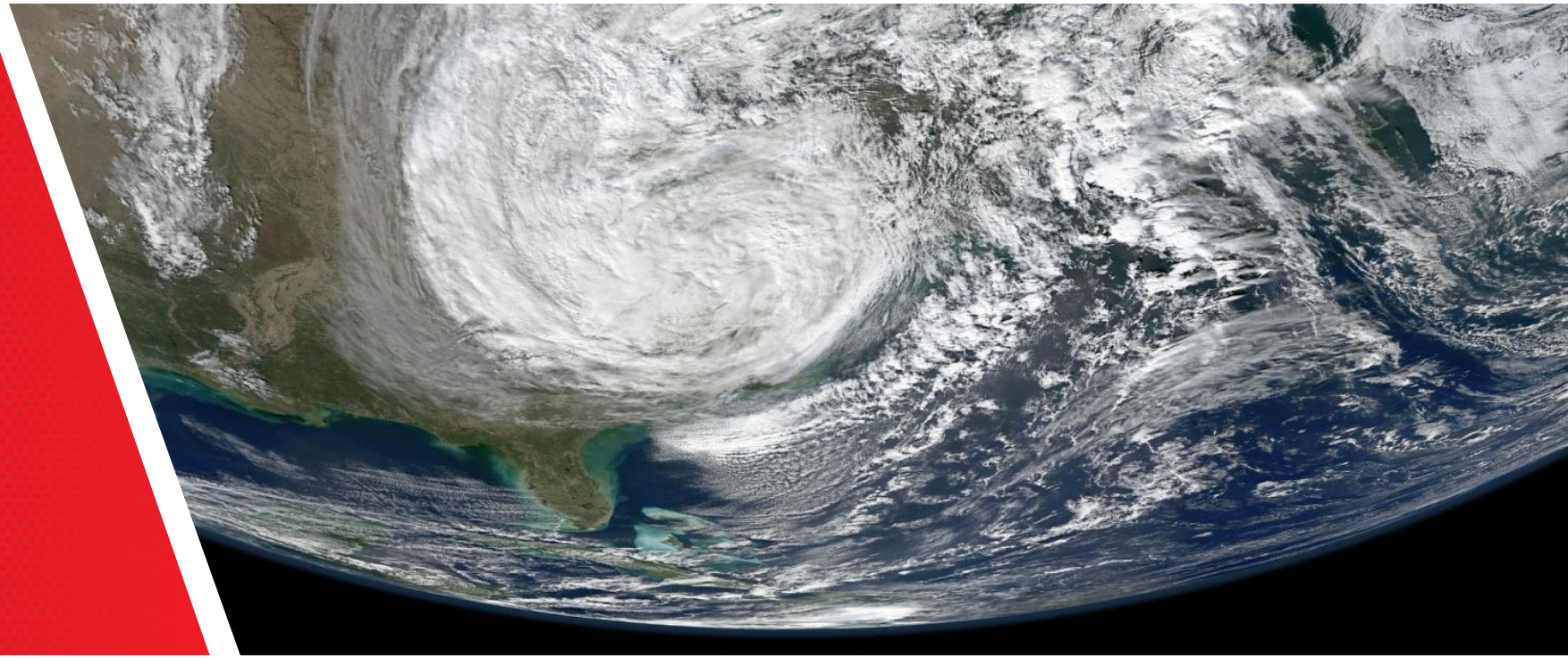
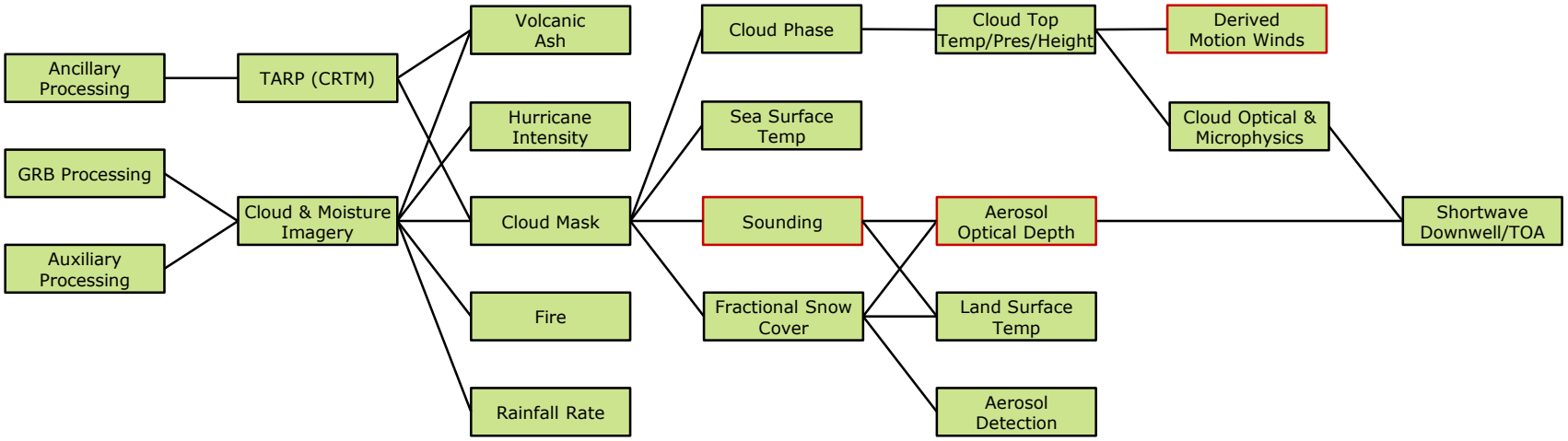


Leveraging HPC Technologies to Accelerate GOES-R Product Processing and Distribution

01/11/2016

Significant Advancements Require HPC Technologies

- The GOES-R mission represents significant advancements in spatial, spectral and temporal resolutions combined with extensive L2+ processing capabilities
  - Raw data processed at a rate of 100Mbps and Level 0/1/2+ data processed at a rate of 16.1TB per day
  - 20 algorithms that produce 35 aerosol, cloud, land, ocean, and space weather environmental products distributed across hundreds of servers
  - Product latencies as low as 1.8 seconds for solar products and 23 seconds for imagery products
  - 10x more data, 6x more often than the current GOES system, providing users 60x more data than before
- To accommodate these advancements GOES-R product processing and dissemination utilizes multiple HPC technologies



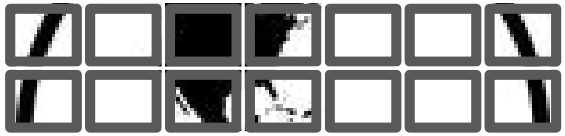
GOES-R Mission Presents Multiple Challenges

Compute Intensive	• Serial execution of algorithms cannot satisfy tight product latencies
High Throughput	• 697,168 product files per day • 16.1TB product data per day
High Reliability	• System availability 99.99% • Product availability of 99.9%
Adaptability	• Complex product dependency model • Add/update algorithms at run-time
Scalability	• Scale 300% without redesign
Security	• FISMA High

Why HPC Technologies Are a Good Fit

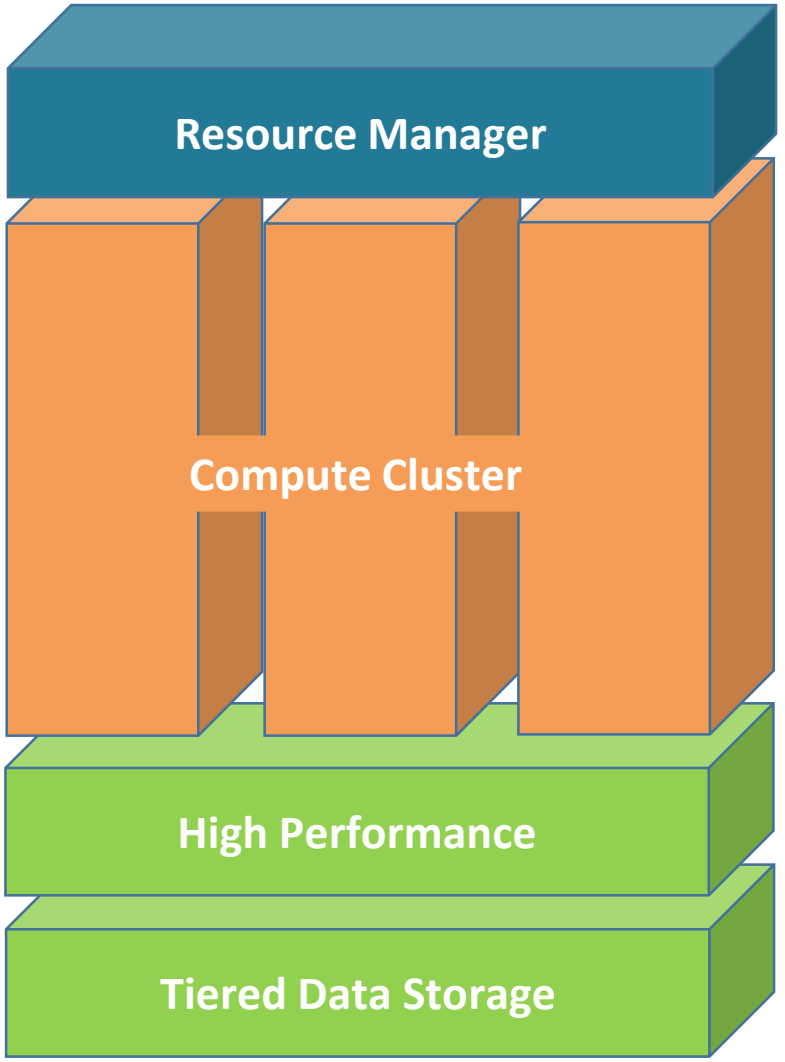
HARRIS

Algorithm Characteristics	*Design Approach*
GOES-R L2 Algorithms are Embarrassingly Parallel...	**Parallelized Block Processing**
❖ Algorithms run independent from each other (do not communicate at runtime with ordering based on product precedence)	An instance of each algorithm is a unique service that can be run on any node within the cluster
❖ Each Algorithm is agnostic to geographic location and dimension	Multiple instances of an algorithm run in parallel working on different geographic regions (blocks)
...With some exceptions	**Distributed Shared Memory**
❖ Some algorithms need larger input area than they output (Neighborhood Window)	Services can request any area they need to process, regardless of the block size produced
❖ Some algorithms need data from previous refresh	Services can request older data sets cached in memory

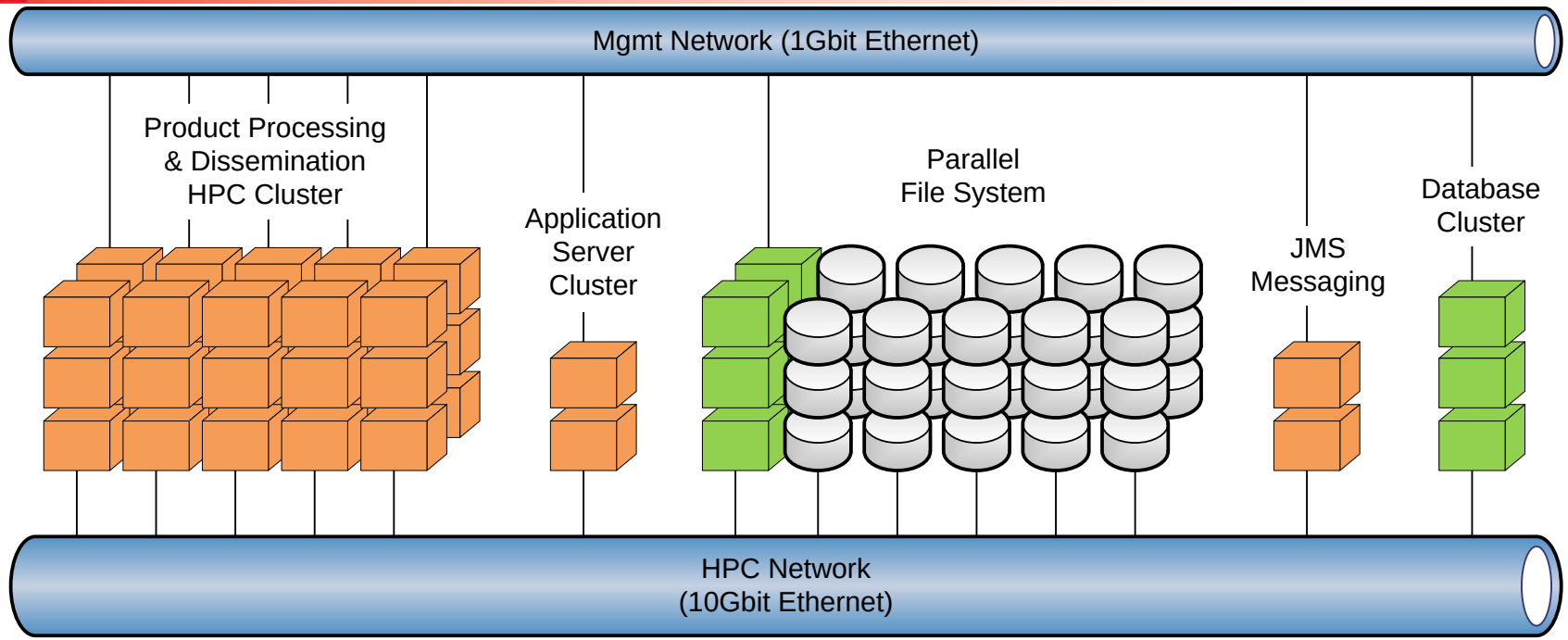


HPC Technologies for Reliable, High Performance Processing & Distribution

HPC Technologies	Benefits
Robust Parallel Processing Framework	Flexibility to configure workloads based on data partitioning Standards-based inter-process messaging Data replication and service resiliency Highly scalable management and monitoring tools
Commodity HPC Cluster	Vendor neutral OTS components grounded in open, standards-based technologies Extensible and scalable accommodating software evolution and growth without redesign
High Performance Tiered Data Storage	Optimized for high throughput, low latency, highly reliable transactions Utilizes a mix of in-memory data operations and a high performance parallel file system
Comprehensive Multi-Level Security	Compliant with latest standards (FISMA, NIST, FIPS, DOC and NOAA) Reduces security vulnerabilities and maintains operational integrity and reliability



GOES-R Product Processing & Distribution Architecture



Compute
HPC cluster of commodity 2P x86 servers running Linux

Supports heterogeneous environment of processors and hardware vendors
---
x86 processing architecture provides the most extensive application and tool support

Network
High throughput, low latency 10 gigabit Ethernet interconnect
---
Converged compute and storage networks reduces complexity

Robust, highly scalable messaging appliances provide greater performance than software-based brokers

Tiered Storage
I/O performance is addressed using a mix of in-memory data operations (Data Fabric) and a high performance parallel file system
---
Storage virtualization combined with a high performance parallel file system provides a unified enterprise storage solution for both file and block data

Built using commercially available technologies

Inter-process Data Messaging

- Original middleware was built on Parallel Virtual Machine (PVM)
  - Support was waning
  - Not reliable enough
- Considered MPI as a replacement
  - Not reliable enough
  - Could not support easily add new algorithms while actively processing data
- Developed a unique, OTS based middleware solution – Data Fabric
  - Distributed shared memory for data persistence
    - High performance in-memory data cache
  - JMS for event notification/messaging
    - High volume messaging

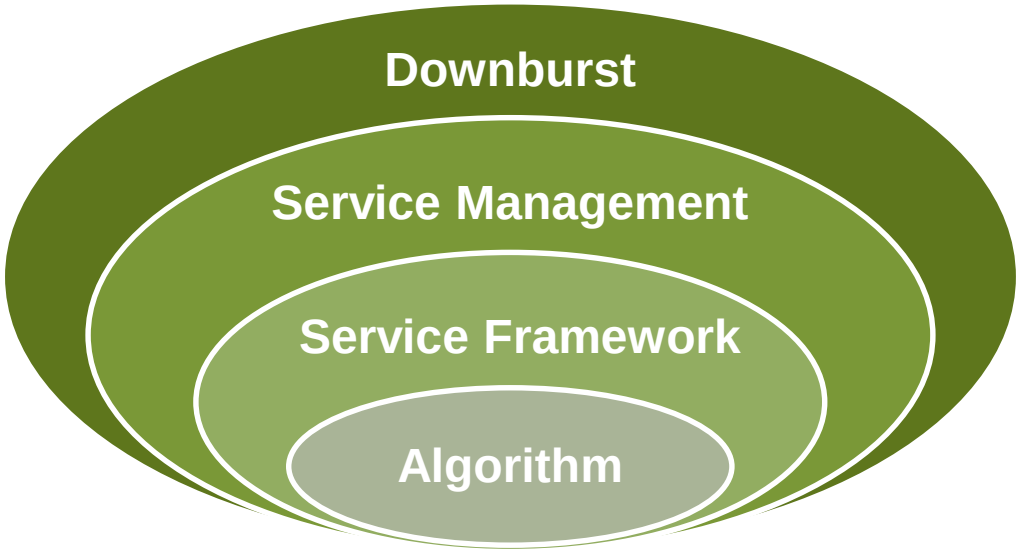
Data Fabric Advantages

High Performance	• Product data resides in RAM for faster throughput • Only data needed for processing individual blocks is sent to service (not entire product)
Data Driven	• Services are notified when data is available through data events over JMS Topics • Processing starts once all data is available
High Reliability	• Data is mirrored in the Data Fabric, so a single Data Fabric server failure does not cause data loss
Flexibility	• Services can connect to the Data Fabric at any time without affecting other services, including upstream services • Data can be retrieved at any block size independent of how it was written to Data Fabric
Scalability	• Data Fabric can scale linearly by adding additional servers to Data Fabric • Hardware resources can be added or removed during operations without affecting performance

DownBurst™ Provides Infrastructure for Parallelism

Robust Parallel Processing Framework – DownBurst™

- Service Framework encapsulates science algorithms as a service
  - Enables multiple instances of an algorithm to run independently in parallel
  - Provides algorithms with an interface to the Data Fabric
- Service Management orchestrates services across the compute cluster
  - Provides overall resource management
  - Controls the number of instances of an algorithm
  - Manages geographic regions an instance processes



Resource Management

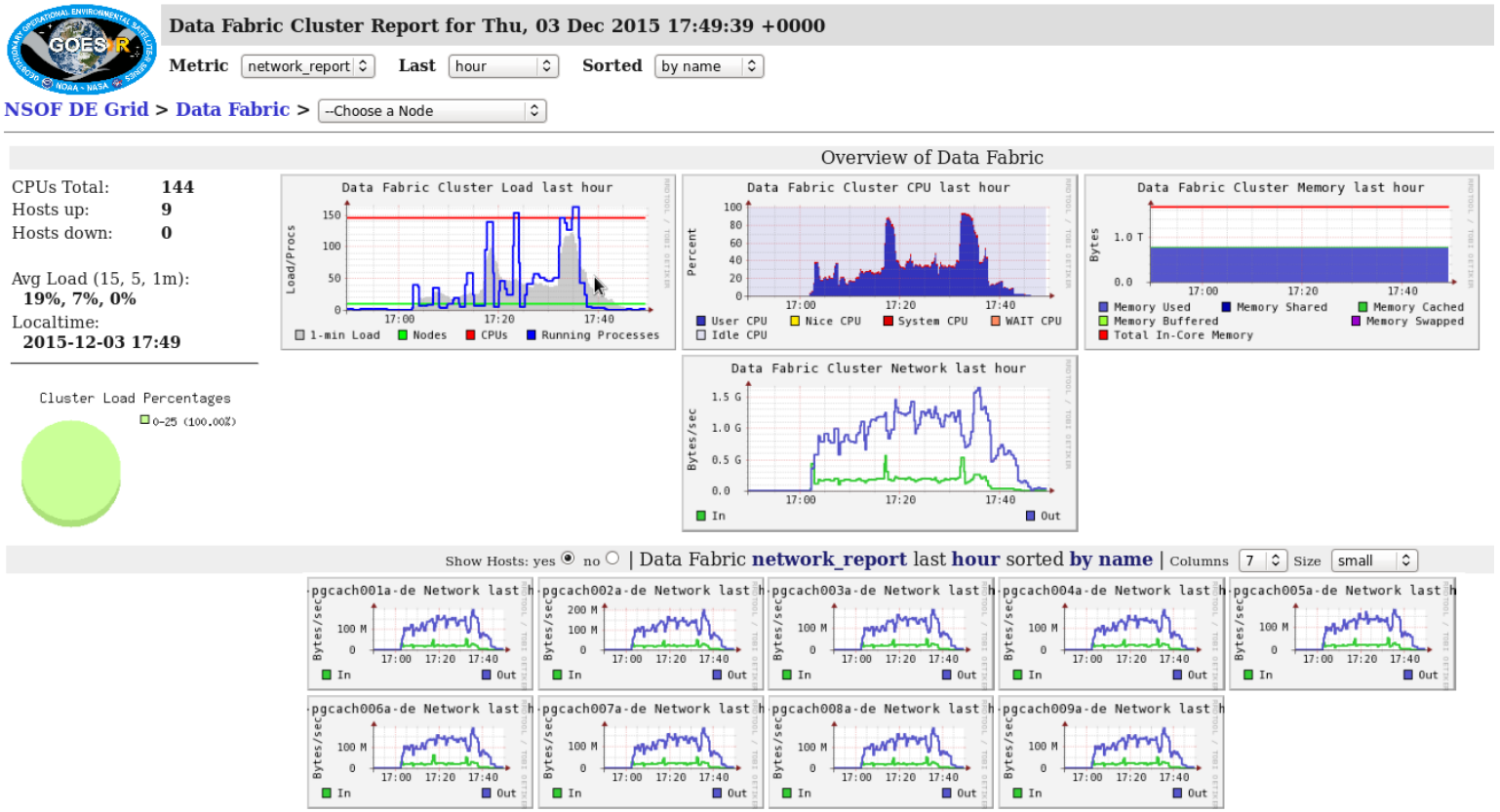
- Need tools to manage and monitor services and the cluster
  - Deploy algorithms/services across the compute cluster based on workload
  - Monitor algorithms/services for:
    - Software failures – algorithms/services crash
    - Hardware failures – Node goes down or loses network connection
    - System performance – CPU, memory, network, etc.
  - Recover failed algorithms/services to ensure application and hardware failures do not result in the loss of significant amounts of product data or system time

- Explored various job scheduler tool suites
  - Included both open source and proprietary solutions
    - Grid Engine, MOAB/TORQUE, PBS and LSF

- Selected Open Grid Engine
  - Low cost
  - Simple management
  - In-house experience
  - Supports redundant management servers
  - Highly scalable
  - DRMAA compliant

Performance Monitoring - Ganglia

- Aggregates performance statistics across distributed systems
  - CPU, memory, network, load, etc.
- Commonly used in HPC Community
- Used for monitoring GOES-R product processing and dissemination to quickly assess potential performance issues



Conclusion

- HPC Technologies are well suited for application to GOES-R Product Processing and Distribution
  - Commodity HPC clusters combined with DownBurst™ provided a cost effective solution for hosting parallel execution of GOES-R algorithms
  - System is easily scaled to accommodate evolution and growth without redesign
  - GOES-R L2+ processing is a good fit for parallel processing, enabling the system to meet rigorous product timelines
  - Inherent redundancy of HPC clusters simplified achieving reliability requirements

This approach is applicable to other remote sensing ground processing systems

For more information on this topic at AMS, visit:

- Poster #373 at GOES-R/JPSS Poster Session 1 (Today)
- Poster #764 at GOES-R/JPSS Poster Session 2 (Wednesday)

Visit Harris Corporation Booth #501