

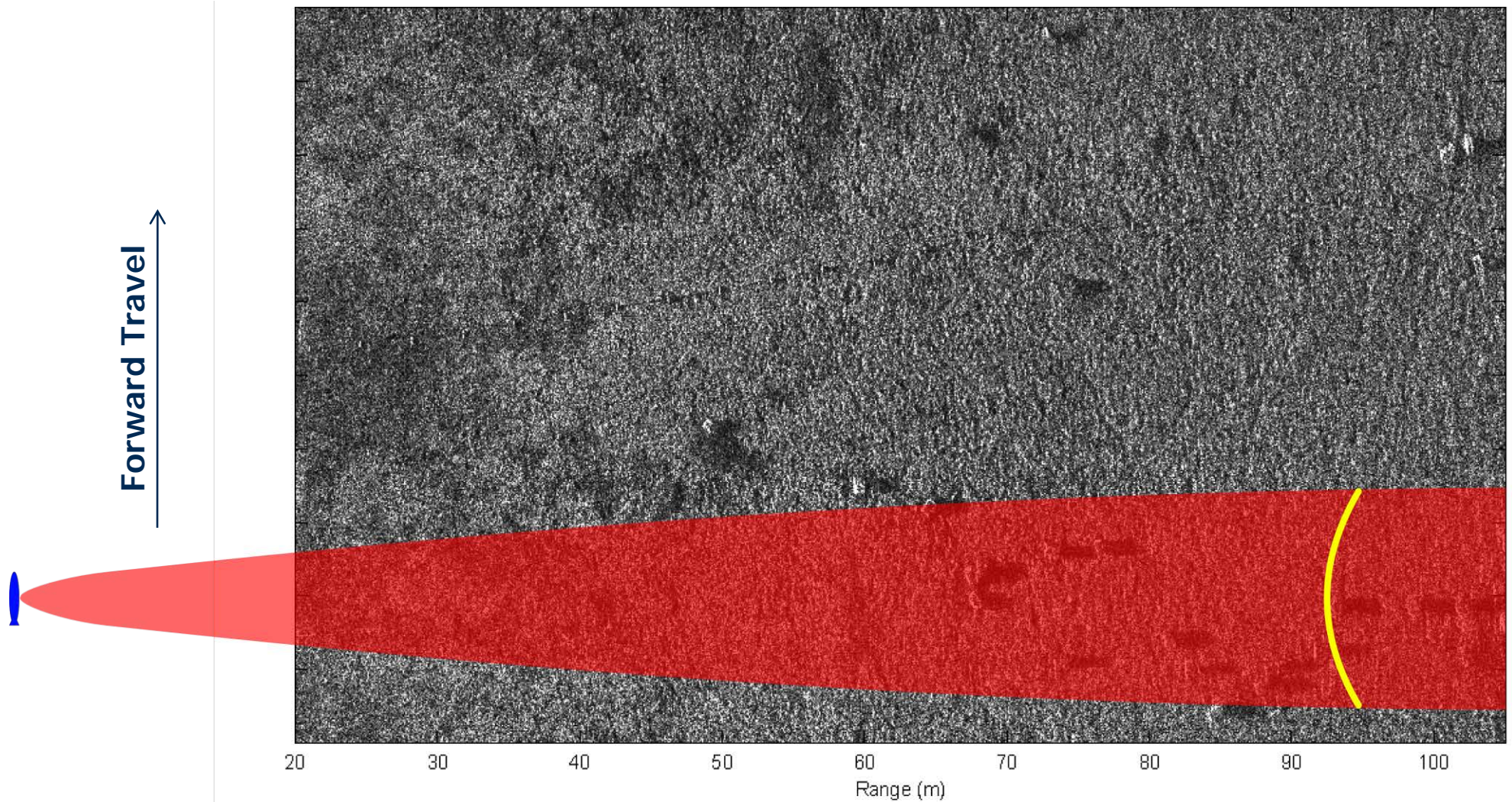
Graphics Processor Clusters for High Speed Backpropagation

2011 High Performance Embedded Computing Workshop
22 September 2011

Daniel P. Campbell, Thomas M. Benson, Daniel A. Cook
dan.campbell@gtri.gatech.edu



Sonar Imaging via Backpropagation



As the sonar passes by the scene, it transmits pulses and records returns. Each output pixel is computed by returns from the pulses for which it is within the beamwidth (shown in red). A typical integration path is shown in yellow.

Backpropagation

•Backpropagation is the simplest synthetic aperture image reconstruction algorithm

$$\begin{aligned} f(\mathbf{x}) &= \int_{\ell} \bar{s}_{\text{ideal}}(\tau_p, t) s(\tau_p, t) d\ell \\ &= \int \delta(\tau_p, t - 2R/c) \bar{s}_{\text{ideal}}(\tau_p, t) s(\tau_p, t) d\tau_p dt \end{aligned}$$

for each output pixel:

Find all pings containing reflections from that location on the ground

Find recorded samples at each round-trip range

Inner product with expected reflection

Sum all of these data points

end

Backpropagation – Practical Advantages

•Procedure

- ~~– Attempt to fly a perfectly straight line~~
- ~~– Compensate for unwanted motion~~
- Form image using Fourier-based method backpropagation
- ~~– Register and interpolate image onto map coordinates~~

•Algorithmic simplicity

- Easier to code and troubleshoot
- Less accumulated numerical error

•Flexibility

- Can image directly onto map coordinates without postprocessing

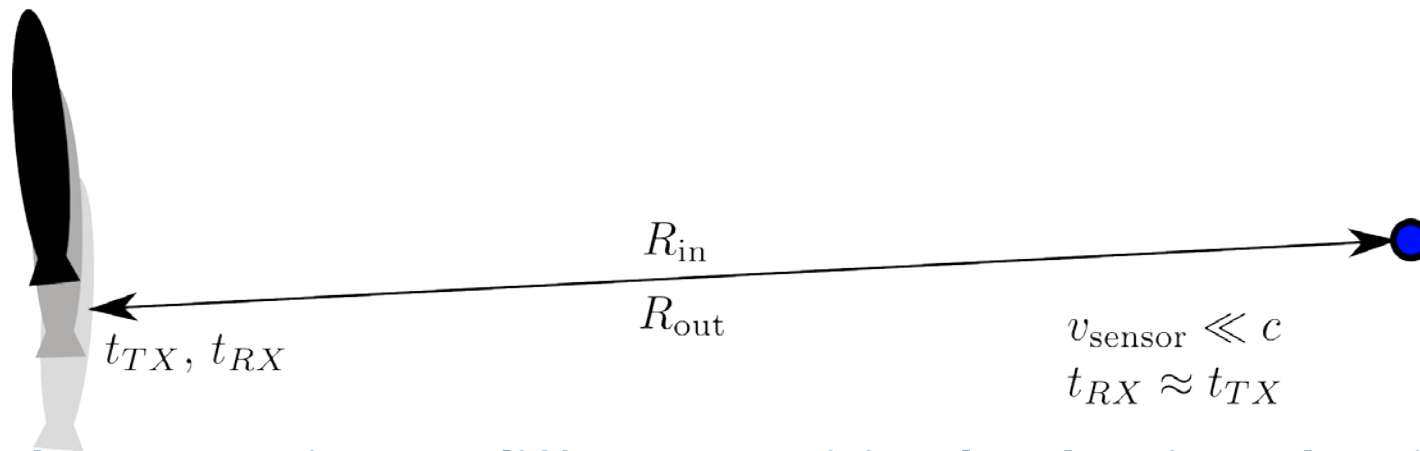
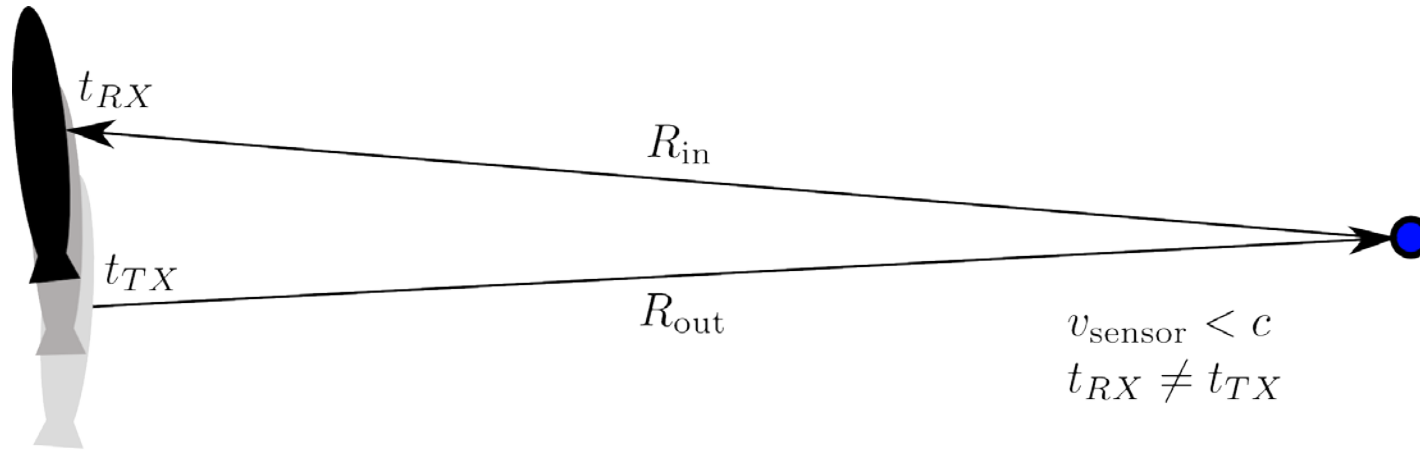
•Expanded operating envelope

- Can image in adverse environmental conditions during maneuvers

Synthetic Aperture Sonar/Radar

- **Typical SAS range/resolution: 100m/3cm**
- **Typical SAR range/resolution: 10km/0.3m**
- **SAS and SAR are mathematically equivalent, allowing (a lot of) the same code to be used for both**
- **The sensor is in continual motion, so it moves while the signal travels to and from the ground**
- **Light travels 200,000 times faster than sound, so SAR processing can be accelerated by assuming the platform is effectively stationary during each pulse.**

Sonar vs. Radar



In general, the sensor is at a different position by the time the signal is received (above). If the propagation is very fast (i.e., speed of light), then the platform can be considered motionless between transmit and receive

Advantages of Backpropagation

- **FFT-based reconstruction techniques exist**
 - Require either linear or circular collections
 - Only modest deviations can be compensated
 - Requires extra steps to get georeferenced imagery
 - Images only onto planar surface
- **Backpropagation is far more expensive, but is the most accurate approach**
 - No constraints on collection path: image during maneuvers
 - Directly image onto any map coordinates desired
 - Image onto height-varying surfaces

Minimum FLOPs per Pixel per Ping

Range out			9
Estimated r/t time			1
Beam Check			5
Final receiver position			65
Final platform orientation	6		
Construct platform final R	35		
Apply R	15		
Add platform motion	9		
Range In			9
Range->Bin			2
Sample & Interpolate			9
Correlate with ideal reflector			9
Accumulate			2
Total			111

Not needed
for Radar

GPU Backpropagation

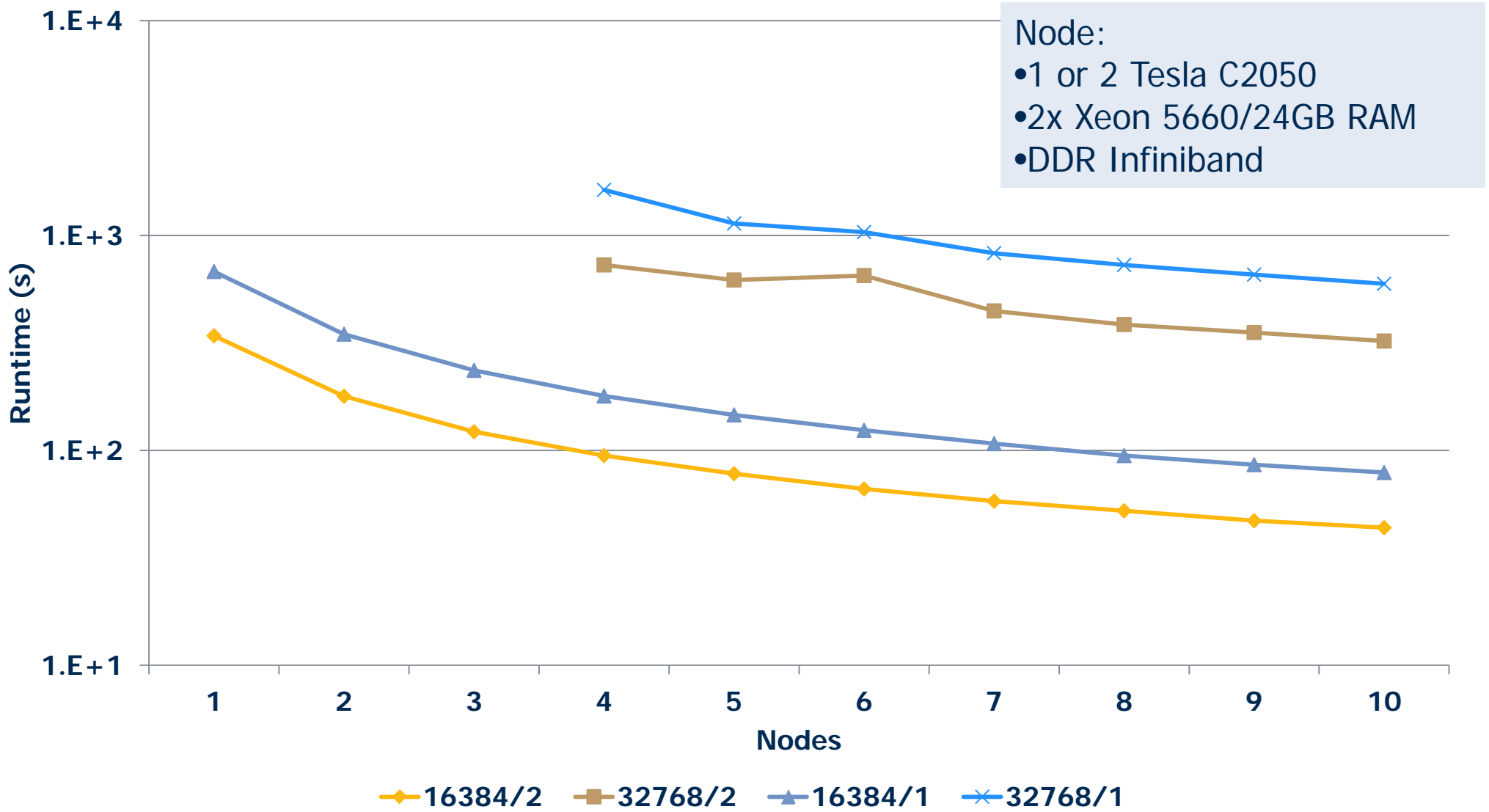
- **GTRI SAR/S Toolbox, MATLAB Based**
 - Multiple image formations
 - Backpropagation too slow
- **GPU Accelerated plug-in to MATLAB toolbox**
- **CUDA/C++**
- **One output pixel per thread**
- **Stream groups of pulses to GPU memory**
- **One kernel invocation per pulse group**

Single-Node Optimization Considerations

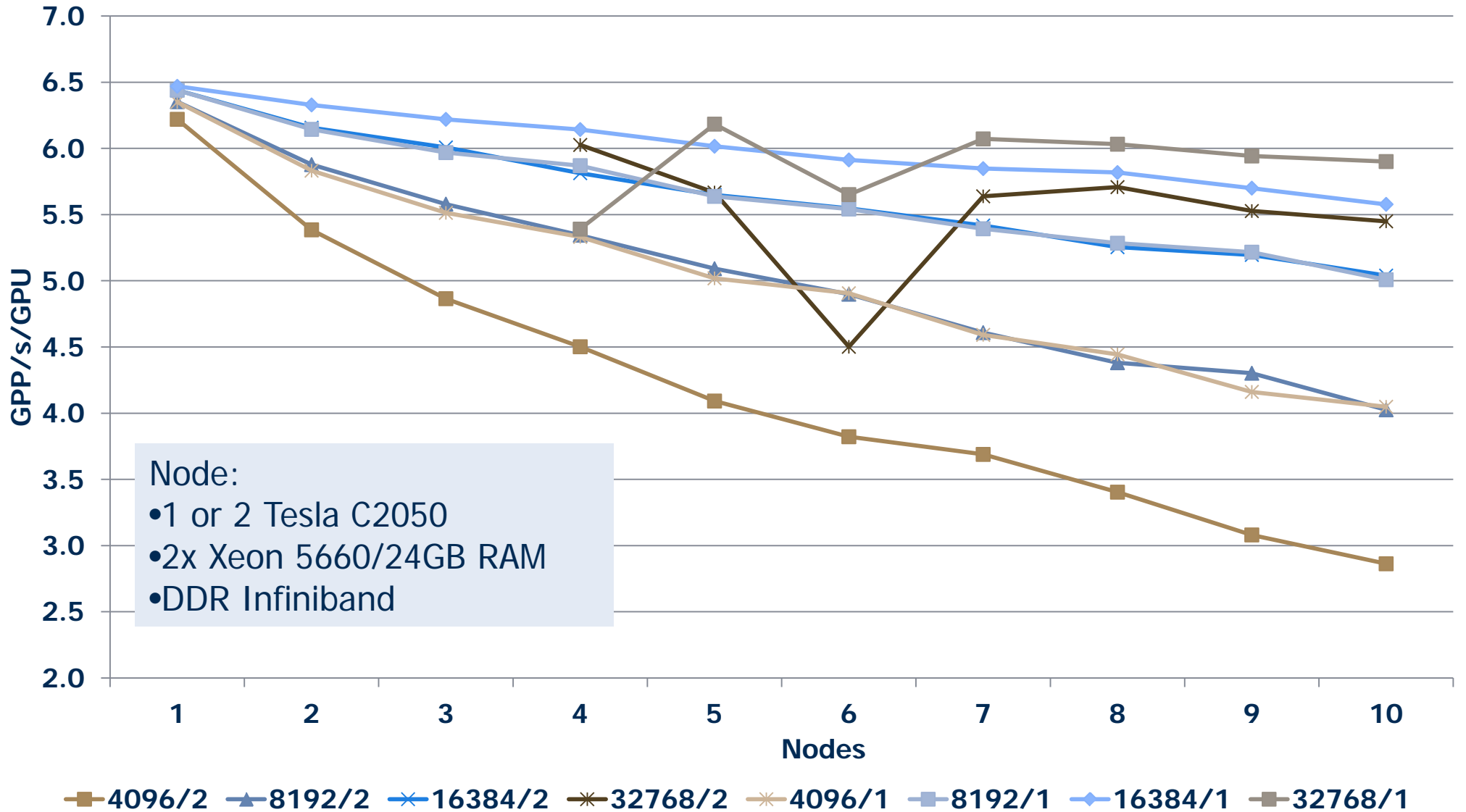
- Textures for clamped, interpolated sampling
- 2-D blocks for range (thus cache) coherency
- Careful control of register spills
 - Shared memory for (some) local variables
 - Reduced precision transcendentals
- Recalculate versus lookup
- Limit index arithmetic
- Careful use of L2 cache versus shared memory
- Eliminate partially dead code

- Peak throughput: 6.47 billion pixel*ping/second

Baseline Scaling



Baseline Scaling



Final Test Platform

- Keeneland Initial Deployment
- 120 Nodes
 - 3 Tesla C2070
 - QDR Infiniband
 - 2 6-core Westmere
- 201 Tflops GNE
- <http://keeneland.gatech.edu/>

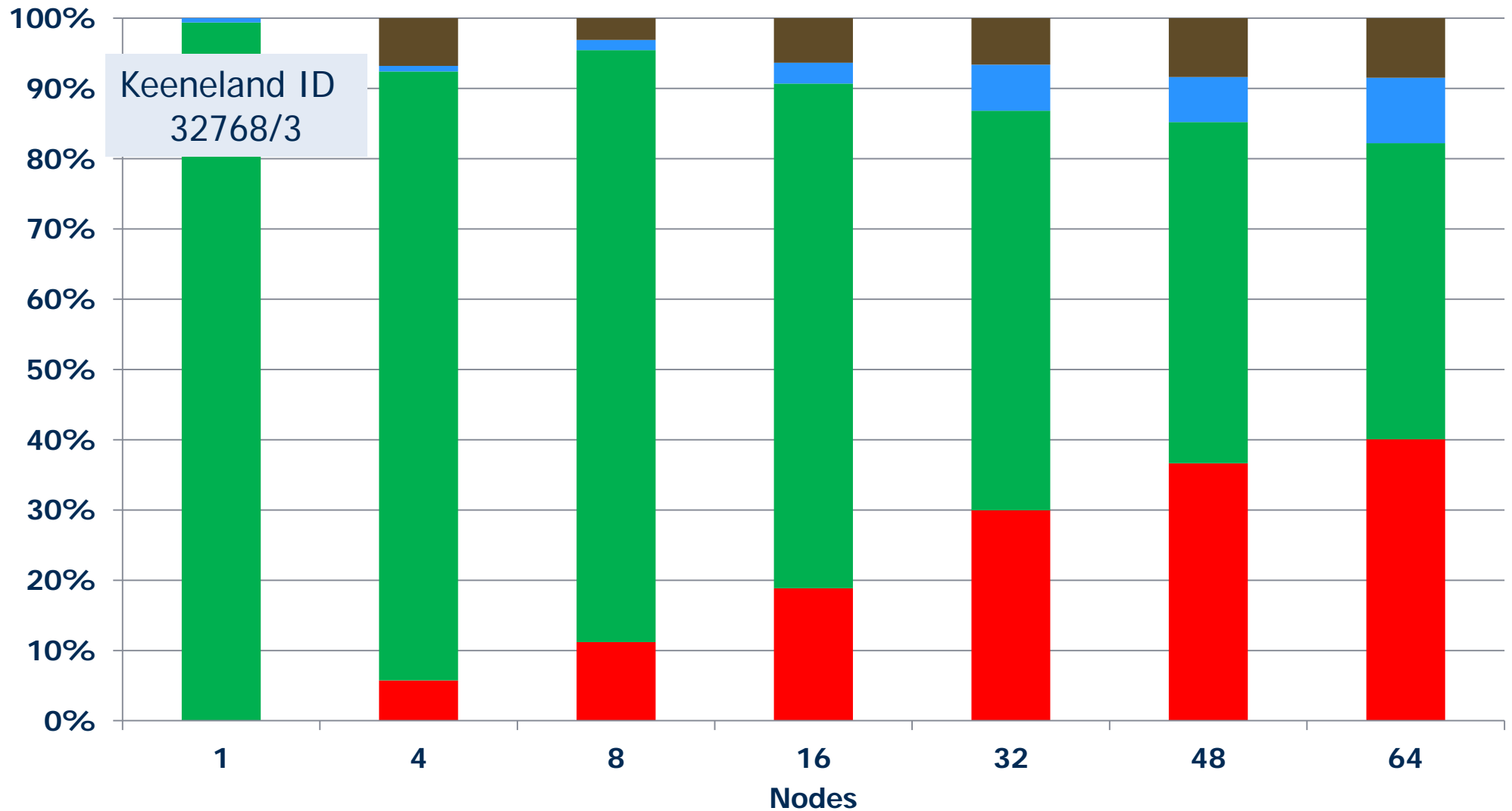
Gigapixel & Large Cluster Considerations

- Full image too big for single GPU memory
- Full input too big for single GPU memory
- Number of pings not used by large groups of pixels quite large
- Load imbalance constraining at high GPU count
- Kernel, PCI-E, and MPI serialized
- Volume of input data not used by a given pixel quite large

Gigapixel & Large Cluster Considerations

- ~~Full image too big for single GPU memory~~
- ~~Full input too big for single GPU memory~~
- Number of pings not used by large groups of pixels quite large
- Load imbalance constraining at high GPU count
- Kernel, PCI-E, and MPI serialized
- Volume of input data not used by a given pixel quite large

Performance Drivers



■ MPI ■ Kernel ■ PCI-e ■ Unbalanced

Gigapixel & Large Cluster Considerations

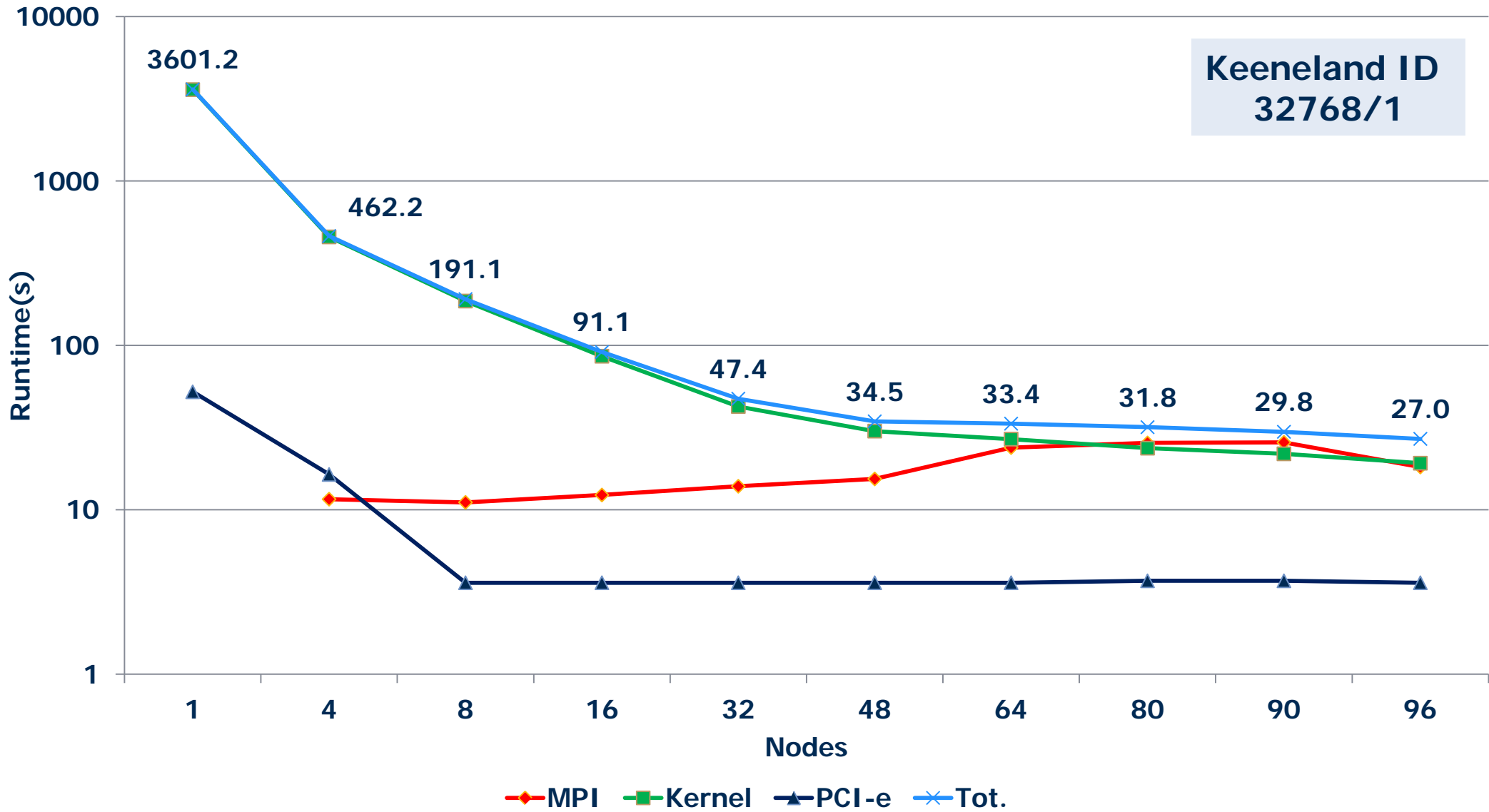
- ~~Full image too big for single GPU memory~~
- ~~Full input too big for single GPU memory~~
- Number of pings not used by large groups of pixels quite large
- Load imbalance constraining at high GPU count
- Kernel, PCI-E, and MPI serialized
- Volume of input data not used by a given pixel quite large

Gigapixel & Large Cluster Considerations

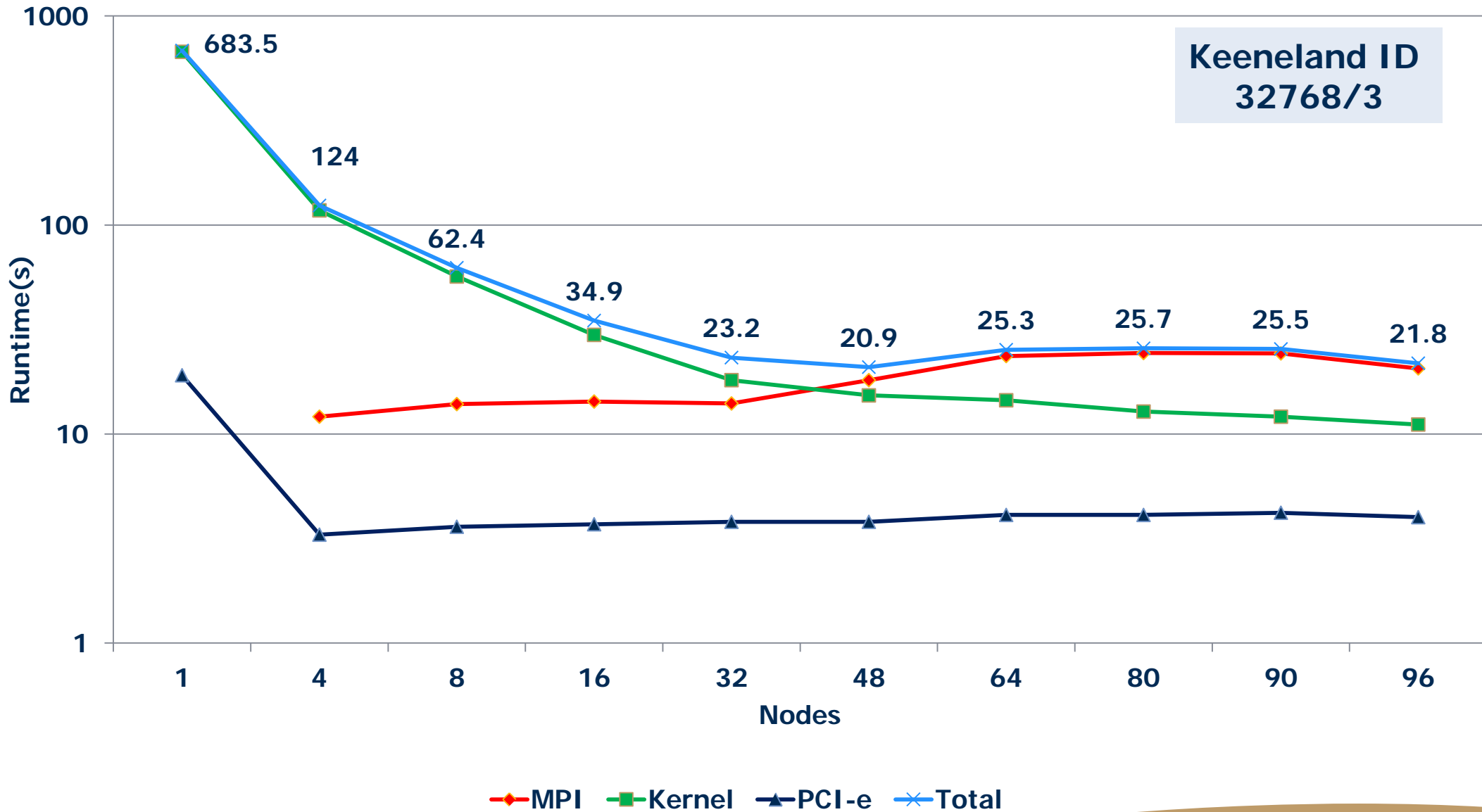
- ~~Full image too big for single GPU memory~~
- ~~Full input too big for single GPU memory~~
- ~~Number of pings not used by large groups of pixels quite large~~
- ~~Load imbalance constraining at high GPU count~~
- ~~Kernel, PCI-E, and MPI serialized~~
- Volume of input data not used by a given pixel quite large

Achieved Performance

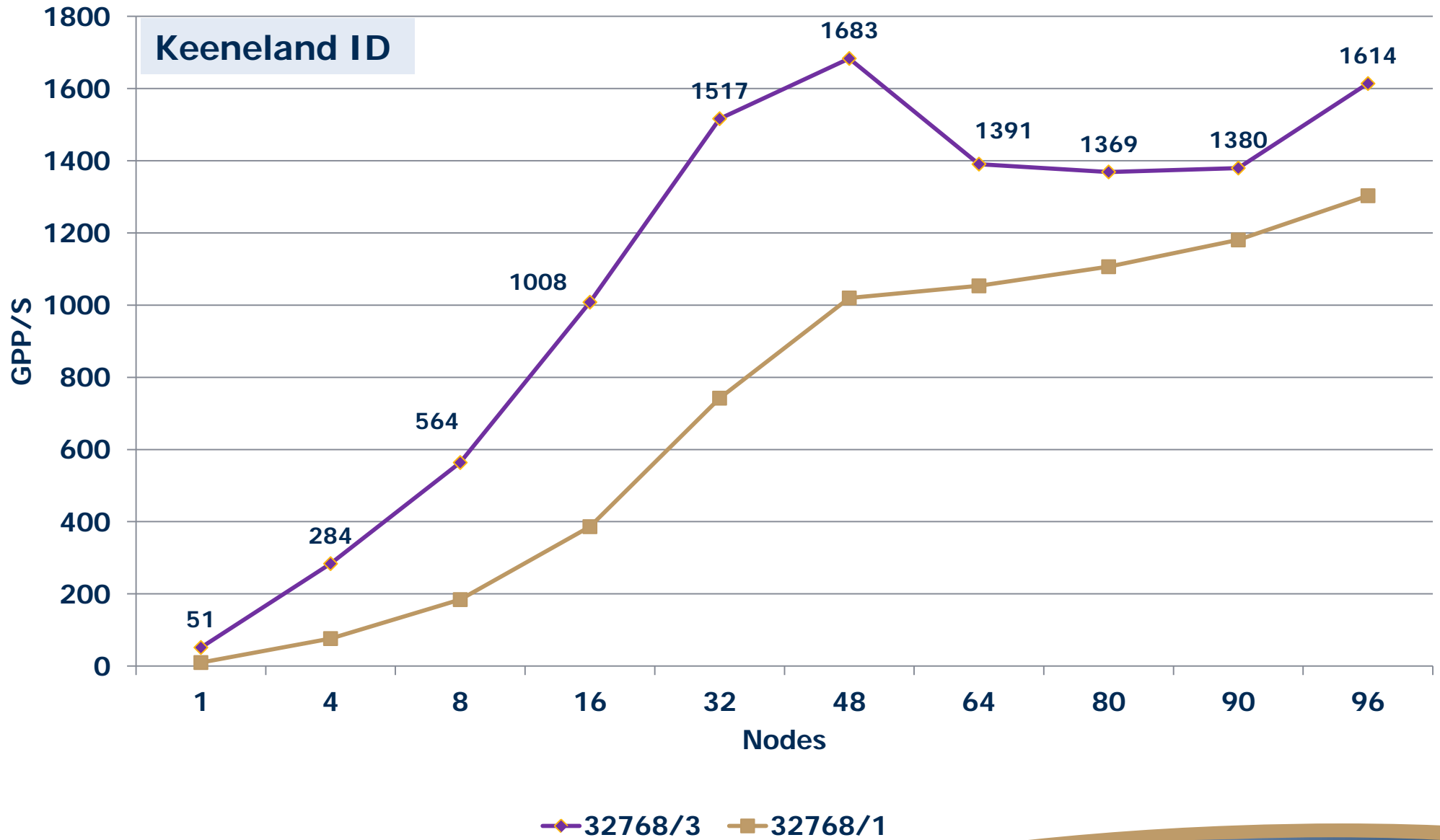
Keeneland ID
32768/1



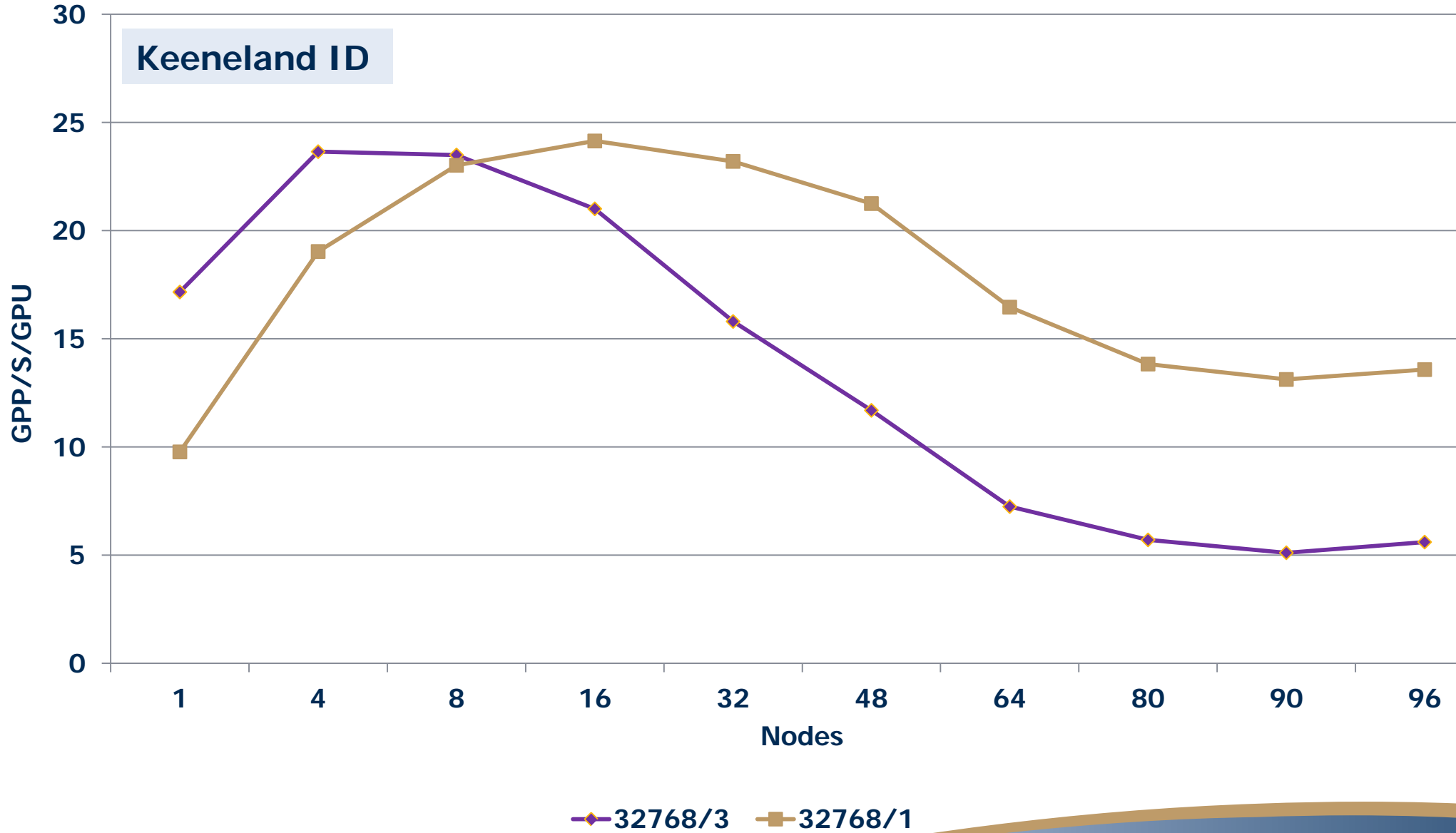
Achieved Performance



Achieved Performance



Achieved Performance



Future Work

- **MPI Culling**
- **Improve tuning for SAR**
- **Preconditioning data reduction**
- **Improve error handling, edge cases, etc.**
- **Cloud service**