Unleashing the Power of Distributed CPU/GPU Architectures:
Massive Astronomical Data Analysis and Visualization case study

Amr H. Hassan, Centre for Astrophysics and Supercomputing, Swinburne University of Technology

Christopher J. Fluke, Centre for Astrophysics and Supercomputing, Swinburne University of Technology

David G. Barnes, Monash e-Research Centre, Monash University

Astronomy In the Petascale Era

Simulation and Observation Facilities

Australian **SKA** **P**athfinder (ASKAP)

3 to 12 TB/ Day

The **L**arge **S**ynoptic **S**urvey **T**elescope (LSST)

30 TB/ Day

Swinburne gStar GPU-Supercomputer

130 TFLOP/S ➔ ?

Square Kilometre Array (SKA)

? PB/ Day

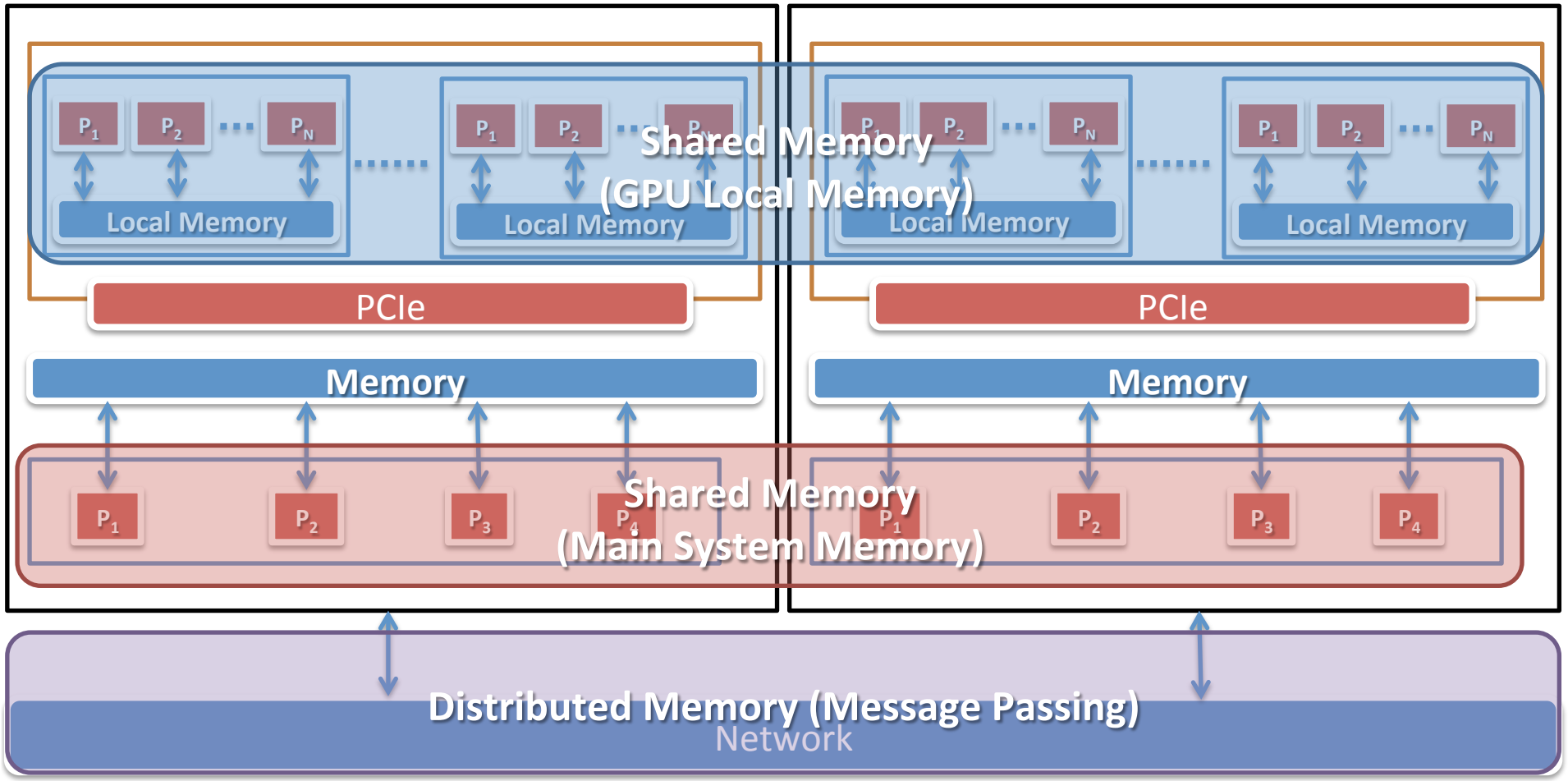
Astronomy In the Petascale Era

Computer-assisted data analysis and visualization

1. Handle data larger-than-memory.
2. Launch autonomous distributed analysis and visualization jobs
3. Employ computational accelerators (GPU and FPGA)
4. Effectively deal with distributed and remote data storage facilities.
5. Minimize the user intervention in the analysis processes.
6. Intelligent schedulers able to minimize data movement and memory usage
7. Adaptive user interfaces to interact and analyze remotely stored data

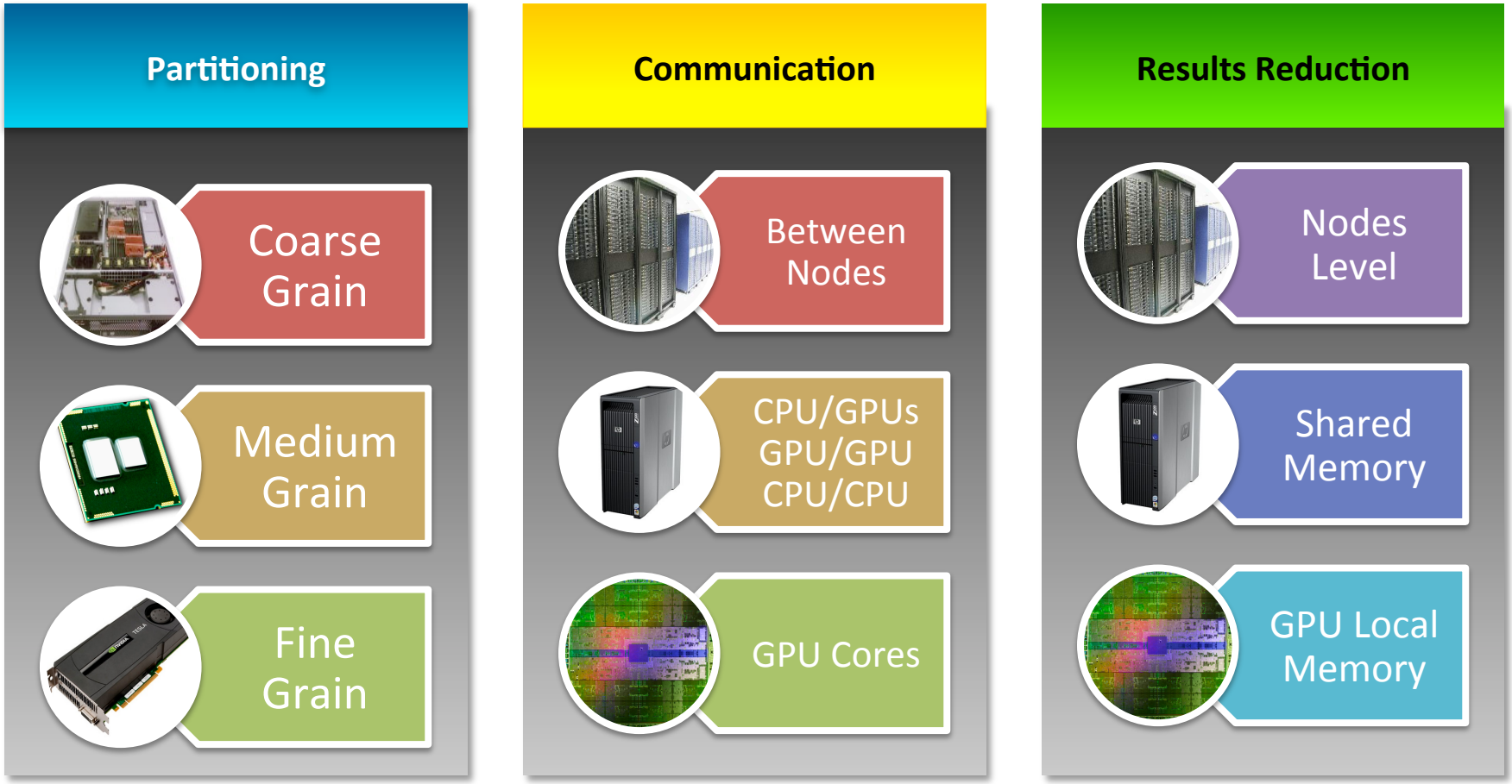
Heterogeneous Architecture

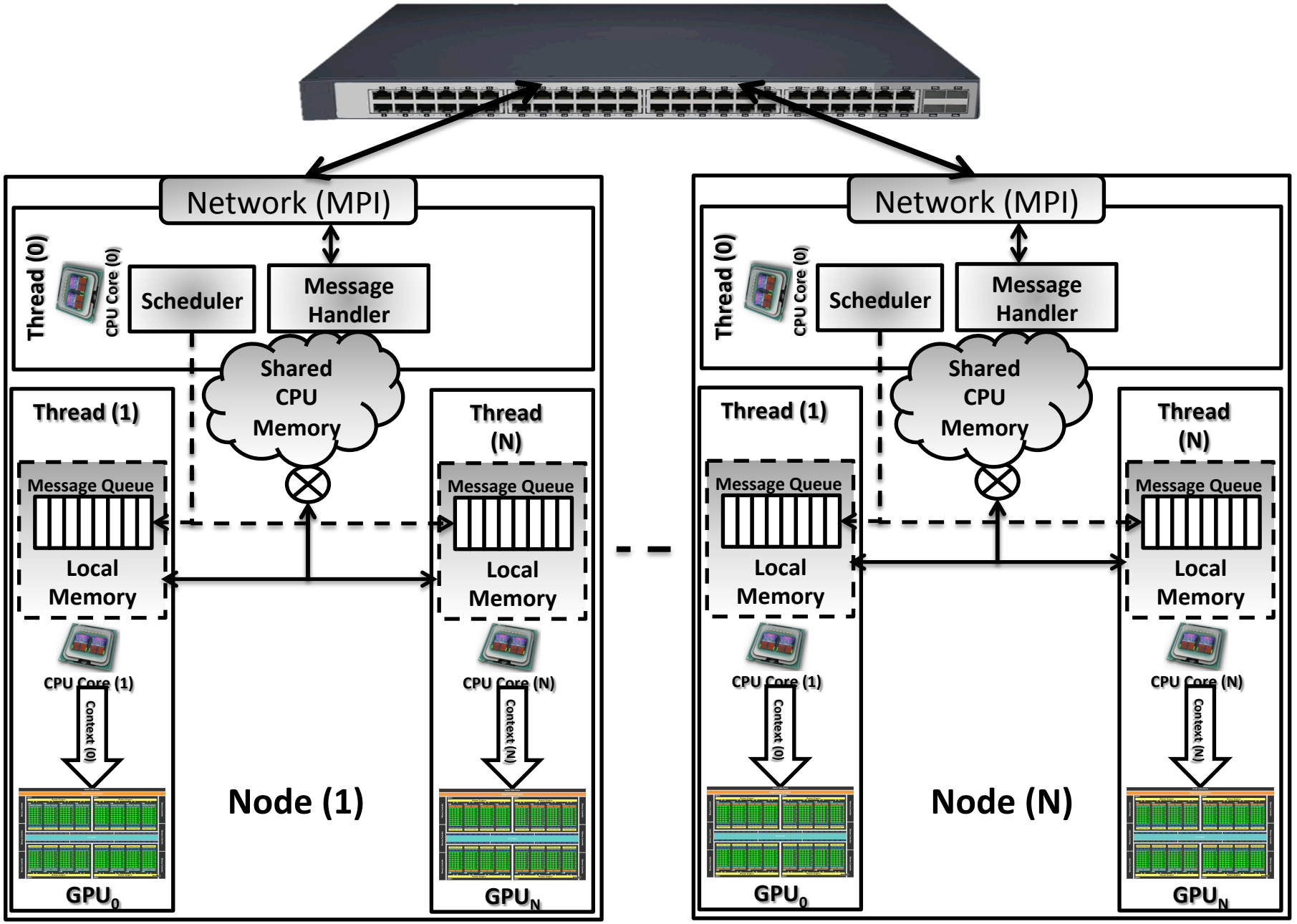
Shared and Distributed Memory – Communication Patterns

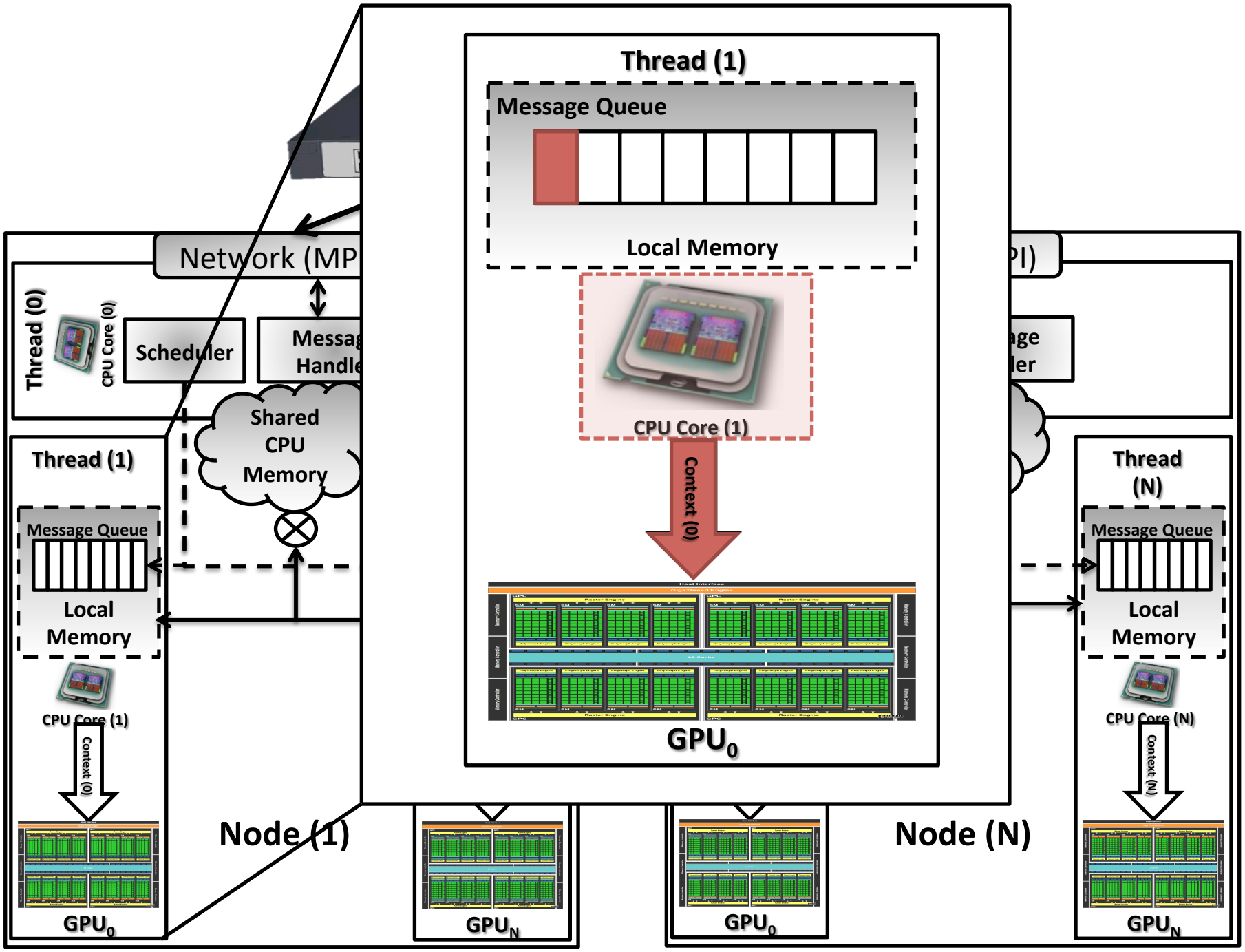


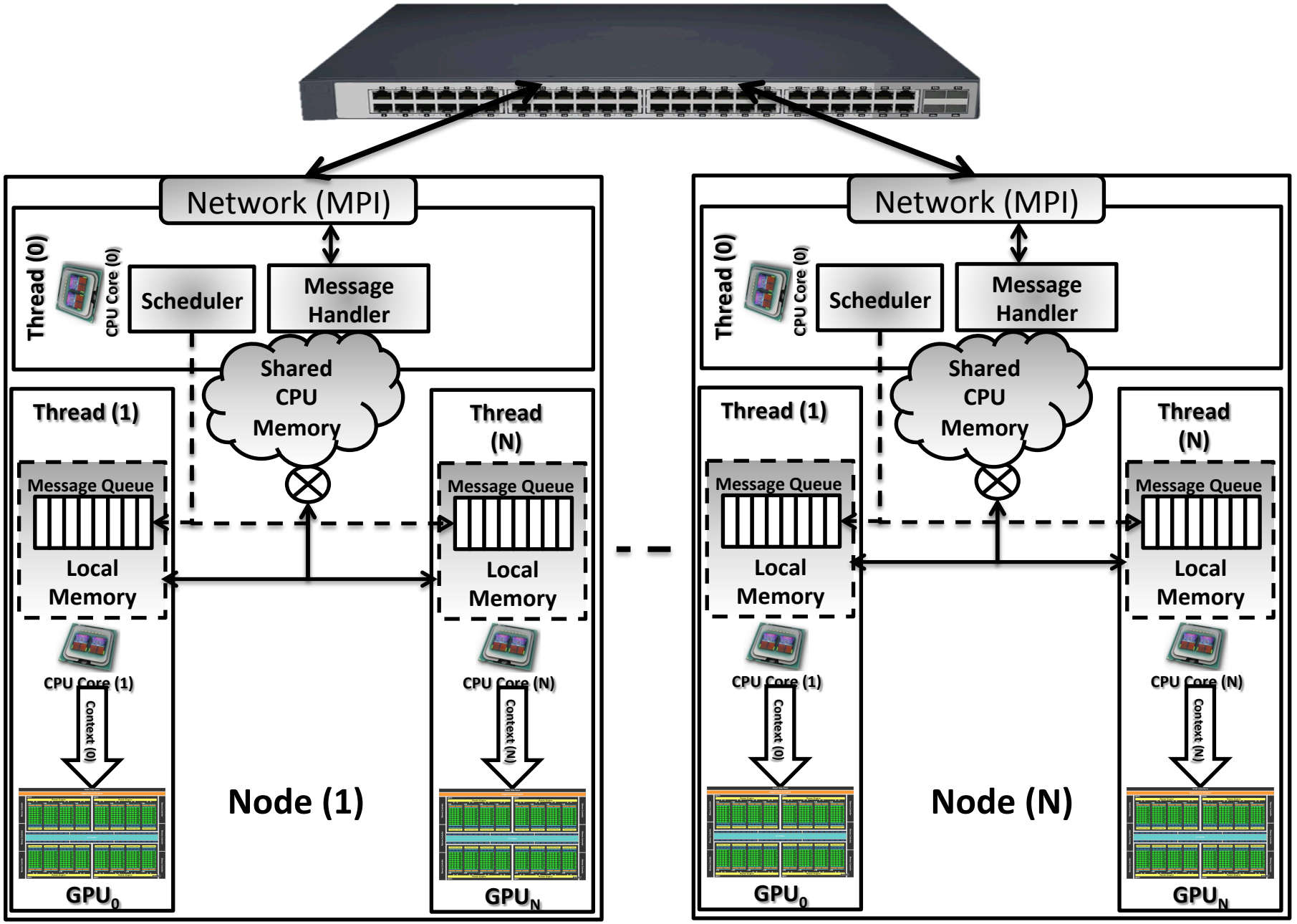
Heterogeneous Architecture

Why it is difficult?




Network (MPI)
Thread (0)
CPU Core (0)
Scheduler
Message Handler
Shared CPU Memory
Thread (1)
Thread (N)
Message Queue
Local Memory
CPU Core (1)
CPU Core (N)
Context (0)
Context (N)
GPU0
GPUN
Node (1)
Node (N)

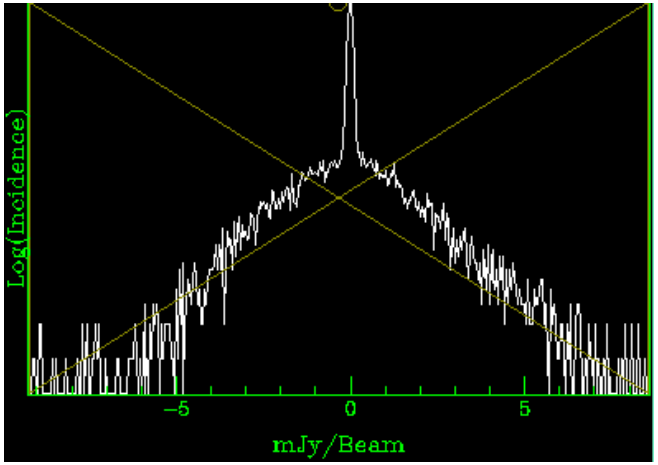

Thread (1)
Message Queue
Local Memory
CPU Core (1)
Context (0)
GPU0
Network (MP
Thread (0)
CPU Core (0)
Scheduler
Shared CPU Memory
Thread (1)
Message Queue
Local Memory
CPU Core (1)
Context (0)
GPU0
Node (1)
GPUN
GPU0
Node (N)
Thread (N)
Message Queue
Local Memory
CPU Core (N)
Context (N)
GPUN

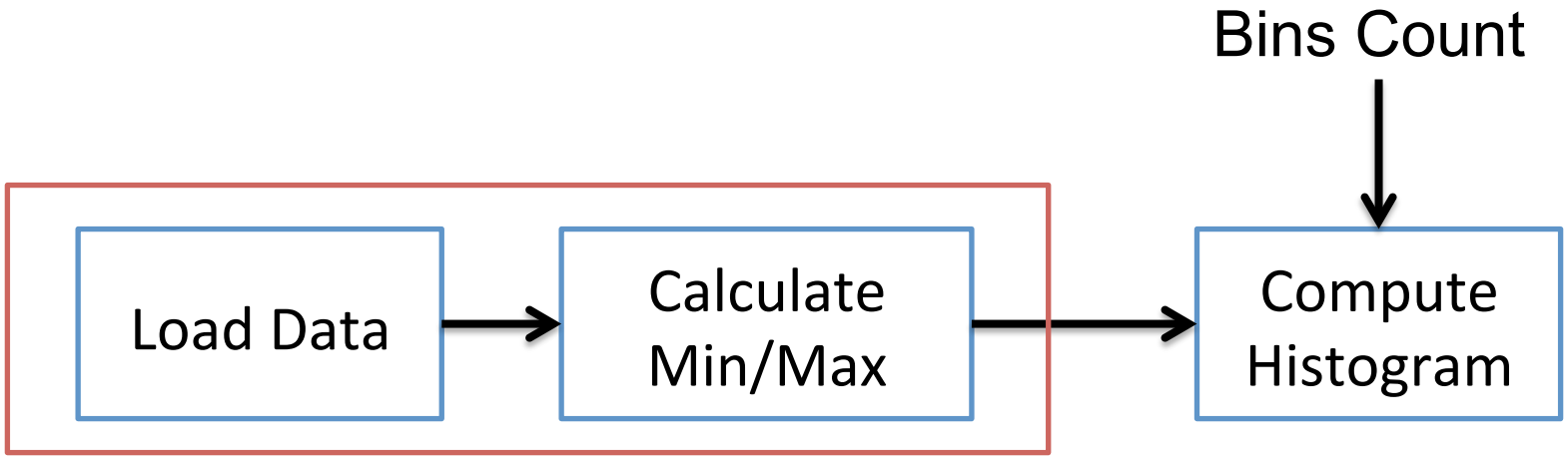

Network (MPI)
Thread (0)
CPU Core (0)
Scheduler
Message Handler
Shared CPU Memory
Thread (1)
Thread (N)
Message Queue
Local Memory
CPU Core (1)
CPU Core (N)
Context (0)
Context (N)
GPU0
GPUN
Node (1)
Node (N)

Example

Computing Histogram

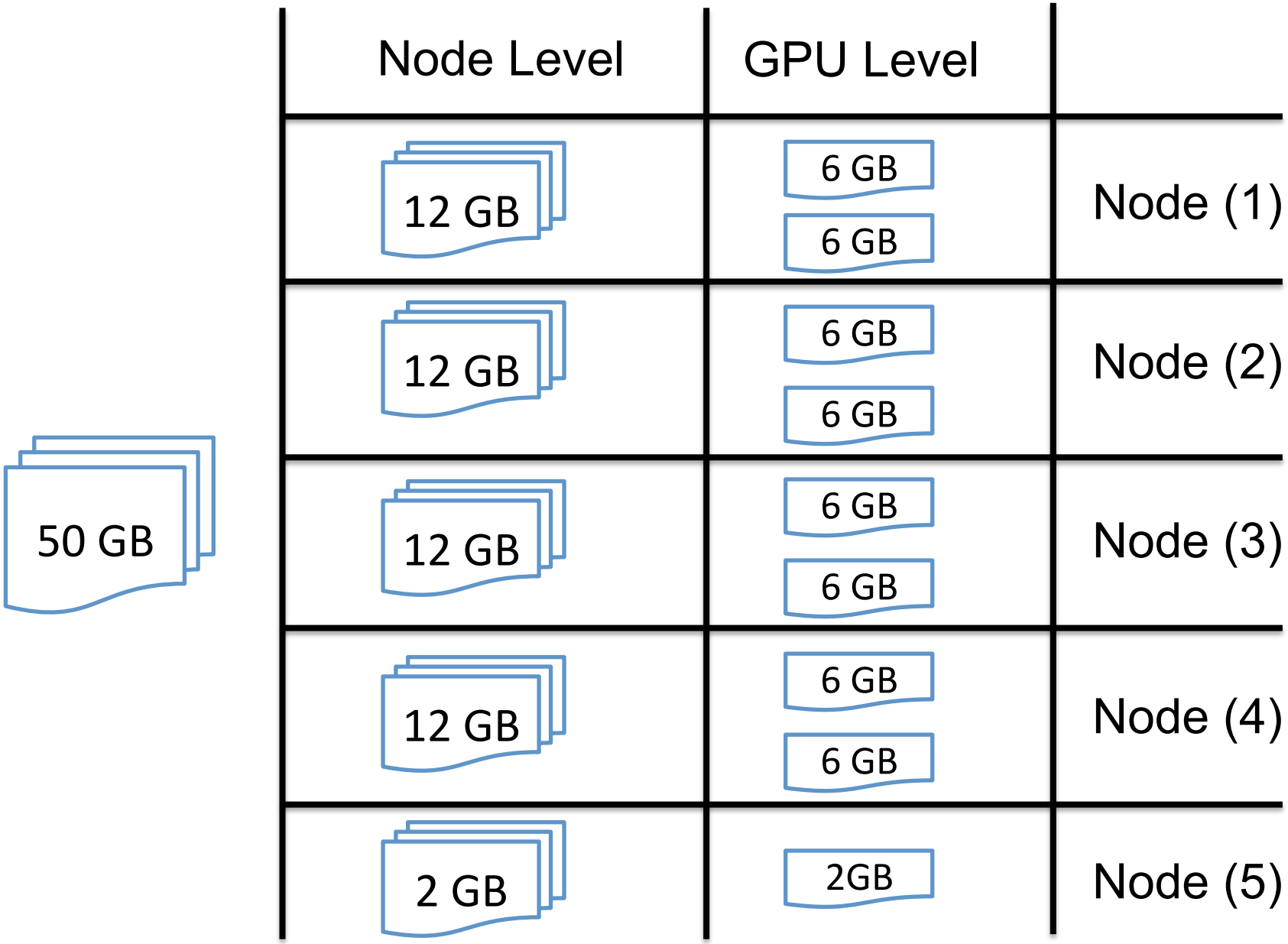
Generate Histogram





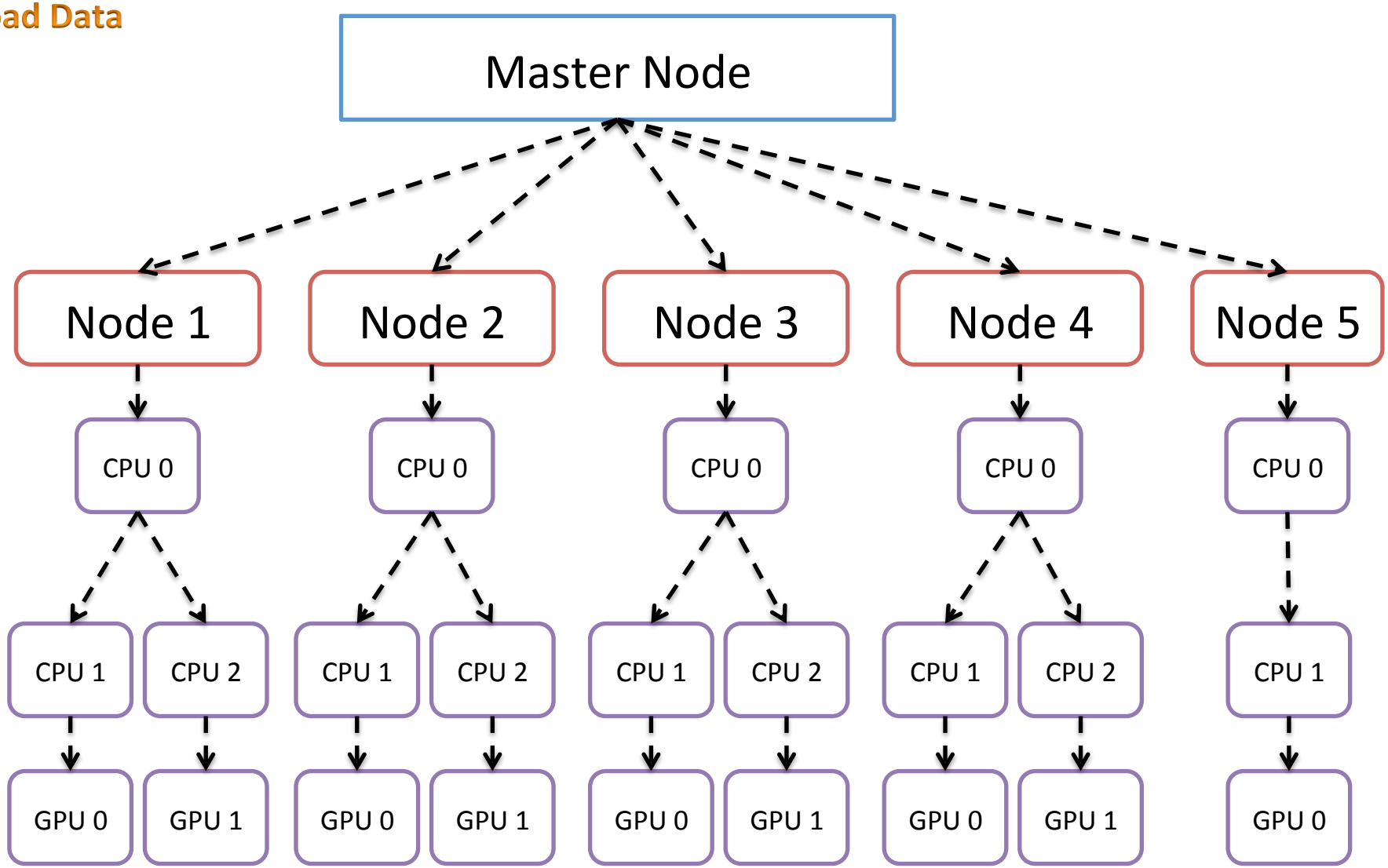
Example

Computing Histogram



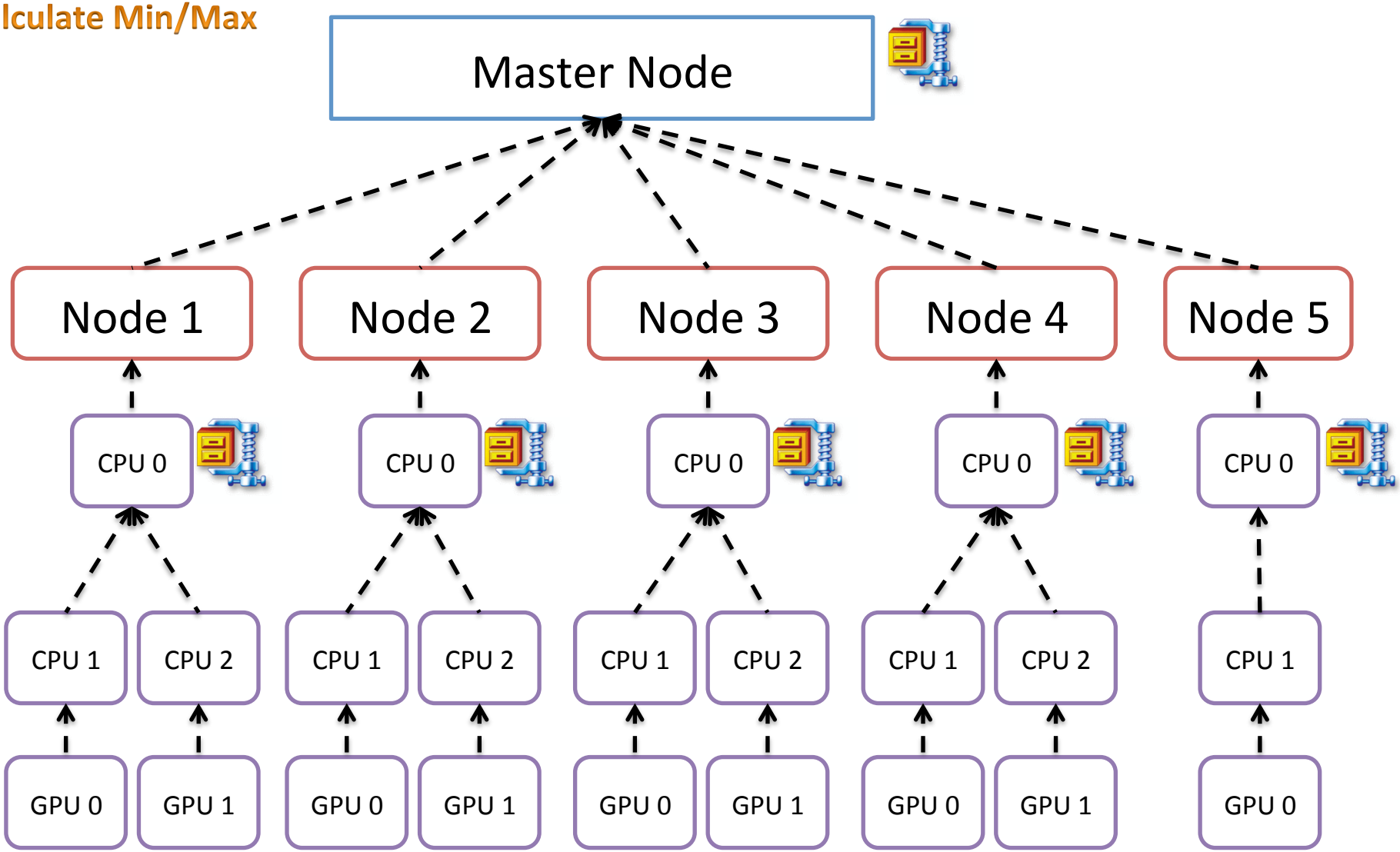
Example

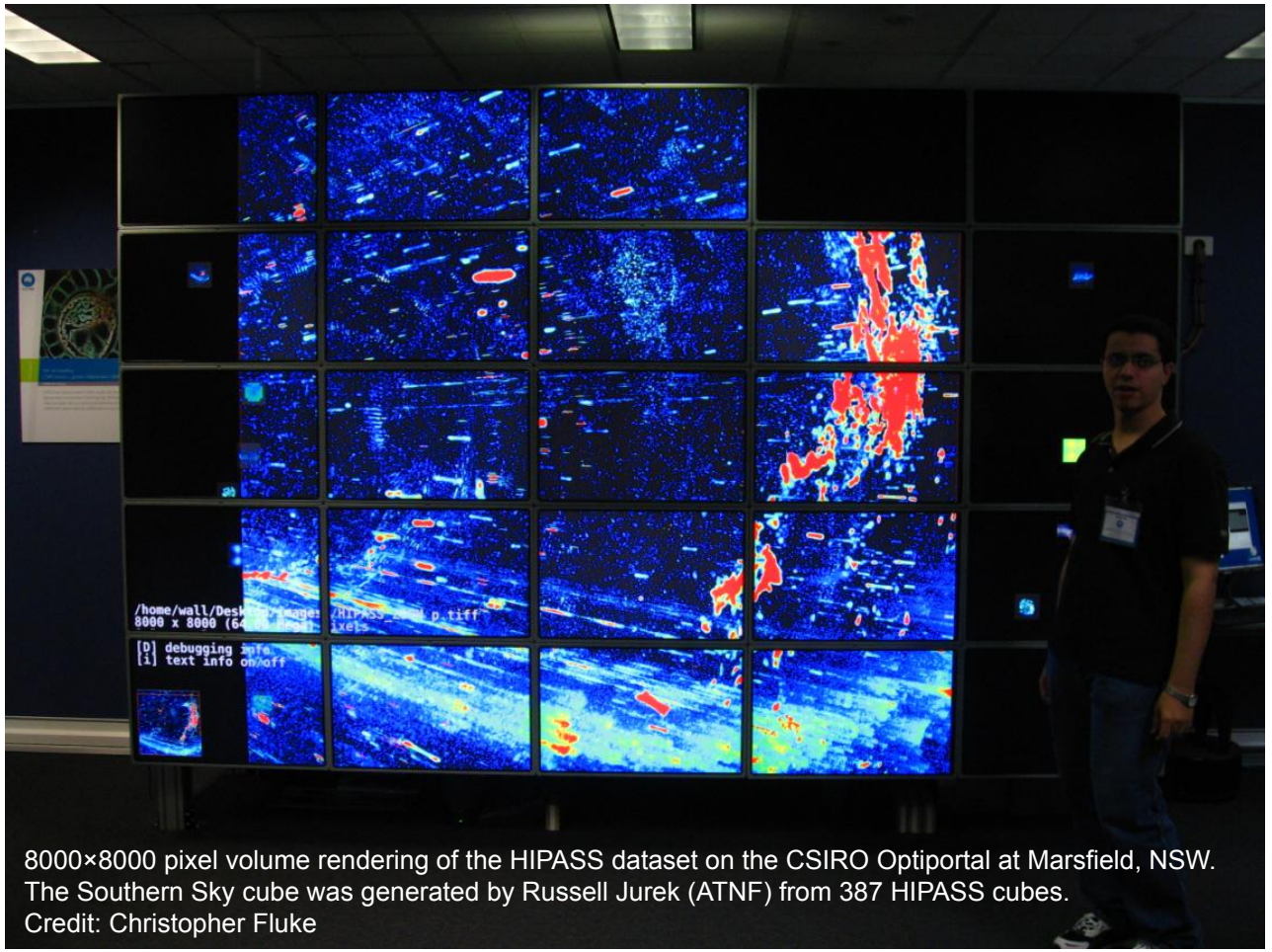
Load Data



Example

Calculate Min/Max





8000×8000 pixel volume rendering of the HIPASS dataset on the CSIRO Optiportal at Marsfield, NSW.
The Southern Sky cube was generated by Russell Jurek (ATNF) from 387 HIPASS cubes.
Credit: Christopher Fluke

Volume Rendering over GPU

MORE COMPLEX APPLICATION

Volume Rendering

Extreme Case!

Communication Overhead

- Large
- Linearly increase with the number of nodes

Computationally Intensive

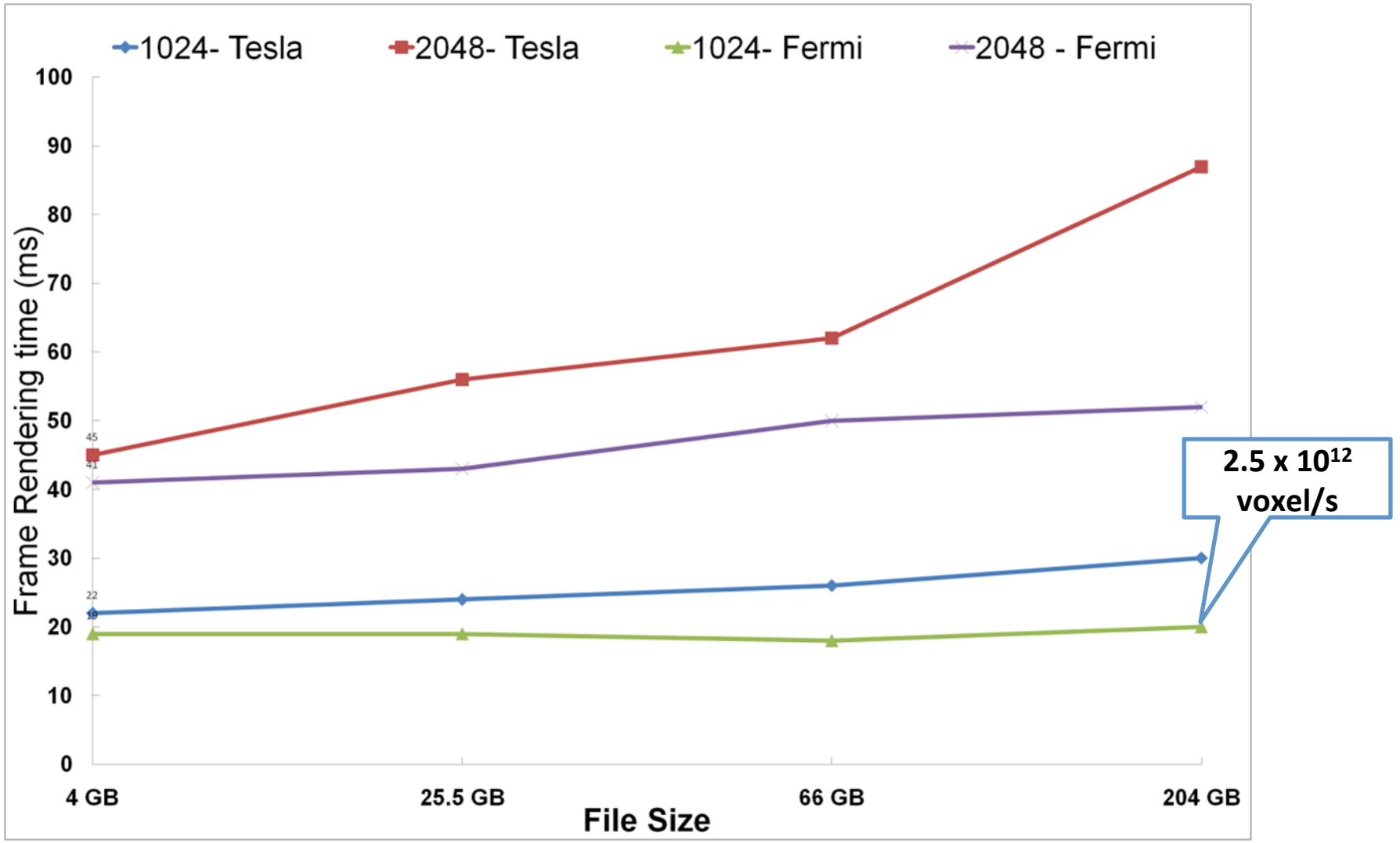
Memory Requirements

Interactivity

- 10 fps required

Distributed GPU

Volume Rendering over GPU- Cluster



See Hassan et. al. New Astronomy,2011

Conclusions

1. GPU is a powerful computational resource but has its limitations
2. Distributed GPUs are harder but worth the effort!
3. Communication bounded rather than processing bounded
4. Overlay computation and communication is important to achieve highest Performance
5. Utilize the shared Memory as long as it is possible to minimize the communication
6. Many-Core Architectures are harder to develop but it might be our wayout …