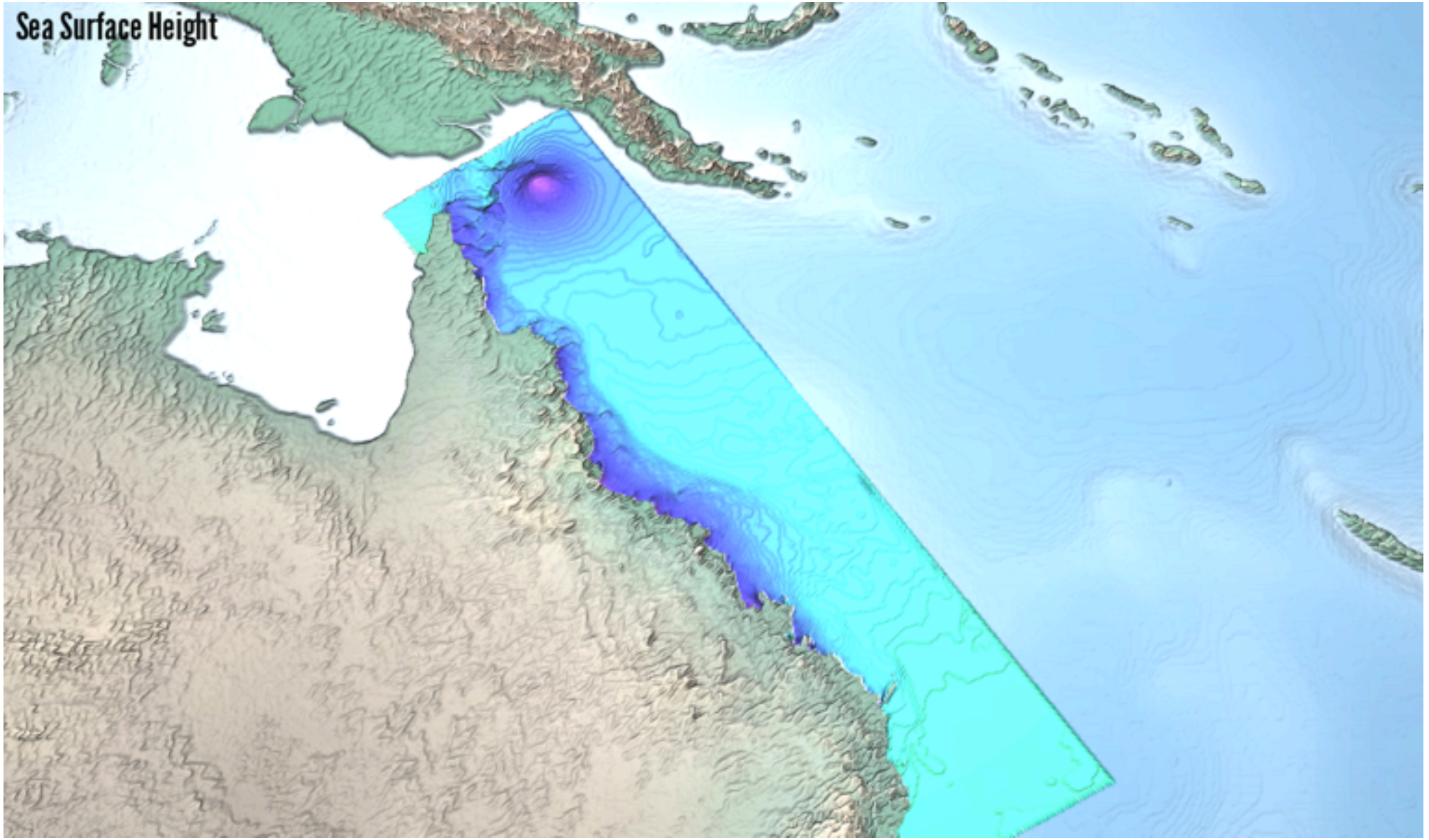
Aerosols and Chemistry

- UKCA

Wave

- WW3

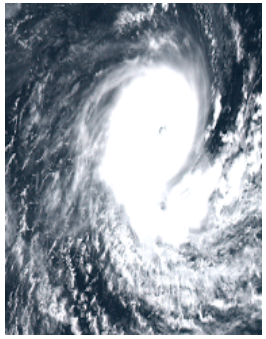
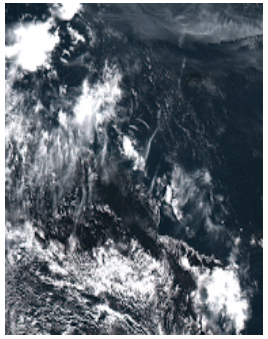
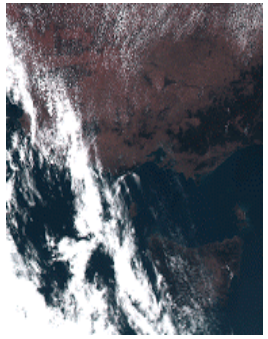
Additional priority codes – Australian storm surge model using ROMS



NCI Research Data Collections: Model, data processing, analysis

1. Climate/ESS Model Assets and Data Products
2. Earth and Marine Observations and Data Products
3. Geoscience Collections
4. Terrestrial Ecosystems Collections
5. Water Management and Hydrology Collections

   http://geonetwork.nci.org.au

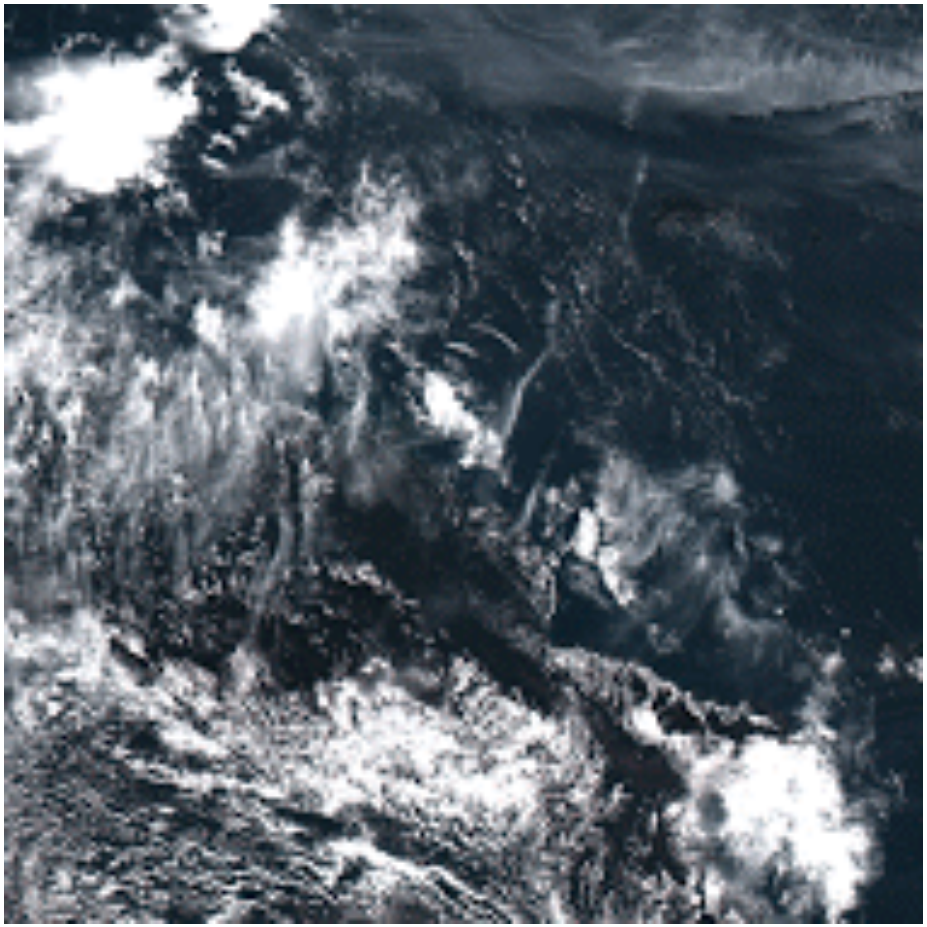
Data Collections	Approx. Capacity
CMIP5, CORDEX, ACCESS Models	5 Pbytes
Satellite Earth Obs: LANDSAT, Himawari-8, Sentinels, plus MODIS, INSAR, …	2 Pbytes
Digital Elevation, Bathymetry Onshore/Offshore Geophysics	1 Pbytes
Seasonal Climate	700 Tbytes
Bureau of Meteorology Observations	350 Tbytes
Bureau of Meteorology Ocean-Marine	350 Tbytes
Terrestrial Ecosystem	290 Tbytes
Reanalysis products	100 Tbytes

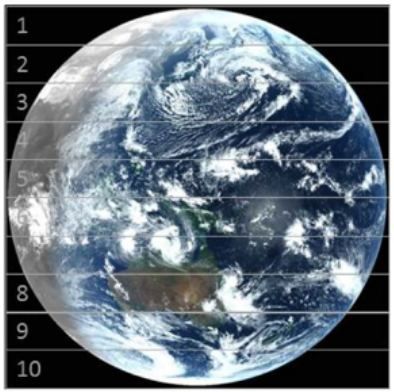
Himawari-8 Observations, Data Assimilation and Analysis

Captured at JMA,
Processed after acquisition at BoM
Made available at NCI

Data Products still to be generated, but first stage was to make the image data available.

10 minute capture and process. Then also need to make it available for broad analysis.

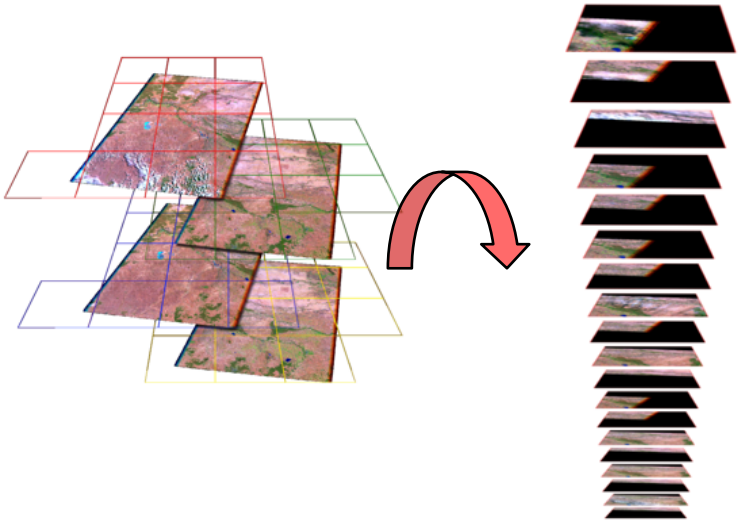




	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	Start (s)	Stop (s)	Delta (s)
Generate low resolution HSF	8																			0	2	2
Solar Geometry 2000m		4	4																	2	24	22
Solar Geometry 1000m				1																24	34	10
Solar Geometry 500m				1	1															24	50	26
Generate IR and VIS 2000m obs (IR grib)		4	4	4																2	34	32
Generate IR images				4	4	4														34	54	20
Generate vis 2000m BRF						3	3	3												54	81	27
Generate vis 1000m BRF						4	4	4	4											54	90	36
Generate vis 500m BRF						1	1	1	1	1	1	1	1	1	1	1	1	1		54	181	127
Generate vis image files										7	7	7	7	7	7	7	7	7		90	179	89

Earth Observation Time Series Analysis

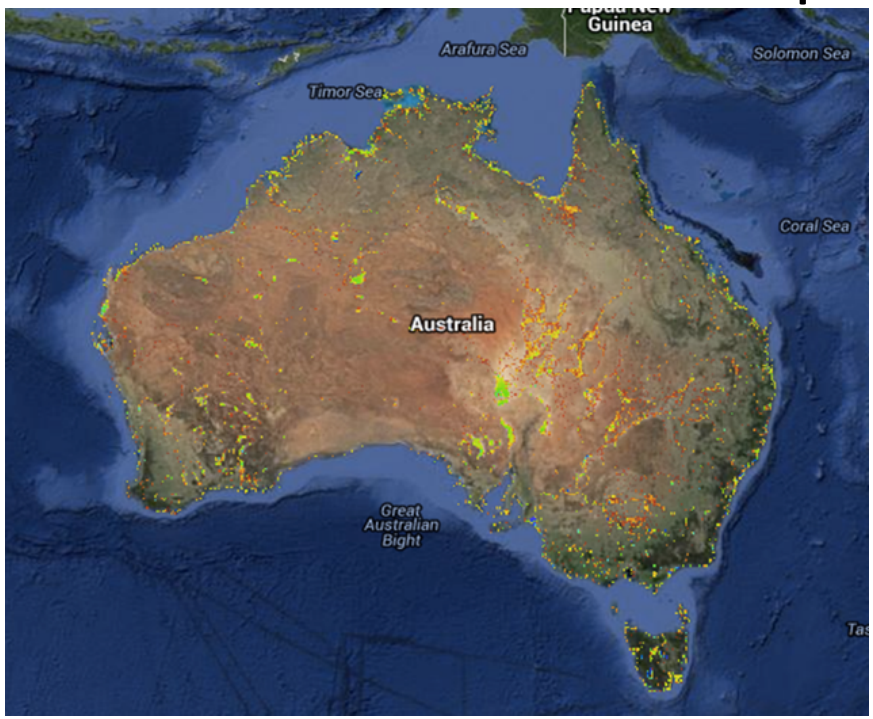
- Over 300,000 Landsat scenes (spatial/temporal) allowing flexible, efficient, large-scale in-situ analysis
- Spatially-regular, time-stamped, band-aggregated tiles presented as temporal stacks.



Spatially partitioned tiles

Temporal Analysis

Continental-Scale Water Observations from Space



WOFS water detection

- **27 Years** of data from LS5 & LS7(1987-2014)
- **25m** Nominal Pixel Resolution
- Approx. *300,000* individual source ARG-25 scenes in approx. 20,000 passes
- Entire 27 years of 1,312,087 ARG25 tiles => **93x10^{12} pixels** visited
- **0.75 PB** of data
- **3 hrs** at NCI (elapsed time) to compute.

Australian Government
Geoscience Australia

CSIRO

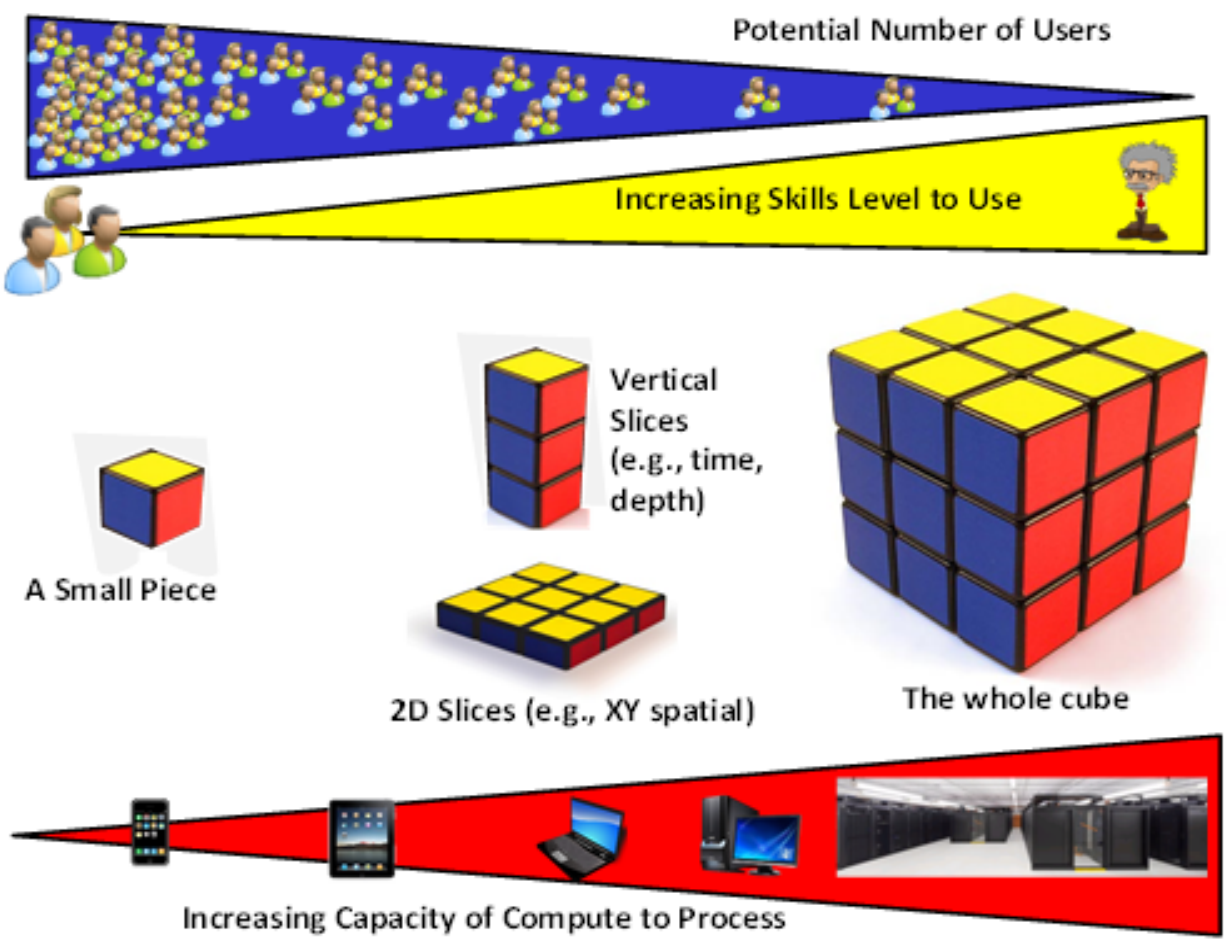
NCI
NATIONAL COMPUTATIONAL INFRASTRUCTURE

Enable global and continental scale … and to scale-down to local/catchment/plot

- Water availability and usage over time
- Catchment zone
- Vegetation changes
- Data fusion with point-clouds and local or other measurements
- Statistical techniques on key variables

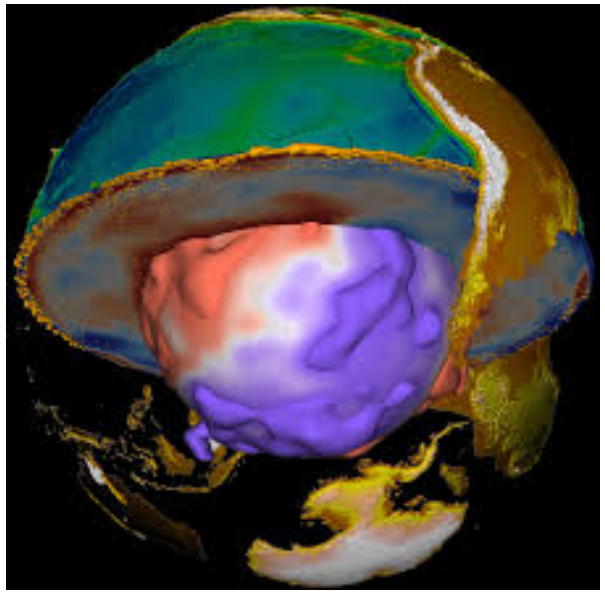
Preparing for:

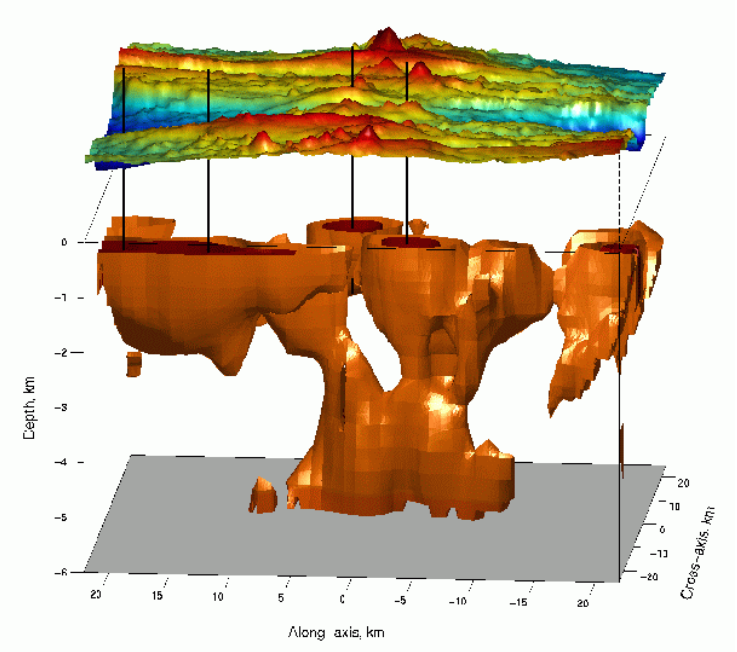
- Better programmatic access
- Machine/Deep Learning
- Better Integration through Semantic/Linked data technologies



Emerging Petascale Geophysics codes

- Assess priority Geophysics areas
  - 3D/4D Geophysics: Magnetotellurics, AEM
  - Hydrology, Groundwater, Carbon Sequestration
  - Forward and Inverse Seismic models and analysis (onshore and offshore)
  - Natural Hazard and Risk models: Tsunami, Ash-cloud
- Issues
  - Data across domains, data resolution (points, lines, grids), data coverage
  - Model maturity for running at scale
  - Ensemble, Uncertainty analysis and Inferencing





NCI High Performance Optimisation and Scaling activities 2014-17

- **Objectives:**
  - Upscale & increase performance of high-priority national codes – particularly Weather and Climate

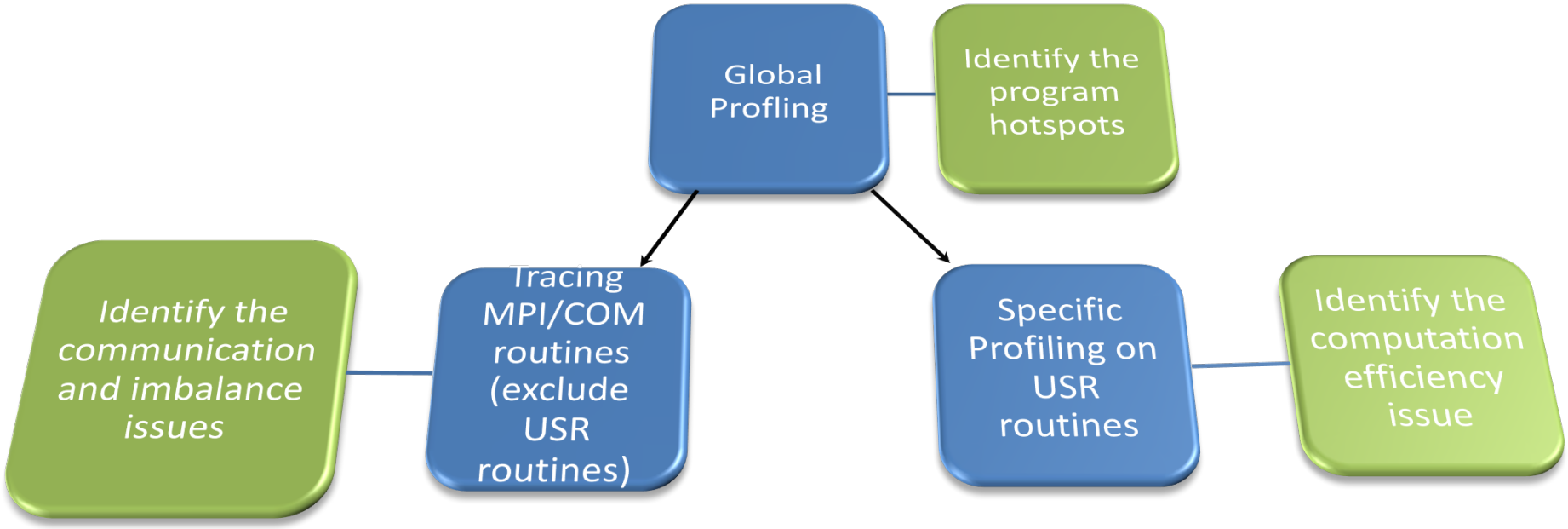
- **Year 1**
  - Characterise, Optimise and Tune of critical applications for higher resolution
  - Best practise configuration for improved throughput
  - Establish analysis toolsets and methodology

- **Year 2**
  - Characterise, Optimise and Tune of next generation high priority applications
  - Select high priority geophysics codes and exemplar HPC codes for scalability
  - Parallel Algorithm Review and I/O optimisation methods to enable better scaling
  - ***Established TIHP Optimisation work package for UM codes (Selwood, Evans)***

- **Year 3**
  - Assess broader set of community codes for scalability
  - Updated hardware (many-core), memory/data latency/bandwidths, energy efficiency
  - Communication libraries, math libraries

Common Methodology and approach for analysis

- Analyse code to establish strengths and weaknesses.
- Full code analysis including hotspot and algorithm choices
- Expose model to more extreme scaling – e.g., realistic higher resolution
- Analyse and compare different software stacks
- Decomposition strategies for nodes and node tuning
- Parallel Algorithms, MPI communication Patterns. e.g. Halo analysis, grid exchanges
- I/O techniques: Evaluate serial and parallel techniques
- Future hardware technologies



Scaling/Optimisation:
BoM Research-to-Operations High Profile cases

Domain	Yr1 – 2014/5	Yr2 – 2015/6	Yr3 – 2016/7
Atmosphere	APS1 (UM 8.2-4) • Global N320L70 (40km) and pre-APS2 N512L70 (25km) • Regional N768L70 • (~17km) • City 4.5k	UM10.x (PS36) • Global N768L70 (~17km) • Regional, City	APS3 prep • UM10.x latest • ACCESS-G (Global) N1024L70/L85 (12km) or N1280L70/L85 (10km) • ACCESS-GE Global Ensemble (N216L70) (~60km) • ACCESS-TC 4km • ACCESS-R (Regional) 12km • ACCESS-C (City) 1.5km

NCI-Fujitsu Scaling/Optimisation: BoM Research-to-Operations High Profile cases

Domain	Yr1 – 2014/5	Yr2 – 2015/6	Yr3 – 2016/7
Data assimilation	4D-VARv30 • N216L70, N320L70		• 4D-VAR Latest for Global at N320L70 • enKF-C

Ocean	MOM5.1
	• OFAM3
	• 0.25°, L50
		MOM5.1
		• 0.1°, L50
			• OceanMAPS3.1 (MOM5) with enKF-C
			• MOM5/6 0.1° and 0.03°
			• ROMS (Regional)
			  • StormSurge (2D)
			  • eReefs (3D)

Wave		WaveWatch3 v4.18 (v5.08 beta)	• AusWave-G 0.4° • AusWave-R 0.1°

NCI-Fujitsu Scaling/Optimisation: BoM Research-to-Operations High Profile cases

Domain	Yr1 – 2014/5	Yr2 – 2015/6	Yr3 – 2016/7
Coupled Systems	Climate: ACCESS-CM • GA6 (UM8.5)+MOM5 with OASIS-MCT • Global N96L38 (135km), 1° and 0.25° ocean.	Climate: ACCESS-CM cont.	Climate: ACCESS-CM2 (AOGCM) • GA7 (UM 10.4+ and GLOMAP aerosol), • Global N96L85, N216L85 (~60km) • MOM5.1 0.25° • CABLE2 • UKCA aerosols
			Earth System: ACCESS-ESM2 • ACCESS-CM2+Terrestrial Biochemistry – CASA-CNP • Oceanic biogeochemistry – WOMBAT • Atmospheric chemistry – UKCA

	Seasonal Climate:
	• ACCESS-S1 - UK GC2 with OASIS3
	• N216L85 (~60km) NEMO 0.25°
		GC2
		NCI profiling methodology applied for MPMD
			• Multi-week and Seasonal Climate:
			  • ACCESS-S2 / UK GC3
			  • Atmos:N216L85 (60km) and NEMO 3.6 0.25° L75

HPC General Scaling and Optimisation Approach

Domain	Yr2 – 2015/6	Yr3 – 2016/7
Profiling Methodology	Create Methodology for profiling codes	Updates to Methodology based on application across more codes
I/O profiling	• Baseline profiling for comparison of NetCDF3, NetCDF4, HDF5 and GeoTIFF and API options (e.g. GDAL). • Profiling comparison of IO performance of Lustre, NFS • Compare MPI-IO vs POSIX vs HDF5 on Lustre	• Advanced Profiling HDF5 and NetCDF4 for compression algorithms, multithreading, cache management • Profiling analysis of other data formats • e.g., GRIB, Astronomy FITS, SEG-Y, BAM
Accelerator Technology Investigation		• Intel Phi (Knights Landing) • AMD GPU
Profiling tools suite	Review Major open source Profiling Tools	Investigation of profilers for Accelerators

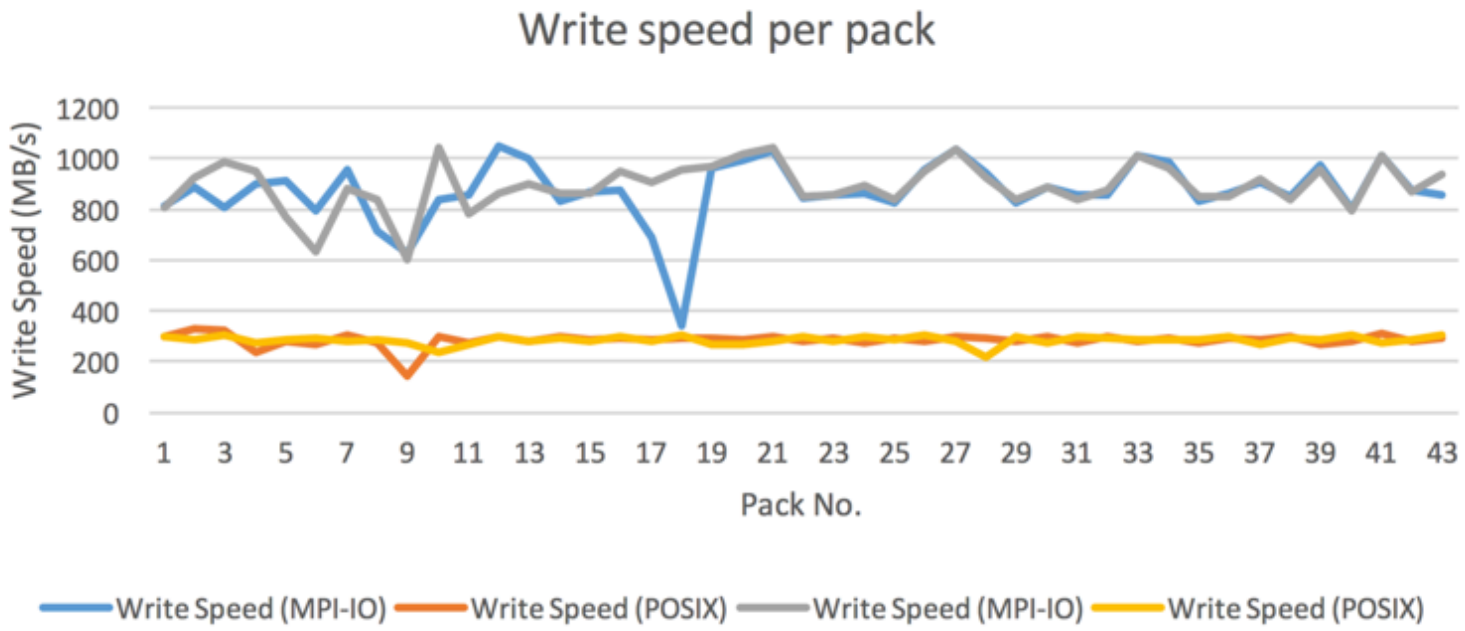
HPC General Scaling and Optimisation Approach

Domain	Yr2	Yr3
Compute Node Performance Analysis	• Partially committing nodes • Hardware Hyper-threading • Memory Bandwidth • Interconnect bandwidth	• Evaluating Energy Efficiency vs performance of next generation processors • Broadwell improvements • Memory speed • Vectorisation • OpenMP coverage
Software Stacks	• OpenMPI vs IntelMPI analysis • Intel compiler versions	• OpenMPI vs IntelMPI analysis • Intel compiler versions • Math Libraries
Analysis other Earth Systems & Geophysics priority codes and algorithms	• Initial Analysis of MPI communications • Commence analysis of high priority/profile HPC codes in	• Detailed Analysis of MPI communication dependent algorithms • Survey of Codes and Algorithms used.

NCI Contributions to UM collaboration so far

- UM 10.4+ IO Server now using MPI-IO
  - Immediately valuable for NWP (UK Met, Aus, …)
  - Critical for next generation processors (i.e., KnL)

- UM 10.5+ OpenMP coverage
  - Increased performance
  - Critical for both current and next architectures, especially with increasing mem bandwidth issues



Knights Landing nodes at NCI

Raijin is a Fujitsu Primergy cluster

- 57,472 cores (Intel Xeon Sandy Bridge technology, 2.6 GHz) in 3592 compute nodes
- Infiniband FDR interconnect
- 10 PBytes Lustre for short-term scratch space
- 30 Pbytes for data collections storage

KnLs

- 32 Intel Xeon Phi 7230 processors
  - 64 cores/256 threads per socket, 1 socket per node
  - 16GB MCDRAM on package (380+GB/s bandwidth)
  - 192GB DDR4-2400MHz (115.2GB/s)
- EDR InfiniBand interconnect between KnLs (100 Gb/s)
- FDR Infiniband links to main lustre storage (56 Gb/s)

Kepler K80 GPUs also available

Basic KNL first impression

- KnL Pros
  - Full x86 compatibility – applications ‘just work’ without need for code major changes
  - AVX 512-bit instruction set gives performance boost for well vectorised applications
  - Potential to process 2 vector operations per cycle

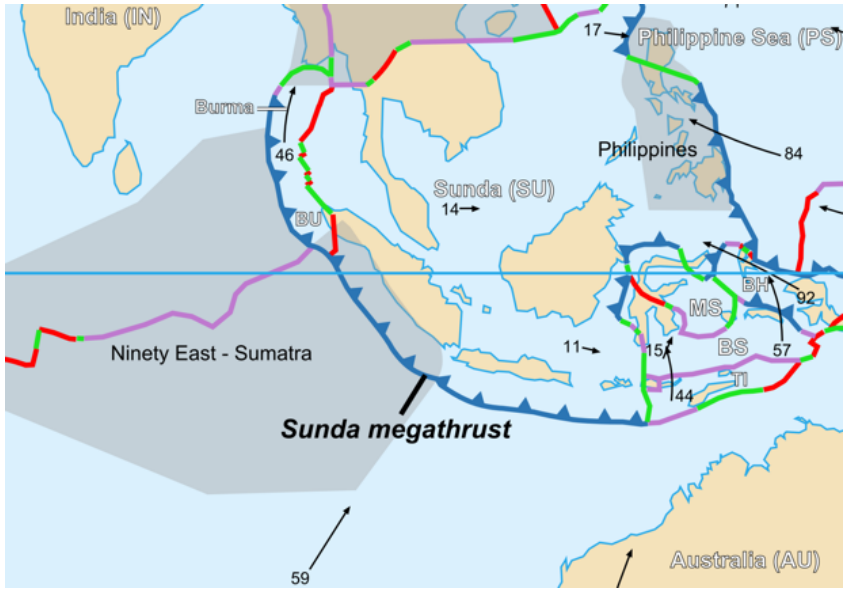
- KnL Cons
  - Cores are significantly slower than typical Xeon processors
    - 1.3GHz KnL vs. 2.5+ GHz for typical Haswell/Broadwell Xeons
    - Simpler architecture means fewer instructions processed per cycle
  - Profiling difficult and hardware not fully exposing what is needed

- Need to understand much more about our applications and their multi-phasic nature
- Deep work on both IO, memory pressure, and interprocessor comms
- Relearn how to project for the value of the processors
- Use experience to look at other emerging technologies in parallel

Experimenting with KnL characteristics

- Australian Geoscience Data Cube LANDSAT processing pipeline
  - Process a series of observations from LANDSAT8 satellite.

- NOAA Method of Splitting Tsunami (MOST) model
  - Wave propagation due to 7.5 magnitude earthquake in Sunda subduction zone

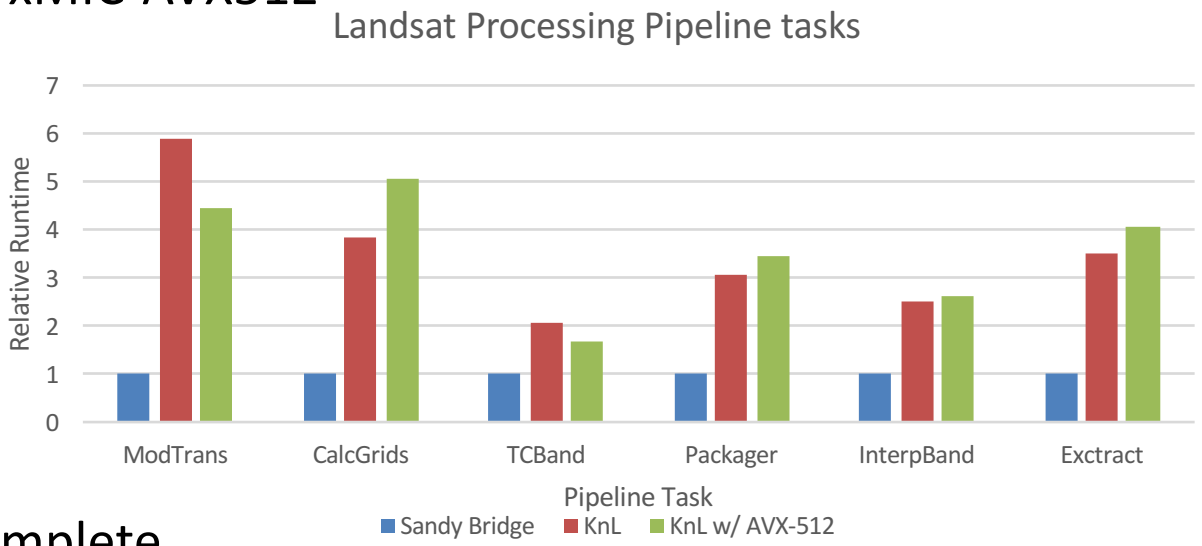


- UKMO Unified Model v10.5
  - N96 AMIP global model
  - N512 NWP global model

- These are not chosen as the best codes for KnL, but ones that were both important and that we could “quickly“ explore.

Landsat NBAR data processing pipeline: KnL vs. Sandy Bridge and 1 thread

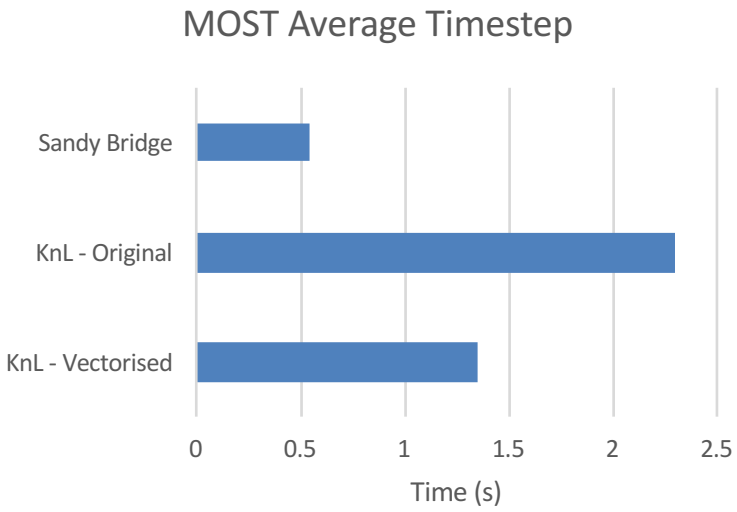
- Same executable, run on both architectures (i.e. no AVX-512 instructions)
- Separately recompiled with -xMIC-AVX512

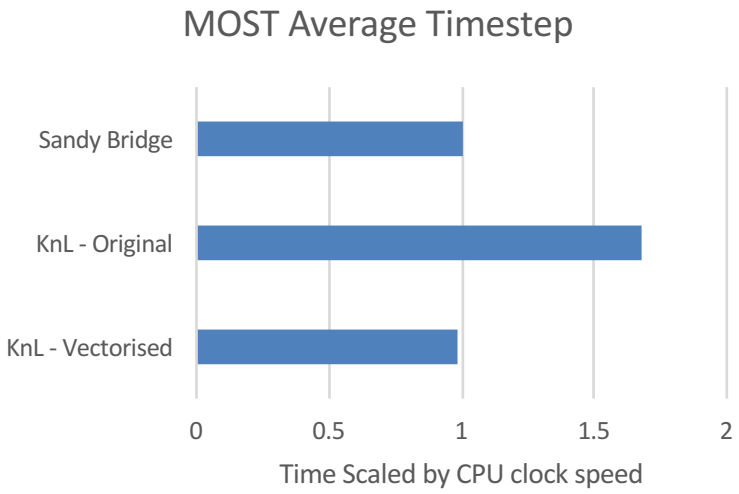


- Most tasks took longer to complete
  - LANDSAT pipeline tasks are mostly point-wise kernels or IO bound
  - Little opportunity for the compiler to vectorise
  - AVX operations run at lower clock speed on the KnL
- 'ModTrans' and 'TCBand' tasks exceptions
  - ModTrans was relatively well vectorised
  - TCBand (Terrain Correction) was converted from point-wise kernels to vector-wise kernels
  - Noted they are faster than SnB (normalised for clock speed)

NOAA MOST Tsunami Code – single threaded performance

- While MOST original code is not vectorised, but does run on KnL
- Replace key routines with vectorised versions
- Compare both raw performance and normalised by clock speed

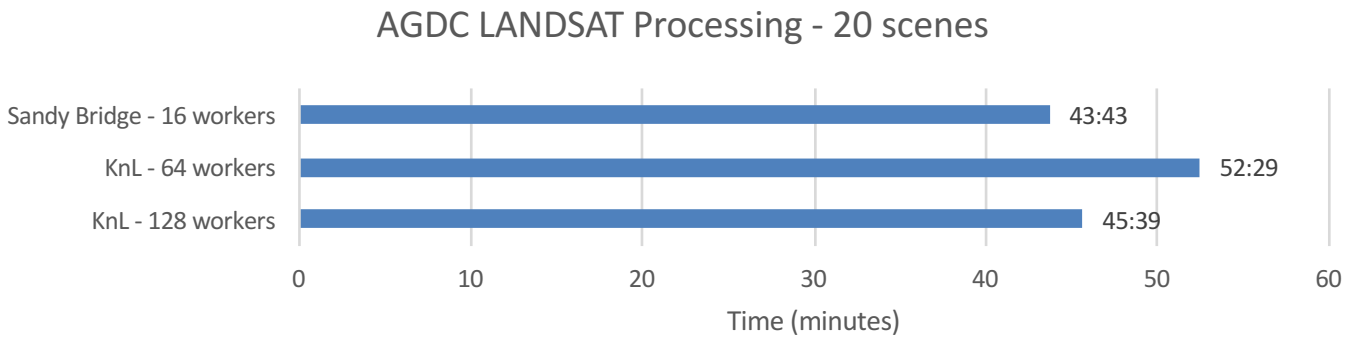




Time spent on vectorisation is the important first step to CPU performance on KnL

LANDSAT processing pipeline – comparing Node-for-Node performance

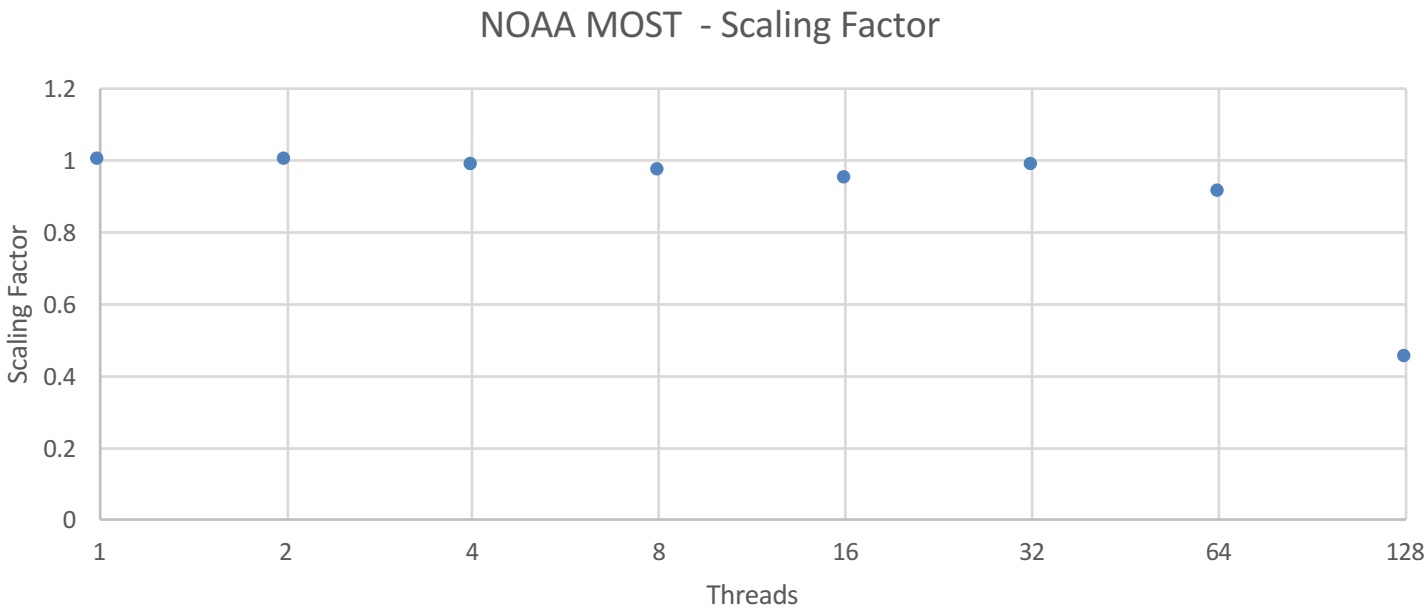
- Parallelism in AGDC LANDSAT is obtained through 'Luigi' python scheduler.
  - Task dependencies are tracked within scenes, embarrassingly parallel
  - For 20 scenes, 2620 tasks in total
- 'ideal' combination of tasks built (with and without AVX-512 instructions)
- AGDC LANDSAT Processing



- Knights Landing is slower than Sandy Bridge in this case
  - Node-for-node has competitive performance.
  - Vectorisation can yet improve
  - noted 128 tasks outperforms 64 tasks by over 20%

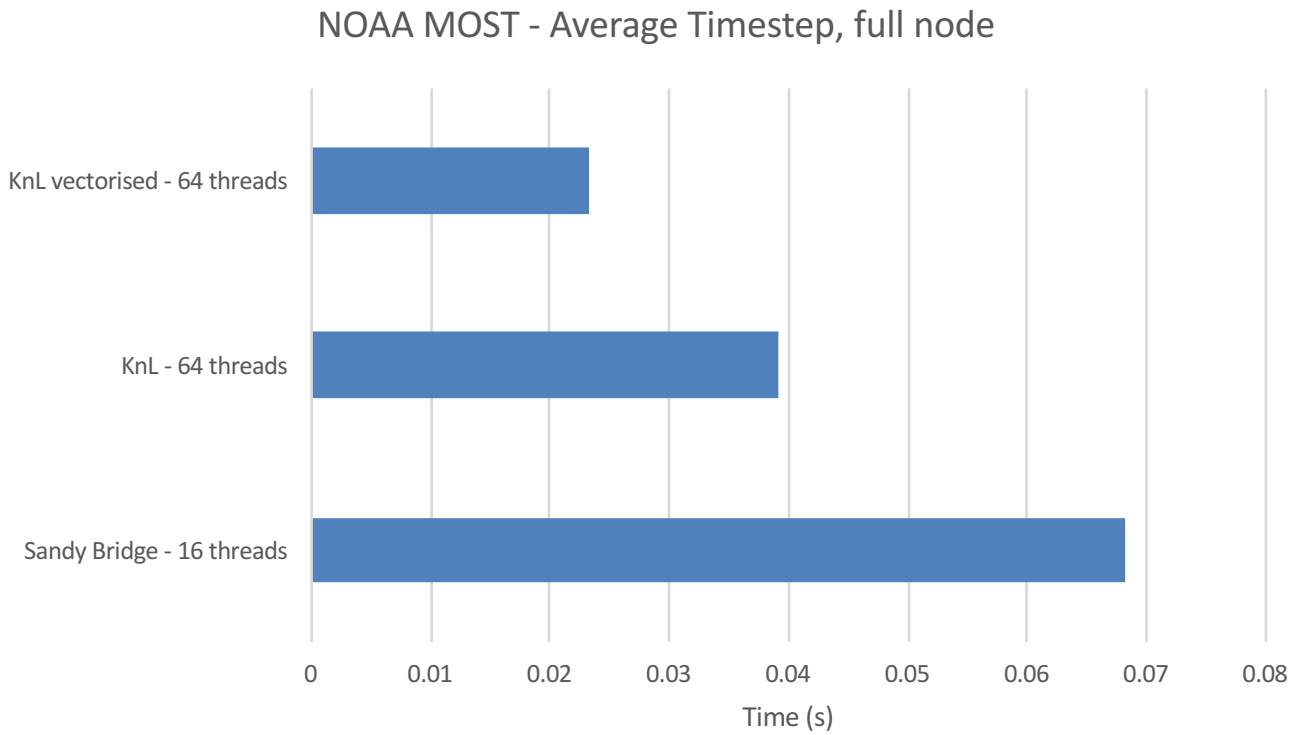
NOAA MOST – OpenMP performance on KnL

- Parallelism in NOAA MOST is obtained through OpenMP



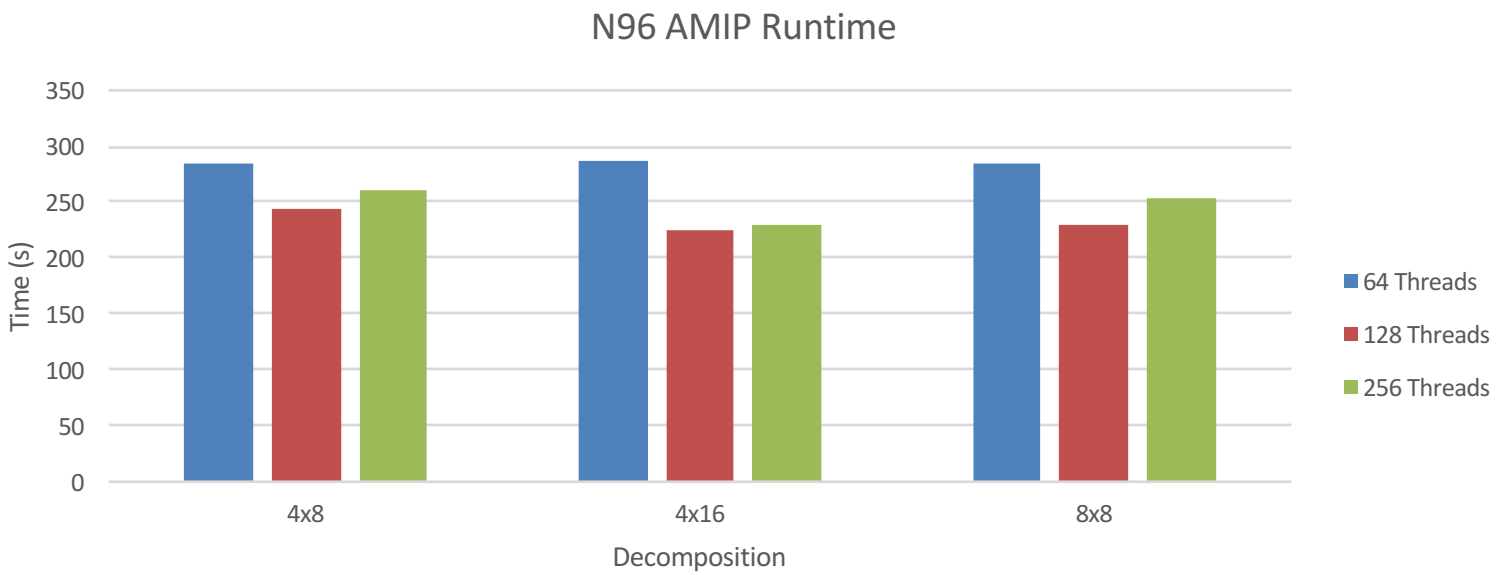
- Good scaling over already good single threaded performance
  - Over 90% efficiency going from 1 thread to fully occupying a node
- Does not benefit from oversubscription
  - Likely due to the subdomains becoming quite small at high thread counts

NOAA MOST: KnL vs Sandy Bridge node-for-node



- 3x Faster node-for-node after vectorisation.
- Note that our experiment shows MOST may be very performant on GPUs

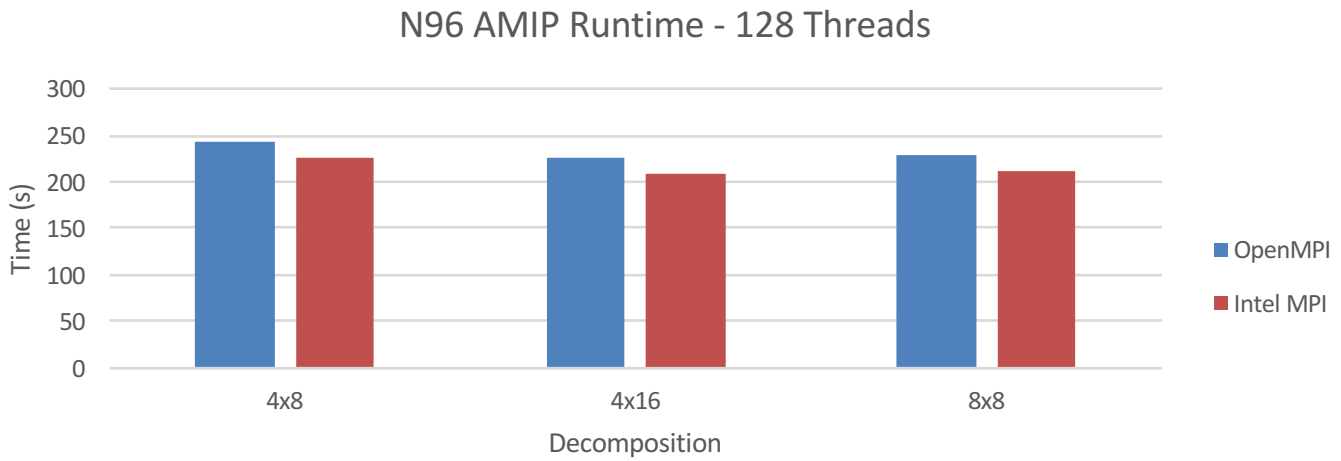
- UM (10.5) is parallelised using both MPI and OpenMP
- Initially chose AMIP N96 global model
  - useful for performance evaluation as run on a single node and no complex IO traffic
  - Find best decomposition on a single KnL node (will compare with best decomposition on a single Sandy Bridge node)



Outcomes:

- Overcommitting KnL proves beneficial to performance with the UM
- All 64 thread jobs are outperformed by 128 and 256 thread jobs

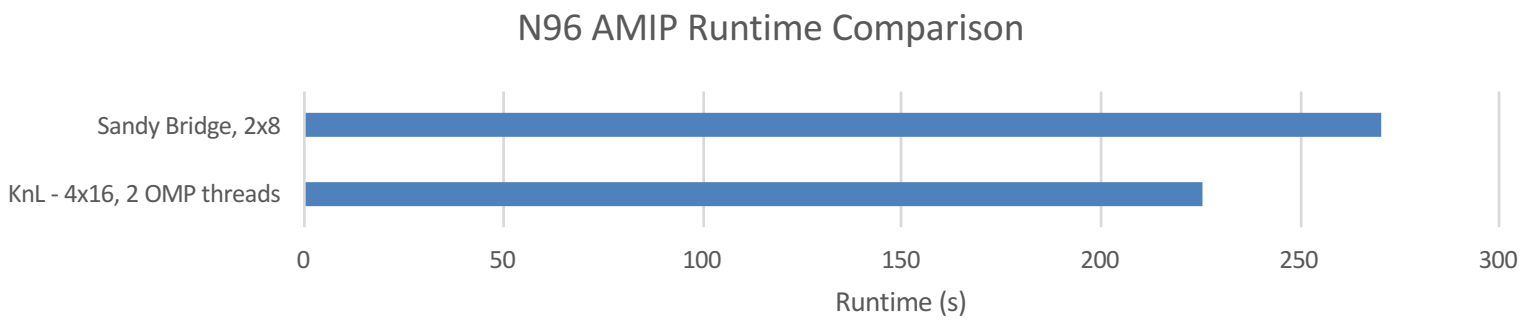
- Intel MPI consistently outperforms OpenMPI for the UM on KnL



- But Intel MPI lacks some of the fine-grained control we need
  - The ability to specify individual cores in a rank file
  - Seemingly unable to bind to 'none' – important for explicit binding with numactl
  - Can't report binding with the same detail as OpenMPI
- We used versions 15 or 16 of the Intel Fortran/C/C++ compilers
  - '-x' compiler options to enable or disable AVX-512 in order to test the effectiveness of the longer vector registers or issues
    - LANDSAT processing slows with AVX-512 enabled
    - Some instability in the UM when AVX-512 enabled

UM decomposition comparison

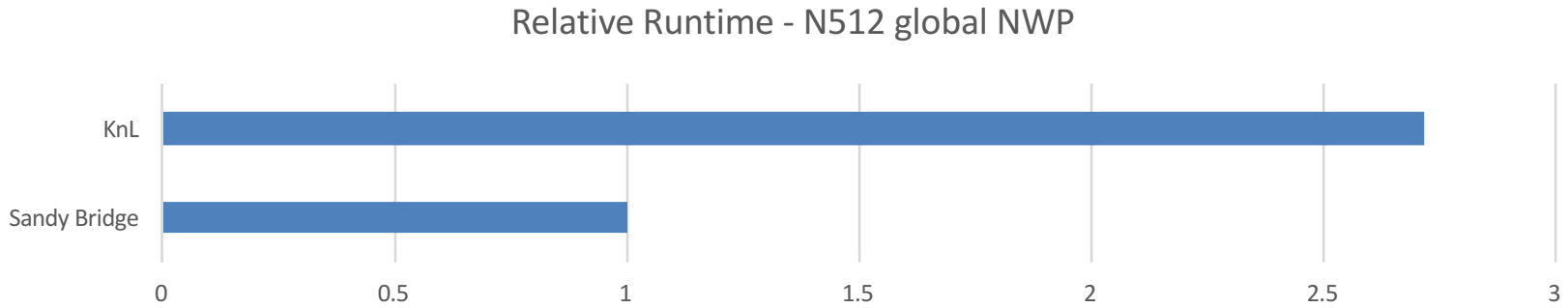
- Best performing decompositions:
  - KnL is 4x16, with 2 OpenMP threads per MPI task
  - SnB is 2x8



- About 20% faster than best decomposition on Sandy Bridge
  - Despite model input I/O stage taking 5x longer on KnL
  - larger MPI decomposition limits multinode scalability for UM on KnL
- Hybrid parallelism can help here
  - More threads per MPI task means smaller decompositions
  - Many threading improvements to come in UM10.6+

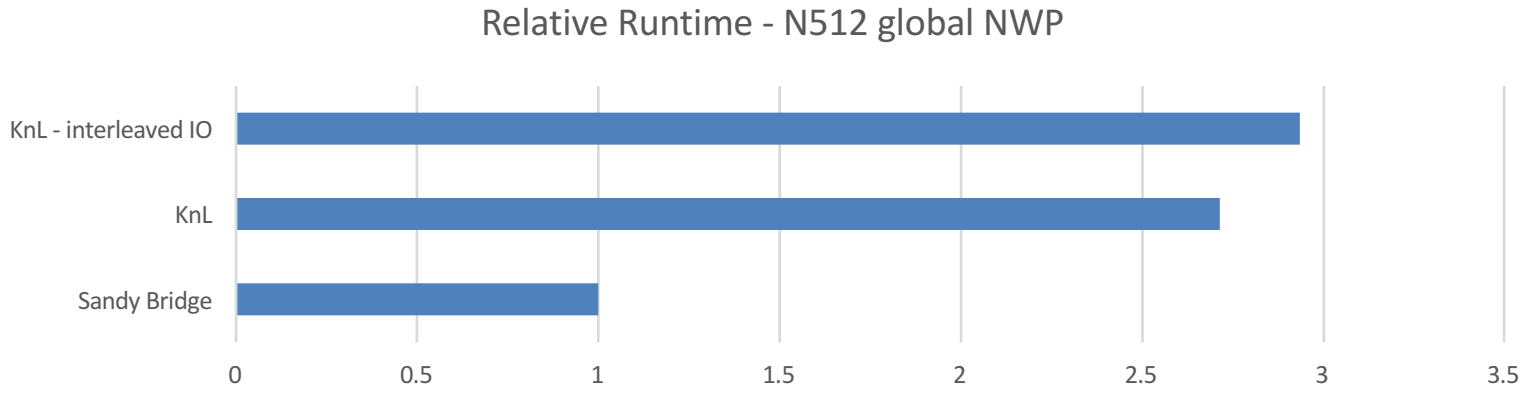
UM N512 global NWP on KnL vs SnB

- Use same decomposition: 16x64, 2 threads per MPI task, total of 2048 threads



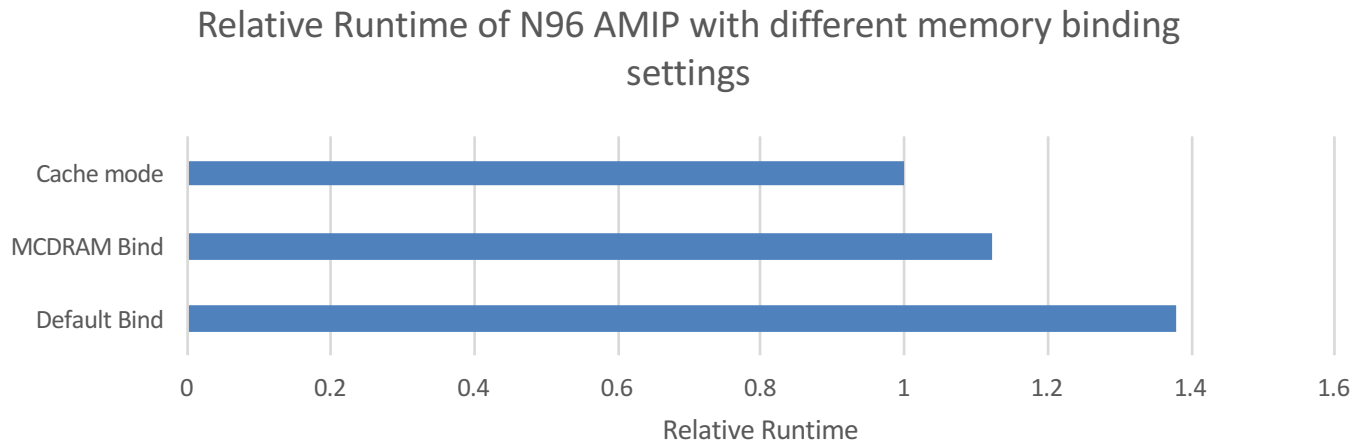
But 16 KnL nodes vs 64 SnB nodes means model uses 33% fewer node-hours on KnL

- MPI
  - Task layout is important on KnL
  - N512 job uses the UM's IO server feature where all IO tasks can run on a separate node
  - When the IO tasks are interleaved with model, runtime increases -> Need to separate IO



UM N96

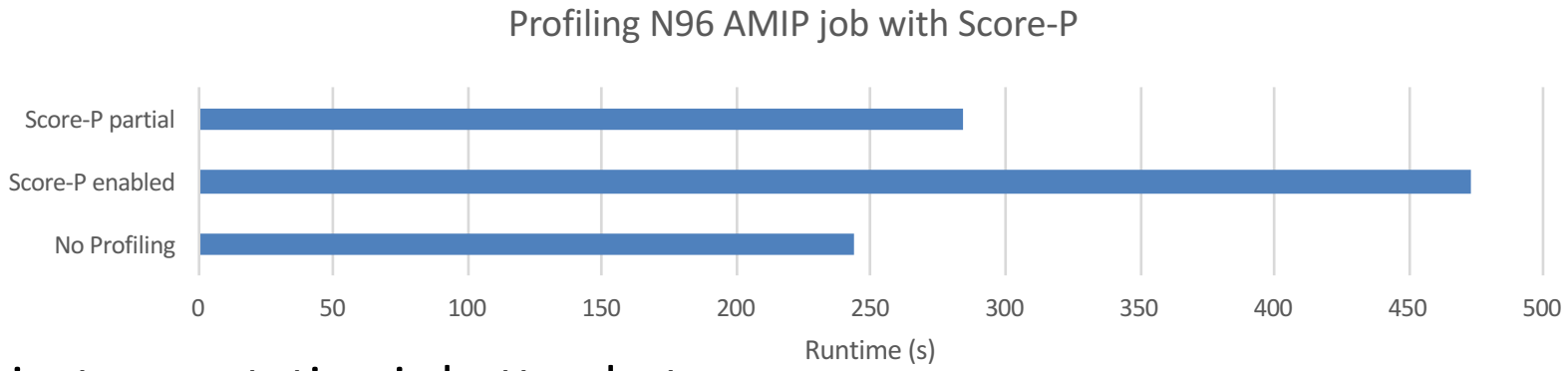
- KnL has a 2 stage memory hierarchy
  - 16GB MCDRAM on-package (Cache or Flat mode)
  - 192GB DDR4
- All our UM tests shown so far have been in ‘cache’ mode.
  - N96 AMIP global occupies just over 16GB RSS when run in a 4x16 decomposition
  - Can additional performance be extracted in ‘flat’ mode?



- No MPI distribution performs the binding correctly, so launch MPI processes using numactl
- Both default binding (DDR4) and MCDRAM binding are slower than cache mode.
- Loss of performance when run on DDR4 implies that the UM is still memory bound, even on slow KnL cores.

Profiling with Score-P

- Score-P is an instrumenting profiler
- Issues – instrumenting on KnL is very costly
  - Entering and exiting instrumented areas seems to cost a fixed number of cycles
  - Cycles take much longer on a KnL
- Compare with limited instrumentation to key ‘control’ subroutines
  - Allows identification of key code areas (e.g. convection, radiation etc.), but nothing within those areas



- Partial instrumentation is better, but
  - if an OpenMP parallel section is not instrumented, time spent in threads other than the main thread is lost
  - Can’t analyse thread efficiency this way

Profiling through sampling – experience so far

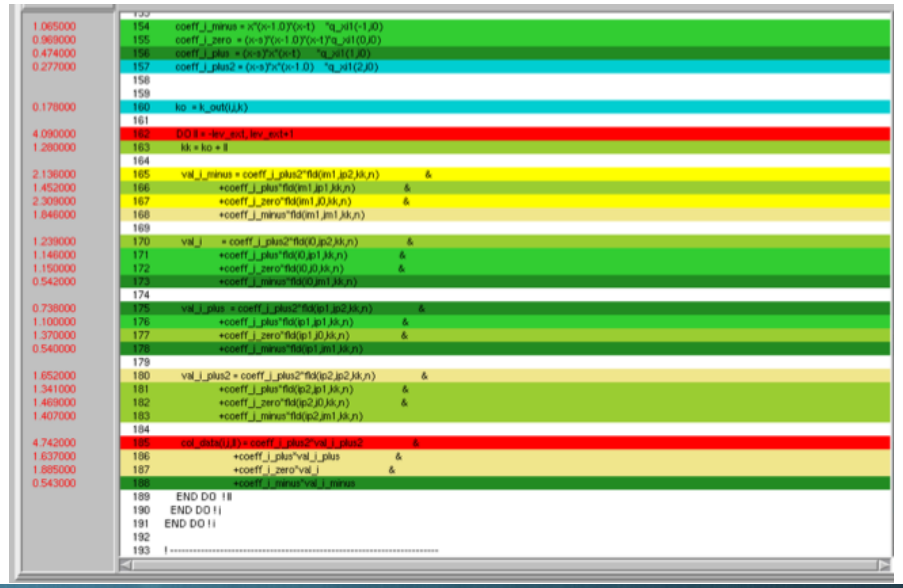
- Sampling profiling can be used instead
  - OpenSpeedShop
  - HPCToolkit
  - Intel Vtune (can't be used with OpenMPI)

OpenSpeedShop

- Profiling UM with OpenSpeedShop produces negligible overhead
- Potential issue with sampling rate, but in practise good agreement

Intel Vtune

- Around 10% overhead in MOST with Intel VTune
- Some features are not available on KnL (e.g. Advanced Hotspots)



Summary of Knights Landing experience so far

- KnL's look like promising technology and worth more investigation
  - Well vectorised workloads are essential to performance on KnL
    - Unvectorised workloads see KnL outperformed by node-for-node by Sandy Bridge
    - Well vectorised workloads run significantly faster
  - Nodes are more energy efficient.
  - Code changes are more generally useful, so not specifically targeted for KnL.
  - Hybrid Parallelism and reducing MPI task management is needed for large-scale jobs
- Data-intensive IO needs more attention for performance – especially parallel I/O
  - Parallel I/O available through NetCDF and HDF5
- Profiling applications is still difficult
  - Instrumented profilers can't be used until the overhead can be reduced
  - Sampling profilers may be missing events
  - Some missing functionality
- Helpful for understanding more details of the behaviour of codes
- How does it compare to GPU and other emerging chip technologies?