APPENDIX

ARTIFACT DESCRIPTION: IDENTIFICATION OF NETWORK DATA TRANSFER BOTTLENECKS IN HPC SYSTEMS

A. Abstract

This provides information about the systems and tools used to study the relationship between file system activity and data transfer throughput. Because this was partially an observational study (file transfers were generated, but we had no control over the states of the file system and data transfer nodes).

B. Description

1. Check-list
   - Hardware
     - NERSC Cori scratch Lustre file system
       - 248 OSTs, 248 OSSs (each OSS controls one OST)
     - OSTs each have 41 disks and 240 TB capacity
       - Aggregate peak IO performance: 744 GB/s
     - NERSC data transfer nodes
       - Each has four 10-GB Ethernet links
     - Data transfer nodes connected to file system via two FDR Infiniband connections (56 Gbps each)
   - Software: LMT (Lustre monitoring tool), Globus file transfer tool

C. Installation

Code and instructions for installing LMT (Luster monitoring tool): https://github.com/LLNL/lmt/wiki

Instructions for installing Globus file transfer tool: http://toolkit.globus.org/toolkit/docs/latest-stable/admin/install/

D. Experiment workflow

- Create files to read from and write to

```
lfs setstripe –stripe-size [stripe size in megabytes]M –count [number of OSTs to stripe file across] [file name]
```

- Running data transfers

  - Write and Network Transfer

```
#!/bin/bash

file_name=$3
parallelism=$1
buffer_size=$2
#echo $buffer_size
timestamp() {
        date +"%Y-%m-%d %H:%M:%S,%3N"
}

echo
timestamp
start=$SECONDS
globus-url-copy -fast -nodcau -vb -p $parallelism -bs ${buffer_size}MB -t 120 file:////dev/zero gsiftp://dtn02.nersc.gov//$CSCRATCH/${file_name}
end=$SECONDS
duration=$(( end - start ))
echo "took $duration seconds to complete"
timestamp
echo
```

  - Read and Network Transfer
    - Similar to write and network transfer, but input source is a file located in the file system and the output is /dev/null
  - Read, Write, and Network Transfer
    - Similar to other transfers, but both input and output are files within the file system

  - Parameters and values
    - File stripe count (how many OSTs the file is striped across): 1, 4, 16, 248
    - File stripe size (MB): 1, 2, 4
    - Data transfer parallelism (number of threads): 1, 2, 4, 8
    - Data transfer buffer size (MB): 1, 2, 4
- Calculate file system activity using Pytokio/LMT
  - Pytokio: tool developed to help with analyzing IO data from large computing systems
    - Within examples folder, the heat map tool was used to calculate aggregate file IO rates
    - Pytokio package available at https://github.com/NERSC/pytokio/wiki
  - Without Pytokio (which was written for NERSC and requires file system access), with only the raw LMT logs requires writing code to parse the log files. Each row is a timestamp, each column is an OST; the data collected is the average number of bytes transferred over a 5 second interval, sampled every 5 seconds.

E. Evaluation and expected result

- Data recorded from data transfers:
  - From computer: start time, end time, duration
  - From globus tool: number of bytes
- Throughput calculated using number of bytes/duration

- File system IO rates using LMT
  - For different number of OSTs, command `lfs getstripe [file name]` is used to get which OSTs the file is located on
  - In the LMT log files each column is an OST; with the list of OSTs it is possible to select only the relevant columns to calculate file IO rates
  - Within time range specified for each data transfer, relevant OST columns were selected and the file IO rates were added up, multiplied by 5 (because the sampling rate is once every 5 seconds), then divided by the duration to calculate the aggregate file IO rate