High Performance Computing Approaches for Processing Hydrographic Data
The need

• Large volumes of information-rich point data are becoming increasingly available

• Greater volumes of data can mean greater detail – but regional-scale mapping requires large amounts of computing power

• Centralisation can be difficult with multiple providers, constant updates, different submission formats etc.

• But transfer speeds and costs can also be a bottleneck

• The ability to process large quantities of information in a distributed manner is needed -> High Performance Computing
Survey to Service

Marine survey

Airborne survey

Raw archive

Processing (cleaning)

Query and analysis

APIs and web services

Analysis Ready Data

• Correct once use many times
• Reduce domain-specific changes
• Correct up to the point before products ‘branch’
• Self-describing data
Building footprints for raw data

http://marine.ga.gov.au
Concurrent processing at NCI

Python

MB System

Days to minutes
Apache Spark

• Provides a means of performing scalable computing across multiple (possibly virtual) machines

• Can read data distributed across machines and platforms (e.g. reading directly from S3 buckets, databases, Lustre, HDFS)

• Can be coded using Python, R, Java or Scala, and can also run SQL (database) commands
Bathymetry processing with Spark

http://bit.ly/2wUwuC0

```scala
val s3 = spark.read.format("csv").load("s3a://test-bathymetry/*")
```
Bathymetry processing with Spark

http://bit.ly/2wUwuC0

ESRI GeoTools for Hadoop (and Spark!)
Bathymetry processing with Spark

http://bit.ly/2wUwuC0

Approximately 45 minutes for >4.6 billion (cleaned) points (at 150m) using 8 m3.xlarge nodes, approximately AUD$0.48 Using AWS.

(previously > 8 hours)
Moving further ahead

If you have questions:

Johnathan Kool (AAD)
johnathan.kool@aad.gov.au

Kim Picard (GA)
kim.picard@ga.gov.au