# CHOPS Design Review
# Consolidated High-throughput Operational Products System (CHOPS)

Lee Marks, NESDIS/OSPO
Joseph Mani, NESDIS/CIO

March, 2015

# CHOPS Overview

## CHOPS Overview:

- CHOPS stands for "**C**onsolidated **H**igh-throughput **O**perational **P**roducts **S**ystem" CHOPS is a platform in ESPC based on an OpenSource project that can harness large collections of distributively owned computing resources
- CHOPS uses a shared file system that is a unified, multi-protocol, scale-out architecture that has the capability of serving many petabytes of data
- CHOPS is built on low-cost white-box commodity X86 server hardware for the compute and storage nodes in the system. With the capability of leveraging Enterprise grade HDDs all the way to consumer HDDs
- CHOPS is designed to provide high-throughput computing for satellite product generation within ESPC. It can produce products from multiple (GOES, POES, LEO) data sets simultaneously

# CHOPS Capabilities (1 of 2)

- CHOPS provides an architecture for compute intensive applications, and can easily extended without redesign to support applications that require large amounts of RAM or very high file IO
- This is all done with physical hosts, but can be just as easily supported in a virtual environment
- Virtualization is used to augment the system. All of the monitoring and management roles within the system are delivered as virtual machines across a cluster of virtualization hosts
- All of the compute and storage nodes exist as physical servers in a clustered configuration
- Both architectures will be used in tandem to provide resources to produce satellite products
- CHOPS allows high utilization of  lower cost white-box commodity hardware
- Reduced the need for fiber optic network connectivity by leveraging 10GBASE-T copper to all of the compute and storage nodes

# CHOPS Capabilities (2 of 2)

## Enterprise Workload Scheduler

- Enterprise Workload Scheduler provides Monitor/Control functions for CHOPS through a GUI interface
- We use an existing Enterprise Workload Scheduler that has been in operational in ESPC for 2.5 years
- The existing Enterprise Workload Scheduler monitors/controls all of our existing Polar PG systems
- The Enterprise Workload Scheduler is able to submit, execute, monitor, control, and report status of CHOPS jobs and DAGs (Directed Acyclic Graph)

# CHOPS – Adding New Applications

- CHOPS is configured with the Application binaries, scripts and config files stored on the Shared File System
- The System Libraries are installed on each compute node
- When a new App is developed on CHOPS,  it is compiled and tested on a VM and verified
- Once the new App development tasks are complete, the binaries and required files are moved to the Shared File System. Normal CM migration is performed
- Scheduler scripts and jobs are developed to run the algorithms within the framework.
- At execution time the Scheduler system automatically queues, monitors, and executes the Applications
- There is no need to stop the existing System workloads to add or remove new Applications. Product generation is not interrupted

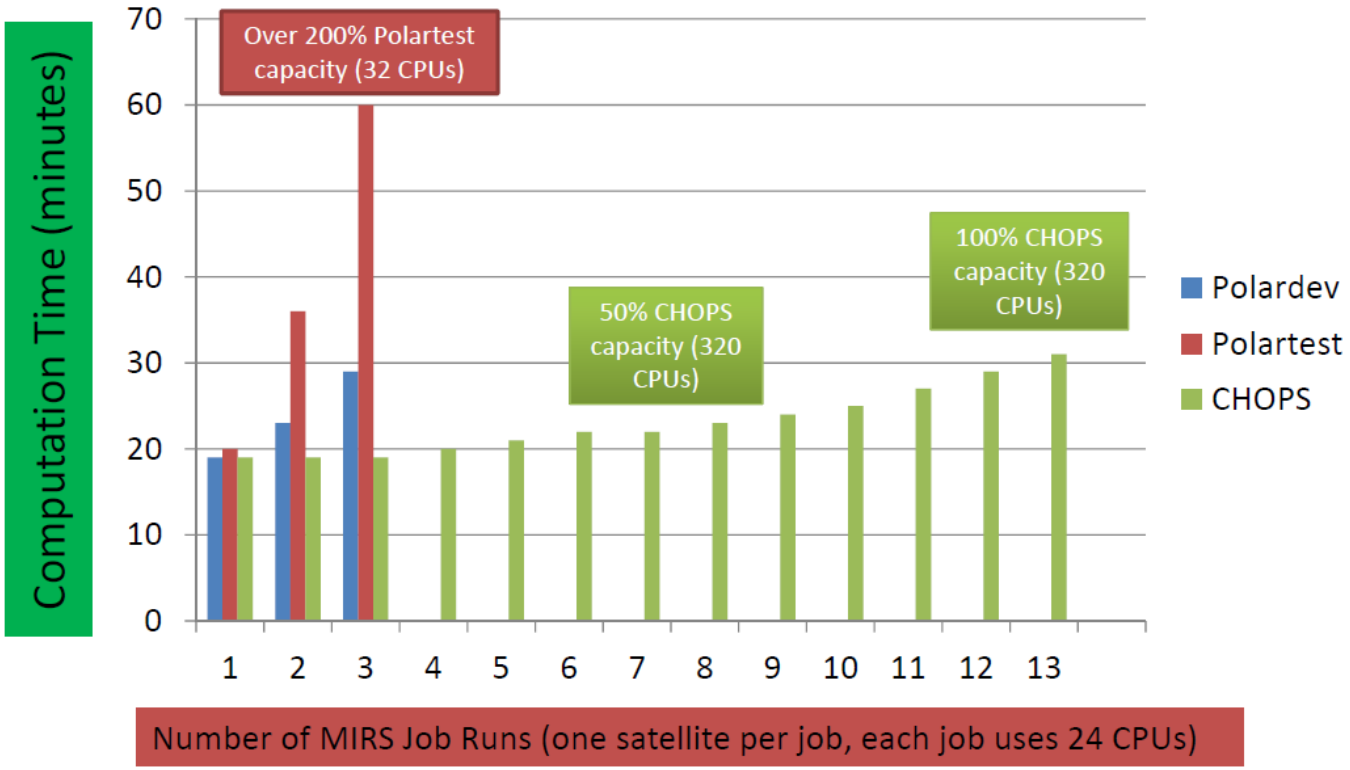
# CHOPS Pros

## CHOPS Pros

- Any X86 server than can run Linux and is connected to the network is a potential resource for the pool
- Any server in the pool is a potential resource for all applications
- High Availability exists by design, losing one or many nodes will not halt operations.
- Resource Contention is resolved dynamically by design
- Designed to be dynamically scaled-out easily, and does not require reconfiguration of the system or software
- Granular assignment of pool resources, we are able to deliver CPU resources at the thread level
- Due to the use of commodity systems, upgrading RAM or storage is relatively inexpensive
- The network infrastructure is a scale out, leaf and spine design

# CHOPS Pros

## MIRS Job Performance on Different Platforms (Tested with different CPU loads)



# Scheduler Framework Capabilities

## Supported Computational Environments

- Standard - Allows for job check-pointing and Remote System Calls
- Vanilla - Does not provide job check-pointing nor Remote System Calls
- Grid - Connects to Cloud Resources
- Java - Runs Java jobs on any operating system running a JVM
- Scheduler - Runs jobs that will not be preempted if the owner reclaims the compute node
- Local - Runs jobs directly on submission computer
- Parallel - Allows for Message Processing Interface (MPI) on multiple compute nodes
- VM - Creates Virtual Machines as needed for jobs