Image credit: Alex Cherney

Ingest pipeline for ASKAP

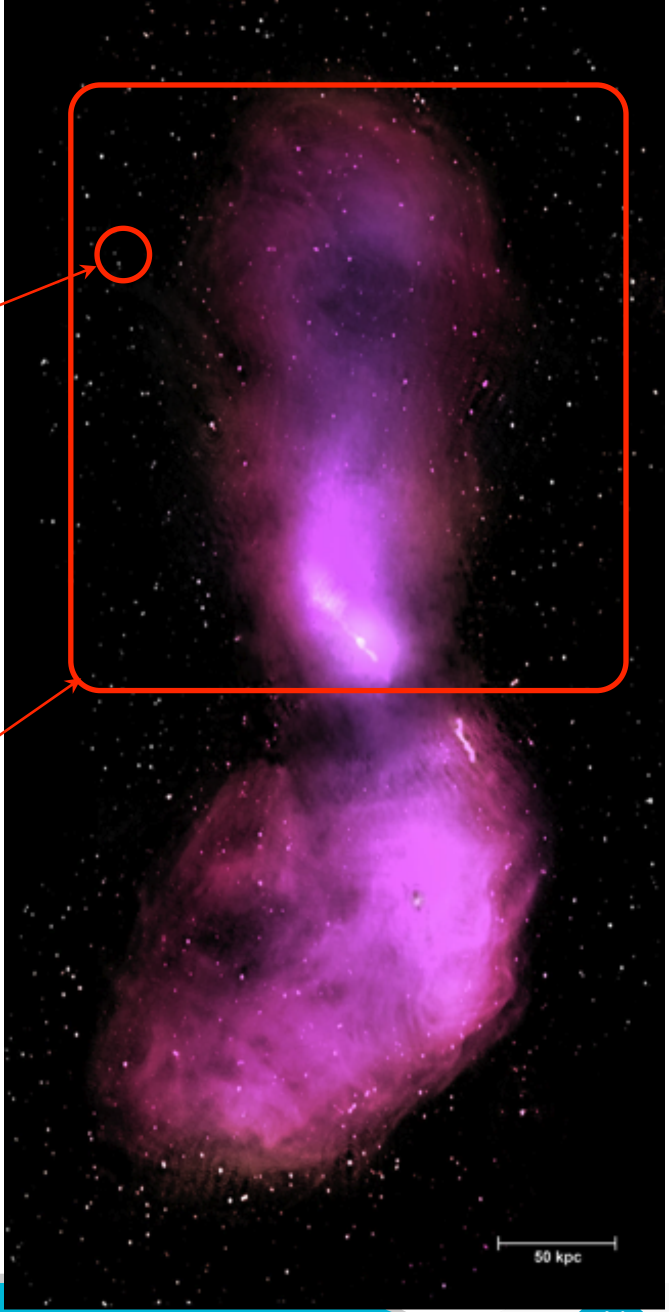
Max Voronkov

Computing for SKA, Auckland, NZ – 14 Feb 2019

ASKAP: Wide Field of View

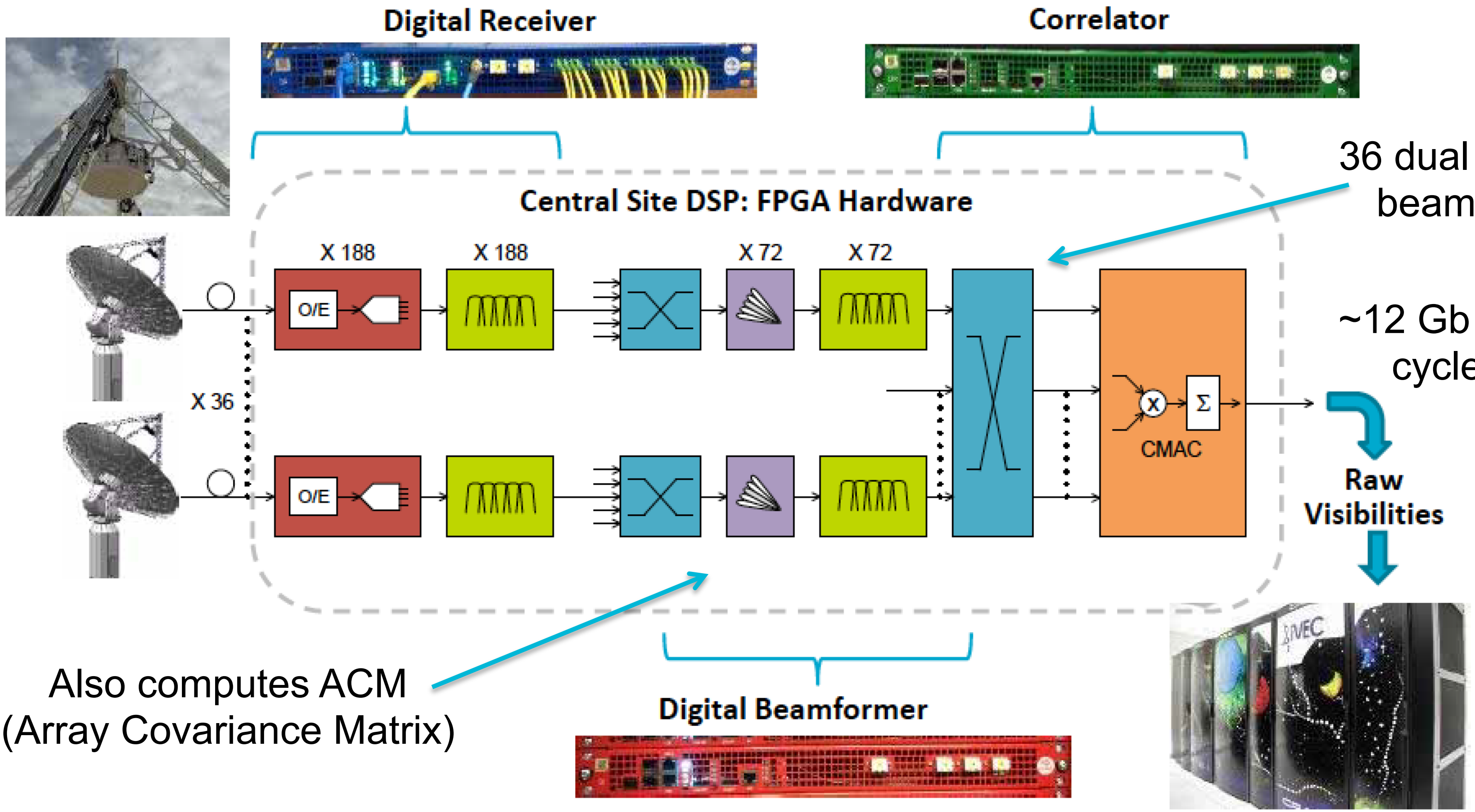
- Required 1200 hours observing with the Australia Telescope Compact Array

- ASKAP will take about 10 minutes



Cen A image credit: Tim Cornwell / Ilana Feain

ASKAP – system architecture (I)



36 dual pol beams

~12 Gb per cycle

Also computes ACM (Array Covariance Matrix)

Slide credit: John Tuthill

Plans vs. reality

- Initial design aimed at automatic processing of data
  - Largely due to inability to store data (run out of current storage in 1 week)
  - One size fits all approach with on-the-fly calibration – just 1 read of dataset
- Traditional processing model for now
  - User preference, more storage/better algorithms, lesser push for commensality
- Commissioning, staged deployment and support of BETA
  - Additional requirements never envisaged at the design stage
  - Lots of data inconsistencies in various ways, sometimes transient
  - Intermediate s/w solutions allowed to get science faster

The role of ingest

- Prepare/re-format data for calibration & imaging pipelines
  - Synchronisation is the main part (especially if something doesn't work right)
  - Standard formats allow us to have more generic imaging & calibration tools
- Some processing which has to be done on-the-fly to reduce I/O
- Crucial part of the online calibration loop (when/if we do it)

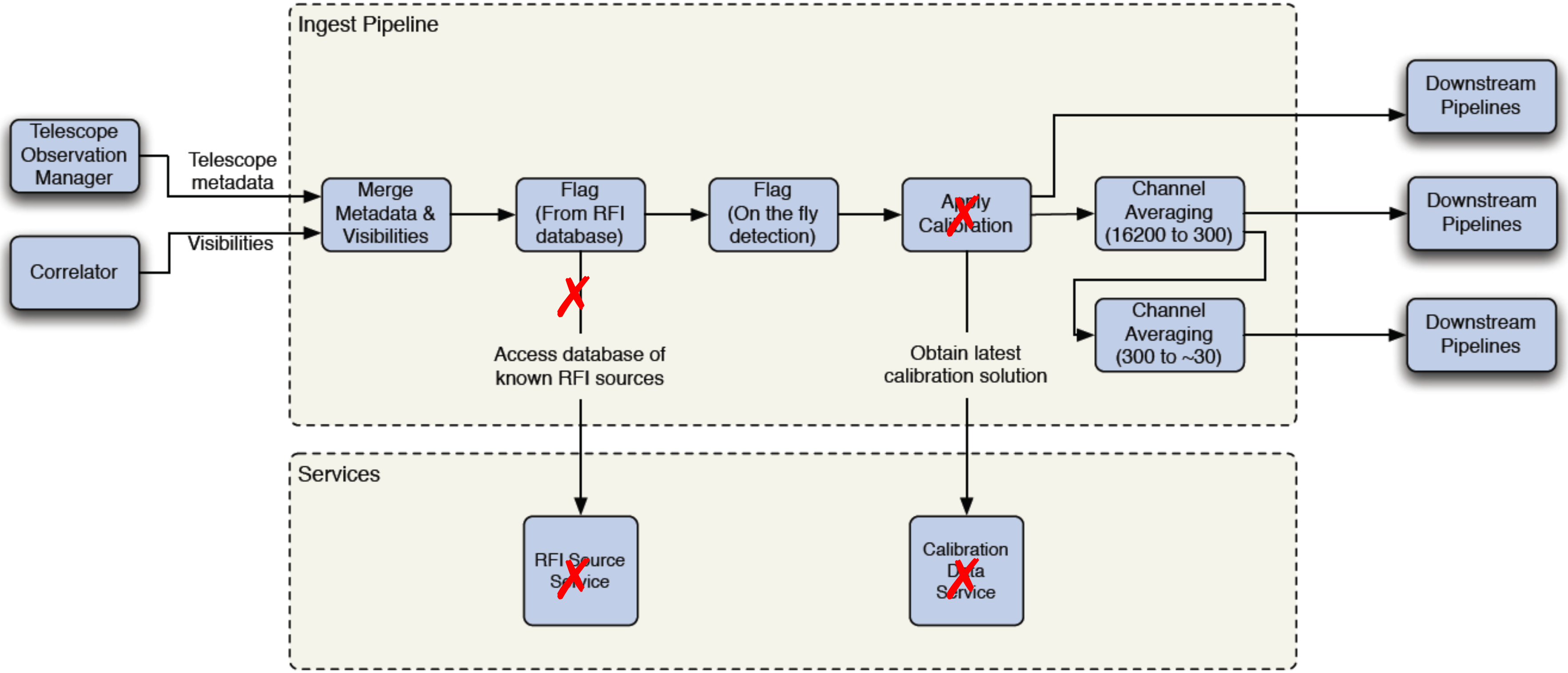
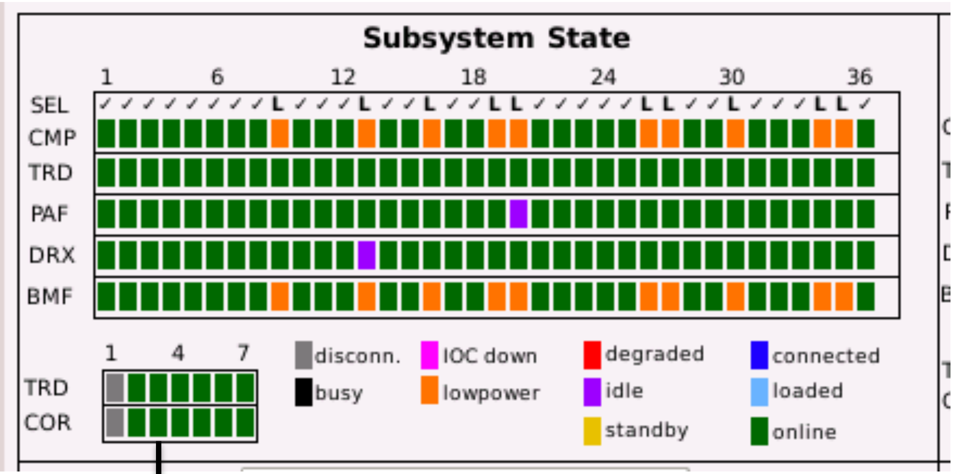

Ingest Pipeline
Telescope Observation Manager
Telescope metadata
Correlator
Visibilities
Merge Metadata & Visibilities
Flag (From RFI database)
Flag (On the fly detection)
Apply Calibration
Channel Averaging (16200 to 300)
Channel Averaging (300 to ~30)
Downstream Pipelines
Downstream Pipelines
Downstream Pipelines
Access database of known RFI sources
Obtain latest calibration solution
Services
RFI Source Service
Calibration Data Service
✗
✗
✗
✗

Diagram from the SW-0020 document (http://www.atnf.csiro.au/projects/askap/ASKAP-SW-0020.pdf)

ASKAP - system architecture (II)



Subsystem State

	1					6					12						18
SEL	✓	✓	✓	✓	✓	✓	✓	✓	L	✓	✓	✓	L	✓	✓	L	✓	✓
CMP	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟧	🟩	🟩	🟩	🟧	🟩	🟩	🟧	🟩	🟩
TRD	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
PAF	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
DRX	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟪	🟩	🟩	🟩	🟩	🟩
BMF	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟧	🟩	🟩	🟩	🟧	🟩	🟩	🟧	🟩	🟩

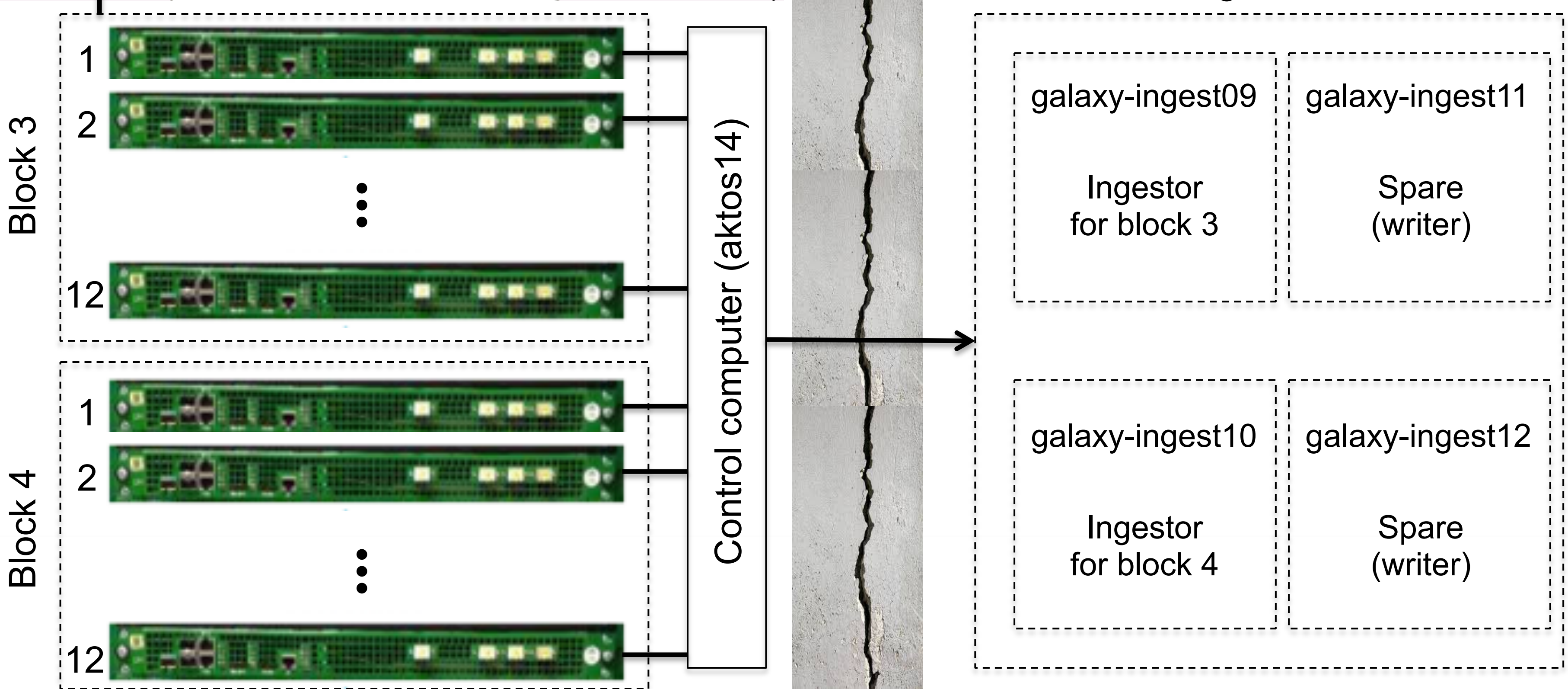
						24						30						36
SEL	L	L	✓	✓	✓	✓	✓	L	L	✓	✓	L	✓	✓	✓	L	L	✓
CMP	🟧	🟧	🟩	🟩	🟩	🟩	🟩	🟧	🟧	🟩	🟩	🟧	🟩	🟩	🟩	🟧	🟧	🟩
TRD	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
PAF	🟩	🟪	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
DRX	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
BMF	🟧	🟧	🟩	🟩	🟩	🟩	🟩	🟧	🟧	🟩	🟩	🟧	🟩	🟩	🟩	🟧	🟧	🟩

	1			4			7
TRD	▩	🟩	🟩	🟩	🟩	🟩	🟩
COR	▩	🟩	🟩	🟩	🟩	🟩	🟩

▩ disconn.	🟪 IOC down	🟥 degraded	🟦 connected
⬛ busy	🟧 lowpower	🟪 idle	🟦 loaded
		🟨 standby	🟩 online

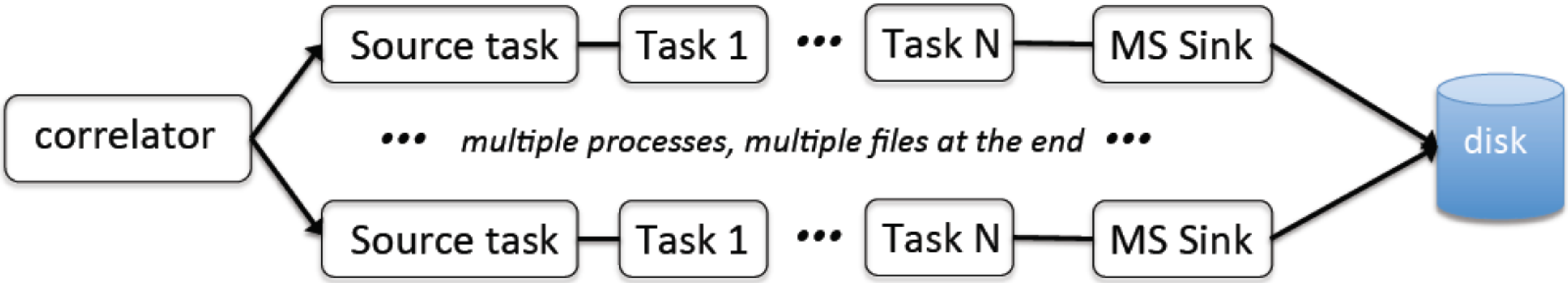
Ingest cluster @ Pawsey

- 16 nodes, 2 sockets per node
- 8 cores CPUs, 64 Gb of RAM per node
- Shared Lustre storage with 30 Gb/s peak I/O performance



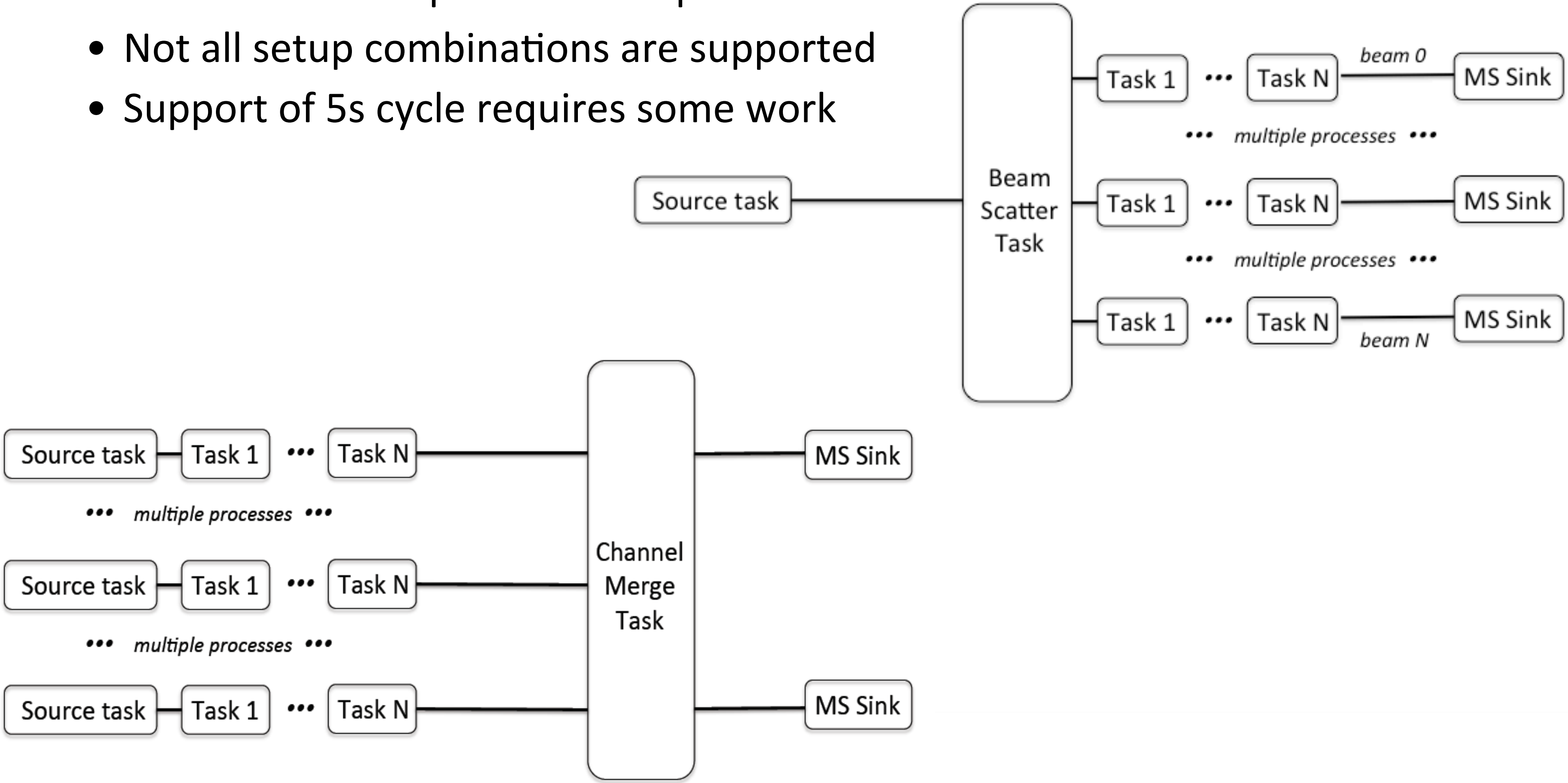
Tasks and data streams

- Ingest can be viewed as a chain of "processing" tasks
  - Configurable (via **tasklist** Facility Control Manager parameter)
  - Processing chain should always start with a source task (two options available)
  - Any task except source can occur in the chain any number of times (with the same or different parameters)
  - Sink task doesn't have to be the last (or even doesn't have to be present)
- Both parallel and serial modes are supported
  - Source tasks are rank-aware and would listen different UDP ports
  - Some tasks (merging, splitting) are specific to the distributed mode
  - Service ranks (i.e. those which do not run source task) are now also supported

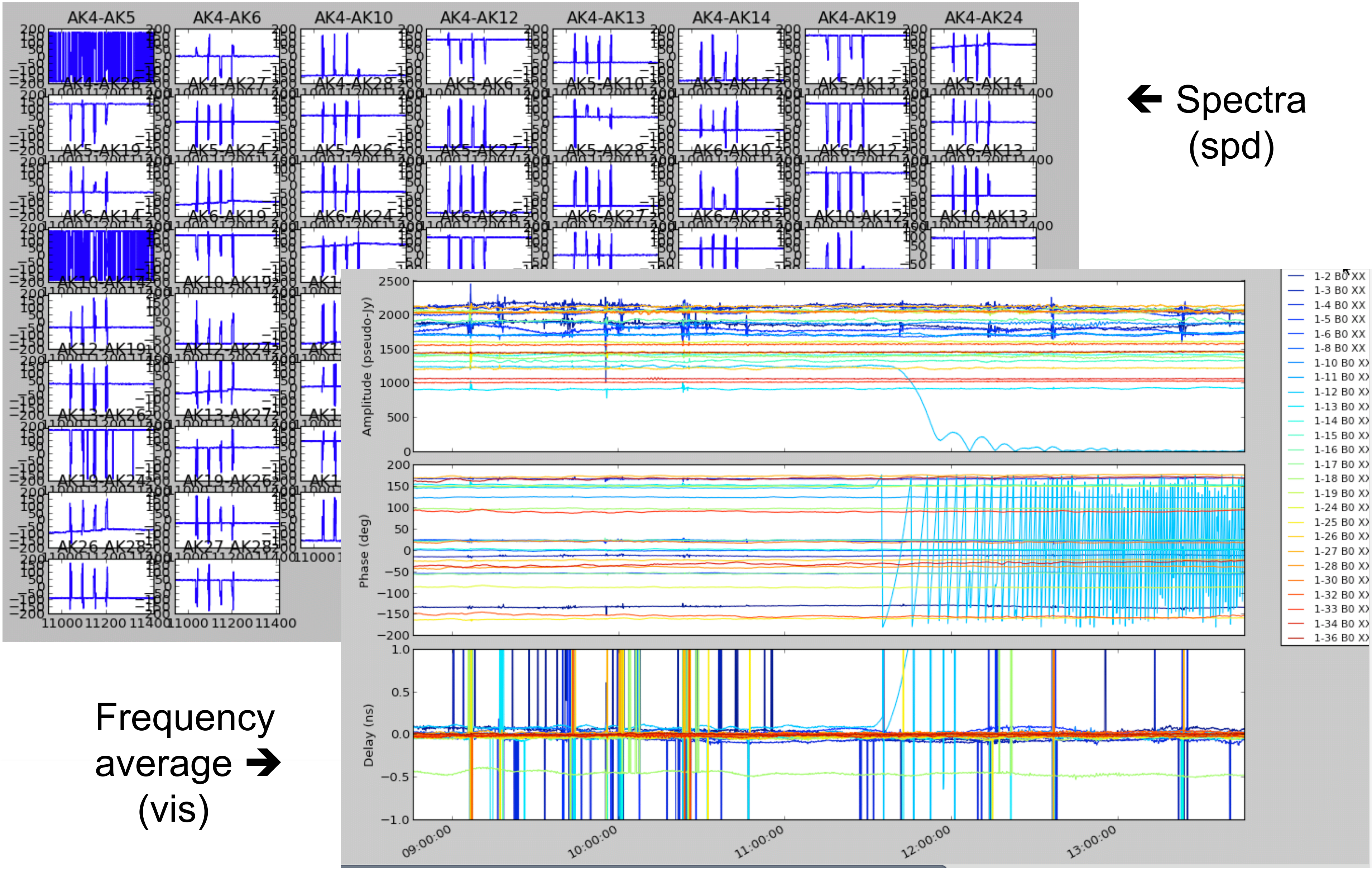


Splitting and merging data streams

- Each stream can be active or deactivated (behind the scene)
  - Splitting/merging tasks can change the state + can use service ranks
  - These tasks are specific to the parallel mode
  - Not all setup combinations are supported
  - Support of 5s cycle requires some work



Data monitoring via TCPSink/vispublisher

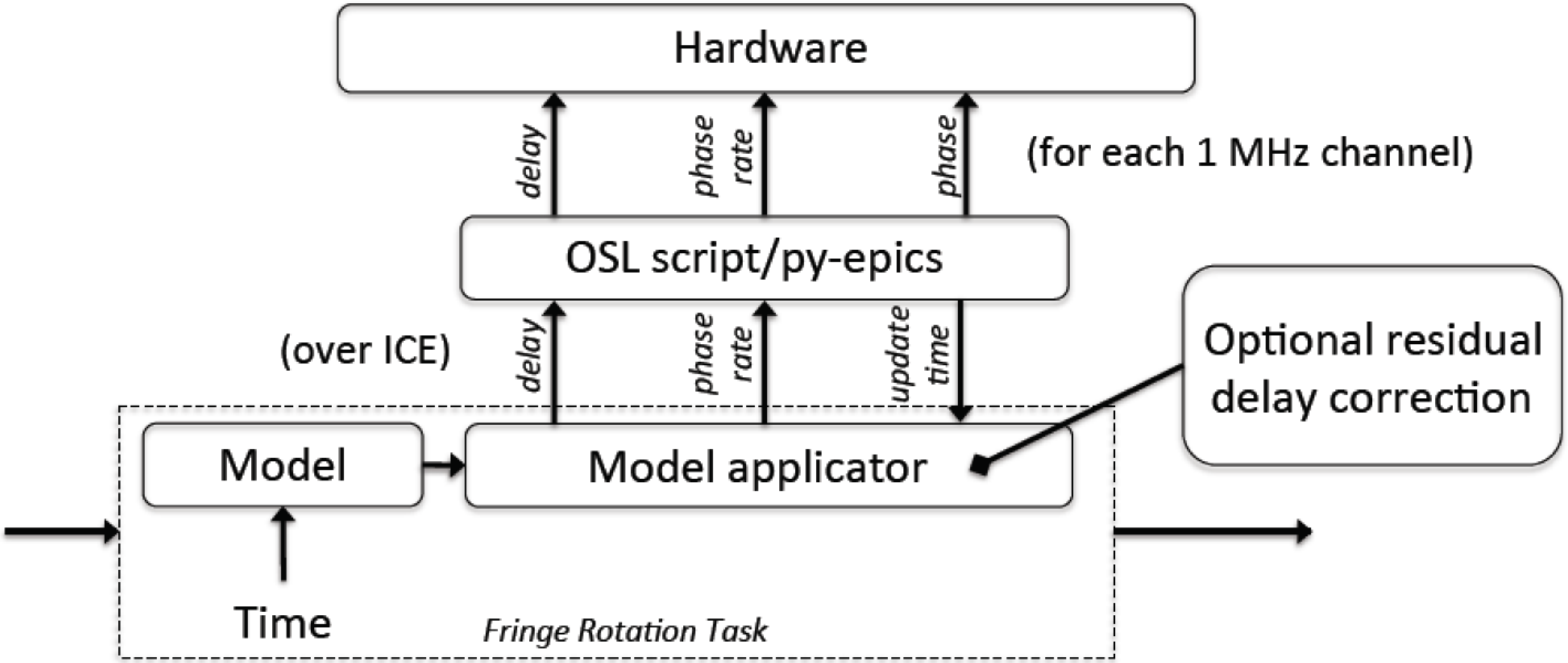


← Spectra (spd)

Frequency average → (vis)

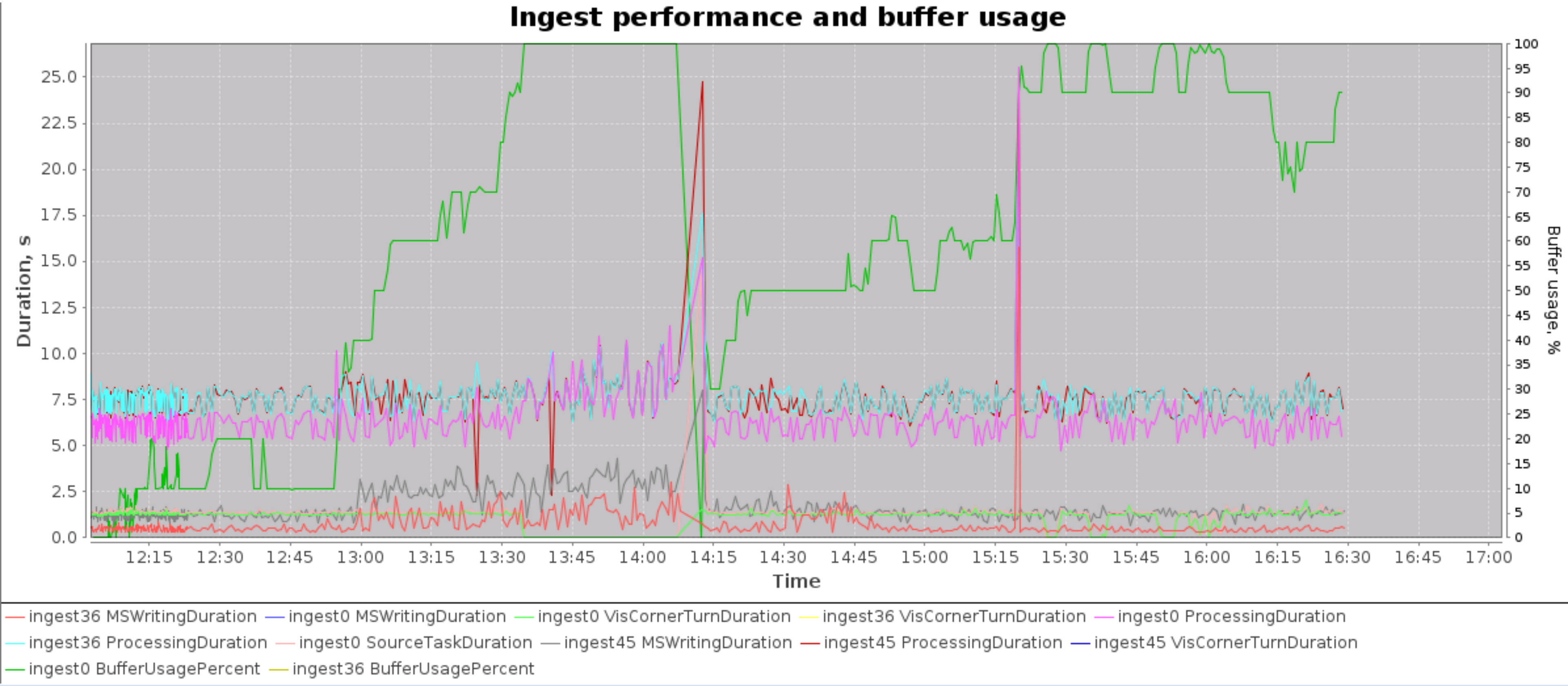
Commissioning experiments

- Intermediate mode to control fringe rotator from ingest
  - Grew up from the need to support early BETA array, extended to early ASKAP
  - Investigate relative timing + well-controlled baseline system
  - Early science done a few years earlier



Performance monitoring

- Many performance metrics are published every cycle via Ice
  - Figures for representative ranks can be monitored (monica or Grafana)
  - As many tasks have implicit synchronisation barriers, interpretation requires knowledge of the system architecture and configuration of ingest pipeline
  - Buffer usage and writing times are the most straightforward metrics



Various performance lessons

- Logical vs. physical isolation of ingest
  - Writing data to shared lustre filesystem (with dedicated OST/metadata nodes)
  - It took surprisingly long time to get to an acceptable level
    - May be not over yet – we are not pushing the system the same way now
    - Our own processes/processing can also affect performance
    - If it locks up, it happens inside 3rd party libraries, so can't really time out
- Real-world astronomy data formats vs. idealistic I/O benchmarks
- Implicit barriers
  - Data sent in staggered fashion (if you need more than one chunk, you wait)
  - Metadata are not available until the end of the cycle
  - Logging at scale may be non-trivial, especially for a synchronous system
  - Consistency cross-checks may require additional communication
- Not in the regime how HPC is typically used – watch out for bugs
  - It matters where each rank goes

What wasn’t in design, but in use now

- On-the-fly averaging as a separate mode
  - Instead of full-resolution mode, not in addition to
  - This is largely to save disk space / for projects which don’t need full resolution
  - Prior flagging is essential
- Per-beam partitioning of the data (one beam in one MS)
  - Single beam mode is a particular case
- Flexible partitioning in frequency (merge/split)
- Real-time monitoring of data after ingest (i.e. vis/spd)
- Various data consistency cross-checks
- Flexible configuration options
  - Zoom modes (user-controlled)
  - Adding correlator hardware (FCM controlled, but requires expert knowledge)
  - Changing antennas included in the array
    - We used to have the main and commissioning array

Summary

- Ingest is a flexible adapter s/w between correlator and processing
  - Allows us to debug/test synthesis jobs from a standard MS
  - Synchronise parallel data and metadata streams
  - Aggregate/split data as required
  - On-the-fly flagging, if necessary
  - On-the-fly calibration application in the future
  - Optional on-the-fly averaging in frequency
  - Interface to on-the-fly data monitoring (vispublisher -> vis and spd)
- Invaluable tool for commissioning
  - Detect oddities in data stream
  - Non-standard experiments
  - Intermediate solutions (e.g. fringe rotation) to get science results faster

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

Thank you

Astronomy and Space Science
Max Voronkov
Senior Research Scientist

t +61 2 9372 4427
e maxim.voronkov@csiro.au
w www.narrabri.atnf.csiro.au/people/Maxim.Voronkov