CARTA Data Flow Prototyping

Exploring a Data Flow Model for the CARTA Back-end System

What is CARTA?

- The **Cube Analysis and Rendering Tool for Astronomy (CARTA)** is designed to visualise and analyse large scale astronomical imagery.
- The CARTA system consists of a **front-end web client** which receives processed information from the **back-end server implemented in multi-threaded C++** .

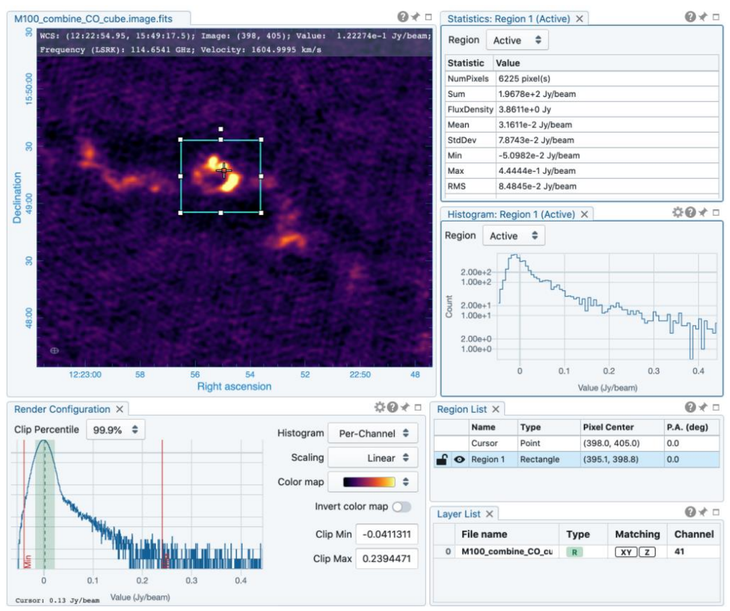

Figure 1: The CARTA interface

Data Flow Who?

- Data flow architecture differs from the traditional von Neumann architecture in that program flow is governed by the **availability of the instruction input data**.
- Modern HPC systems are expected to process data quicker to cope with increasingly large data sets and shifting to a data flow model can be a **sustainable way forward** for these systems.

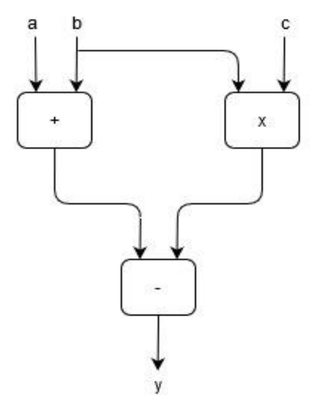

Figure 2: Data flow model of y = (a + b) - (b x c)

Objectives

Using the **Python Dask data flow environment**, a data flow model is explored for the CARTA back-end system to investigate the implications of such a change.

Design

Zainab subjected the CARTA back-end to an **architectural re-design** following systems engineering best practices.

Implementation

Dylan implemented a set of **prototype back-end components** to gauge their performance and scalability.

To Data Flow or Not?

- We find that the shift to the Dask data flow environment for the CARTA back end may be a **worthwhile venture**.

- ☑ Better Scalability
- ☑ Better Performance
- ☑ Server Modularity
- ☑ Code Simplicity

- The proposed design revealed that using Python Dask leads to a **simpler** and easier to follow code base than the C++ system with better **server modularity** and potential for **heterogeneity**.
- The data flow model allows for **scaling** out to large distributed clusters of machines with **minimal input** from the programmer.
- Performance testing on the prototype components revealed that the data flow model would **perform significantly better** in the best case, and about the same in the worst case.

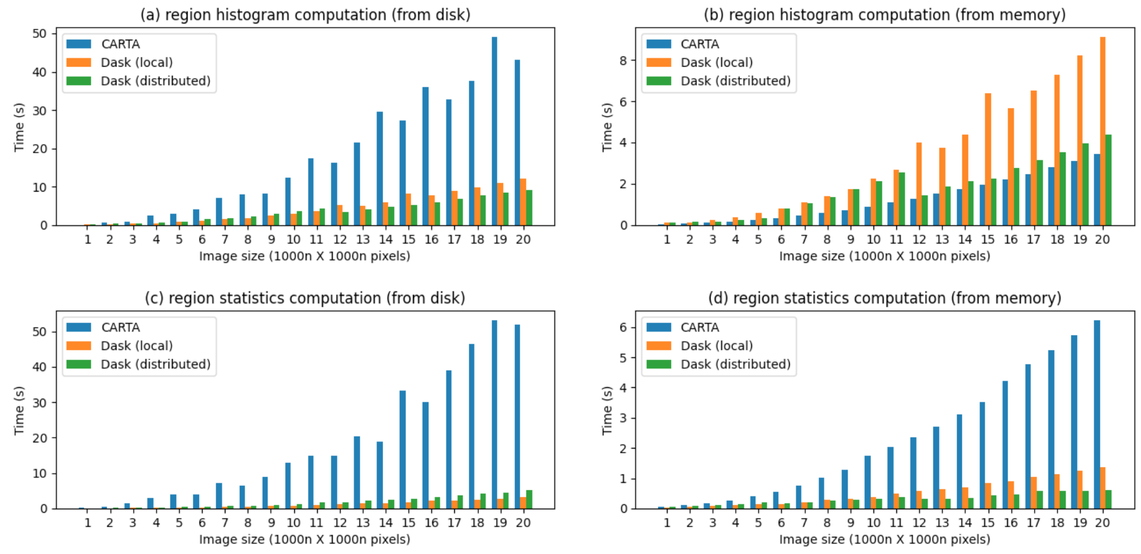

Figure 3: Performance test results showing compute times for CARTA and Dask (lower is better)

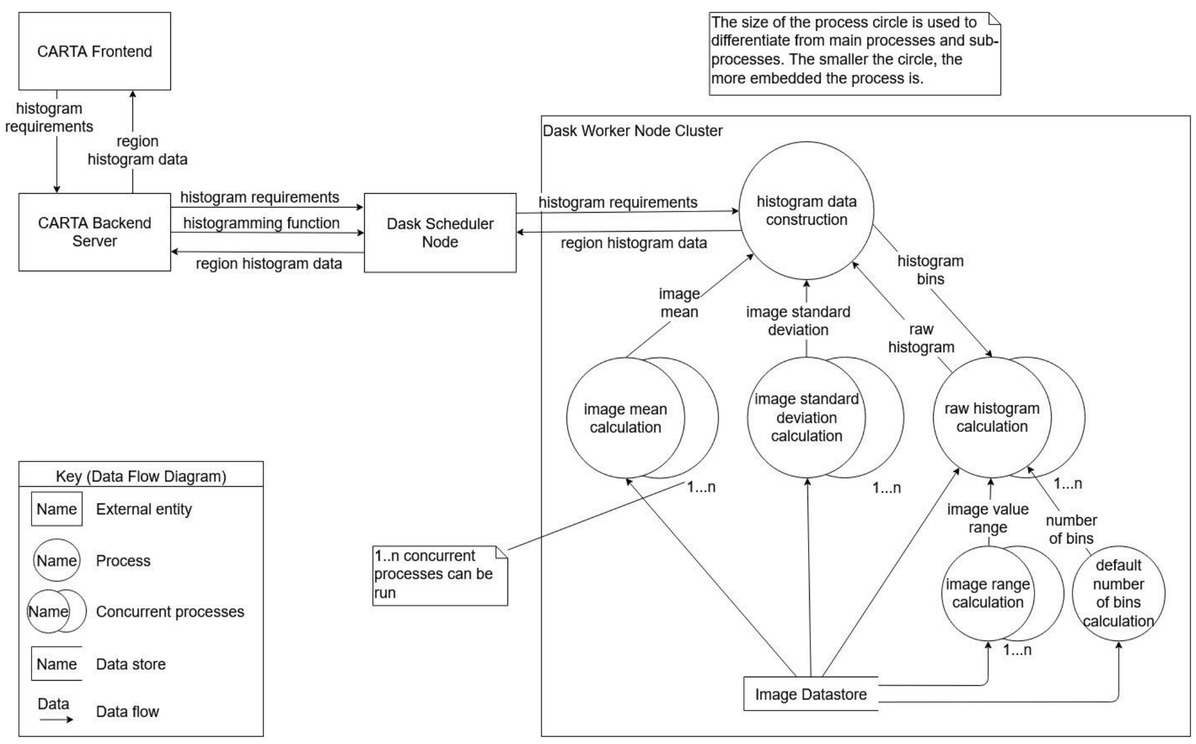

Figure 4: Data flow diagram showing how the Dask Scheduler handles distributing computation to the Worker Node cluster

Acknowledgements

We acknowledge funding from the National Research Foundation (NRF), the use of Ilifu cloud computing resources, and guidance from the Inter-University Institute for Data-Intensive Astronomy (IDIA) academics and engineers.

University of Cape Town
Computer Science Dept.
Email: dept@cs.uct.ac.za
Tel: 021 650 2663

Team Members
Zainab Adjiet
Dylan Fouche

Supervisor
Rob Simmonds

Co-Supervisors
Adrianna Pinska
Kerchil Kirkham