SCAMPI

Scalable Copy Accelerated by MPI

Scalable Copy Accelerated by MPI (SCAMPI) is a command line utility to accelerate data transfer, and is currently installed on most HPCMP HPC systems. SCAMPI accelerates data movement through the use of multiple streams over multiple sockets of multiple IPs. In other words, SCAMPI uses multiple threads across multiple edge nodes to perform its transfer.

Due to overhead and set up requirements, SCAMPI should only be used when needing to copy more than 1GB among the HPCMP systems. Smaller transfers should use scp or mpscp.

To use SCAMPI refer to ***Quick Start Guide*** on the reverse of this page.

Internal SCAMPI (ISCAMPI) is a system utility to run internal to batch jobs on a given system (no external communication or authentication). Parallel nodes and threads help to accelerate the movement of data internal to an HPCMP asset using native compilers/libraries and high speed networks. The ISCAMPI command should be used in-line in your batch job.

Syntax Instructions

Call the run script, “iscampi.” Use -nh (num hosts) to specify the total number of hosts you wish to use in your transfer. Use the -pph (processes per host) flag to specify the number of processes/threads you wish for each host to run (nh*pph<=128) . The remaining syntax is modeled after the scp command. However, there is no host information allowed as nodes will be selected from your batch pool in your job. The “-l” option should always be used when targeting Lustre file systems as striping parameters will be invoked. Other options:

-s: histogram summary of file size/age
-b x: use x MB block size for reading/writing
-m x: use x MB file size to invoke gang reading/writing
-o: don’t write over existing files

Examples:

- Transfer from home directory on local system to $WORKDIR on the local system using 4 nodes with 6 threads (4*6=24 threads), where $WORKDIR is a Lustre file system (-l), and provide a summary histogram of the files (-s: size/age histograms) using shared version of iscampi:

```
/app/mpiutil/iscampi -l -s -nh 4 -pph 6 $HOME $WORKDIR
```

- Transfer from $WORKDIR on local system to $HOME/results on the local system using 6 nodes (-nh 6), each with 6 threads (-pph 6), where $HOME is a Lustre file system (-l), using a block size of 16MB (-b 16):

```
/app/mpiutil/iscampi -l -b 16 -nh 6 -pph 6 $WORKDIR $HOME/results
```

DOD HPC
DEPARTMENT OF DEFENSE
HIGH PERFORMANCE COMPUTING
MODERNIZATION PROGRAM

ARL DSRC

Quick Start

SCAMPI

Scalable Copy Accelerated by MPI

Introduction

SCAMPI has been deployed to most large unclassified systems in the HPCMP. It is installed at the same location (/app by default) on each system and can be run from: /app/mpiutil/scampi

Syntax instructions

Call the run script, “scampi.” Use -nh (num hosts) to specify the total number of hosts you wish to use in your transfer. Use the -pph (processes per host) flag to specify the number of processes/threads you wish for each host to run. The remaining syntax is modeled after the scp command. Specify paths to or from the remote system using <remotehost>:<desired path>. To specify paths on the local system, simply specify <desired path>

Examples: To clarify, here are some examples on how to use the code.

- This example invokes a transfer from the home directory on the local system to the home directory on Centennial using 2 nodes with 5 threads each. There are a total of 2 nodes involved; one on the sending system and one on the receiving system. So in this example, there will be 1*5 = 5 streams of data.

```
/app/mpiutil/scampi -nh 2 -pph 5  ~/myfile centennial:~
```

- This example invokes a transfer from the home directory on the local system to the home directory on topaz using 4 nodes with 8 threads. There are a total of 4 nodes involved; two on the sending system and two for the receiving system. So in this example there will be 2*8 = 16 streams of data.

```
/app/mpiutil/scampi -nh 4 -pph 8  ~/myfile topaz.erdc.hpc.mil:~
```

Tips:

- The first time you use SCAMPI, you will likely need to manually approve of additions to your ~/.ssh/known_hosts file (this allows ssh to supress subsequent prompts from the target host system).
- Typically, using more than 8 threads per node shows diminishing returns on transfer times.
- Use –y flag to utilize the dynamic load balancing heuristic. This is faster for unstable environments caused by factors such as busy login nodes or WAN transports.
- Specify a value for the –m flag to decide which files to split before transferring in parallel. This example will split any files larger than 512MB.

```
/app/mpiutil/scampi -nh 2 -pph 4 -m 512 ~/mydir centennial:~
```

- Use the -h flag for a list of all currently available options with a short description.

DOD HPC DEPARTMENT OF DEFENSE HIGH PERFORMANCE COMPUTING MODERNIZATION PROGRAM

ARL DSRC

Contact the HPC Help Desk team at: help@helpdesk.hpc.mil