Hermes 0.1.0 Beta Webinar

1 Building Hermes

- Docker Image
- CMake
- Spack
- More info in the README

2 Example Deployments

The example output below was produced with the following simple Hermes daemon program:

```
#include <cstdlib>
#include <mpi.h>
#include "hermes.h"

int main(int argc, char* argv[]) {
  MPI_Init(&argc, &argv);
  char* hermes_config = getenv("HERMES_CONF");
  auto hermes = hermes::InitHermesDaemon(hermes_config);
  hermes->RunDaemon();
  MPI_Finalize();
  return 0;
}
```

This daemon program can be found in the `bin` directory of your Hermes installation.

2.1 1 Tier, 1 Slab

2.1.1 Config file

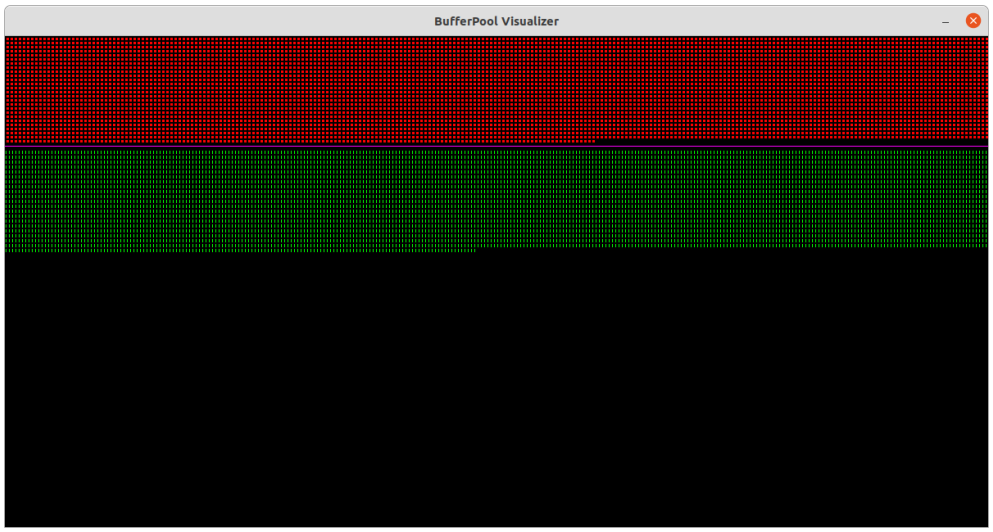
```
num_devices = 2;
capacities_mb = {16, 6144};
```

```
block_sizes_kb = {4, 1024};
num_slabs = {1, 1};
slab_unit_sizes = {
  {1},
  {1},
 };
 desired_slab_percentages = {
   {0.0},
   {1.0},
 };
```

2.1.2 Output

```
I1130 19:43:03.381317  6735 buffer_pool.cc:1074] 246824 bytes requi
I1130 19:43:03.381330  6735 buffer_pool.cc:1113] Device: 0
I1130 19:43:03.381335  6735 buffer_pool.cc:1116]     0-Buffers: 0
I1130 19:43:03.381340  6735 buffer_pool.cc:1121]     Num Headers: 0
I1130 19:43:03.381348  6735 buffer_pool.cc:1122]     Num Buffers: 0
I1130 19:43:03.381353  6735 buffer_pool.cc:1113] Device: 1
I1130 19:43:03.381357  6735 buffer_pool.cc:1116]     0-Buffers: 614
I1130 19:43:03.381366  6735 buffer_pool.cc:1121]     Num Headers: 6
I1130 19:43:03.381372  6735 buffer_pool.cc:1122]     Num Buffers: 6
I1130 19:43:03.381388  6735 buffer_pool.cc:1124] Total Buffers: 614
I1130 19:43:03.381963  6735 metadata_storage_stb_ds.cc:912] Metadat
I1130 19:43:03.391978  6735 rpc_thallium.cc:53] Serving at ofi+sock
I1130 19:43:03.498203  6735 rpc_thallium.cc:503] Buffer organizer s
```

2.1.3 Visualization



2.2 4 Tiers, Multiple Slabs

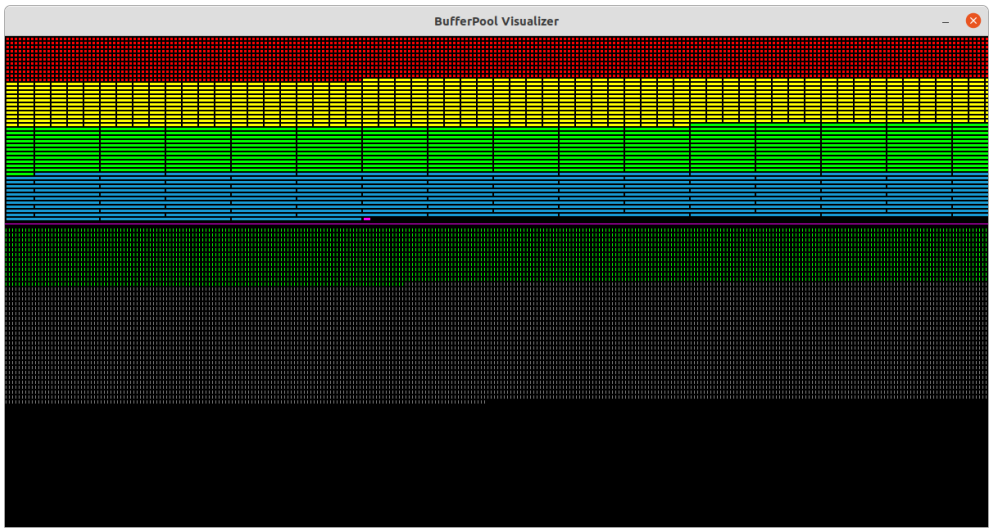
2.2.1 Config file

```
num_devices = 4;
capacities_mb = {50, 50, 50, 50};
block_sizes_kb = {4, 4, 4, 4};
num_slabs = {4, 4, 4, 4};
slab_unit_sizes = {
  {1, 4, 16, 32},
  {1, 4, 16, 32},
  {1, 4, 16, 32},
  {1, 4, 16, 32},
};
desired_slab_percentages = {
  {0.25, 0.25, 0.25, 0.25},
  {0.25, 0.25, 0.25, 0.25},
  {0.25, 0.25, 0.25, 0.25},
  {0.25, 0.25, 0.25, 0.25},
};
```

2.2.2 Output

```
I1130 19:59:49.809983  7462 buffer_pool.cc:1074] 953088 bytes requi
I1130 19:59:49.810103  7462 buffer_pool.cc:1113] Device: 0
I1130 19:59:49.810113  7462 buffer_pool.cc:1116]     0-Buffers: 248
I1130 19:59:49.810119  7462 buffer_pool.cc:1116]     1-Buffers: 680
I1130 19:59:49.810125  7462 buffer_pool.cc:1116]     2-Buffers: 170
I1130 19:59:49.810132  7462 buffer_pool.cc:1116]     3-Buffers: 85
I1130 19:59:49.810137  7462 buffer_pool.cc:1121]     Num Headers: 1
I1130 19:59:49.810146  7462 buffer_pool.cc:1122]     Num Buffers: 3
I1130 19:59:49.810153  7462 buffer_pool.cc:1113] Device: 1
I1130 19:59:49.810160  7462 buffer_pool.cc:1116]     0-Buffers: 320
I1130 19:59:49.810166  7462 buffer_pool.cc:1116]     1-Buffers: 800
I1130 19:59:49.810173  7462 buffer_pool.cc:1116]     2-Buffers: 200
I1130 19:59:49.810181  7462 buffer_pool.cc:1116]     3-Buffers: 100
I1130 19:59:49.810187  7462 buffer_pool.cc:1121]     Num Headers: 4
I1130 19:59:49.810196  7462 buffer_pool.cc:1122]     Num Buffers: 4
I1130 19:59:49.810215  7462 buffer_pool.cc:1113] Device: 2
I1130 19:59:49.810220  7462 buffer_pool.cc:1116]     0-Buffers: 320
I1130 19:59:49.810223  7462 buffer_pool.cc:1116]     1-Buffers: 800
I1130 19:59:49.810230  7462 buffer_pool.cc:1116]     2-Buffers: 200
I1130 19:59:49.810235  7462 buffer_pool.cc:1116]     3-Buffers: 100
I1130 19:59:49.810238  7462 buffer_pool.cc:1121]     Num Headers: 4
I1130 19:59:49.810245  7462 buffer_pool.cc:1122]     Num Buffers: 4
I1130 19:59:49.810251  7462 buffer_pool.cc:1113] Device: 3
I1130 19:59:49.810258  7462 buffer_pool.cc:1116]     0-Buffers: 320
I1130 19:59:49.810266  7462 buffer_pool.cc:1116]     1-Buffers: 800
I1130 19:59:49.810271  7462 buffer_pool.cc:1116]     2-Buffers: 200
I1130 19:59:49.810278  7462 buffer_pool.cc:1116]     3-Buffers: 100
I1130 19:59:49.810285  7462 buffer_pool.cc:1121]     Num Headers: 4
I1130 19:59:49.810289  7462 buffer_pool.cc:1122]     Num Buffers: 4
I1130 19:59:49.810297  7462 buffer_pool.cc:1124] Total Buffers: 163
I1130 19:59:49.812152  7462 memory_management.cc:141] PushSize adde
I1130 19:59:49.812175  7462 metadata_storage_stb_ds.cc:912] Metadat
I1130 19:59:49.820807  7462 rpc_thallium.cc:53] Serving at ofi+sock
I1130 19:59:49.926741  7462 rpc_thallium.cc:503] Buffer organizer s
```

2.2.3 Visualization



3 Running IOR with Hermes

3.1 PFS Baseline

3.1.1 Client Info

- 4 nodes
- Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz
- 40 cores (Hyperthreading enabled)
- 46 GiB DRAM
- 40 Gbps ethernet with RoCE capability

3.1.2 Server Info

- 8 nodes
- OrangeFS backed by HDDs

3.1.3 Run

```
set -x

NPROCS=48
PROCS_PER_NODE=12
OUTPUT_FILE=/mnt/nvme/${USER}/pfs/ior.out

mpirun -n ${NPROCS} -ppn ${PROCS_PER_NODE} \
  ior -w -o ${OUTPUT_FILE} -t 1m -b 128m -F -e -Y -O summaryFormat=CSV
```

Here we launch 48 IOR processes across 4 nodes. Explanations of the IOR options follow.

w
: Perform write

o
: Output/Input file

t
: Size per write

b
: Total I/O size per rank

F
: Create one file for each process

e
: Call `fsync` on file close.

Y
: Call `fsync` after each write.

O summaryFormat
: Show the output in a compact, CSV format

3.1.4 Results

```
access,bw(MiB/s),IOPS,Latency,block(KiB),xfer(KiB),open(s),wr/rd(s)
write,29.0104,29.0280,1.4748,131072.0000,1024.0000,28.1393,211.6576
```

Here's the same run, but without the `-Y` flag:

```
access,bw(MiB/s),IOPS,Latency,block(KiB),xfer(KiB),open(s),wr/rd(s)
write,471.6021,475.6229,0.0318,131072,1024,10.376,12.9178,8.813,13.
```

3.2 1 Tier: Burst Buffer

OrangeFS on 2 nodes backed by SSDs.

3.2.1 Configuration file

```
num_devices = 2;
capacities_mb = {16, 6144};
block_sizes_kb = {4, 1024};
num_slabs = {1, 1};
slab_unit_sizes = {
  {1},
  {1},
};
desired_slab_percentages = {
  {0.0},
  {1.0},
};
mount_points = {"", "/mnt/nvme/chogan/bb"};
```

3.2.2 Run with Hermes

1. Start a daemon
2. `LD_PRELOAD` the Hermes adapter library

```
set -x
HERMES_DIR=${HOME}/dev/hermes
HERMES_BIN_DIR=${HERMES_DIR}/build/bin
HERMES_POSIX_SO=${HERMES_BIN_DIR}/libhermes_posix.so
HERMES_CONF_PATH=/mnt/common/${USER}/dev/misc_hdf/webinar_12_01_21/ior_
OUTPUT_FILE=/mnt/nvme/${USER}/pfs/ior.out
NPROCS=48
PROCS_PER_NODE=12
NUM_NODES=4
HOSTS=ares-comp-25,ares-comp-26,ares-comp-27,ares-comp-28

# Start 1 daemon on each node
mpirun -n ${NUM_NODES} -ppn 1 -hosts ${HOSTS} \
   -genv HERMES_CONF ${HERMES_CONF_PATH} \
   ${HERMES_BIN_DIR}/hermes_daemon &

# Give the daemons a chance to initialize
sleep 3

# Run IOR
mpirun -n ${NPROCS} -ppn ${PROCS_PER_NODE} -hosts ${HOSTS} \
   -genv LD_PRELOAD ${HERMES_POSIX_SO} \
   -genv HERMES_CONF ${HERMES_CONF_PATH} \
   -genv HERMES_WRITE_ONLY 1 \
   -genv ADAPTER_MODE SCRATCH \
   ior -w -o ${OUTPUT_FILE} -t 1m -b 128m -F -e -Y -O summaryFormat=CSV
```

3.2.3 Results

```
access,bw(MiB/s),IOPS,Latency,block(KiB),xfer(KiB),open(s),wr/rd(s)
write,335.2401,337.9022,0.1326,131072.0000,1024.0000,1.6561,18.1828
```

3.3 BB + Local NVMe

3.3.1 Configuration file

```
num_devices = 3;
capacities_mb = {16, 2048, 4096};
block_sizes_kb = {4, 1024, 1024};
num_slabs = {1, 1, 1};
slab_unit_sizes = {
  {1},
  {1},
  {1},
};
desired_slab_percentages = {
  {0.0},
  {1.0},
  {1.0},
};
mount_points = {"", "/mnt/nvme/chogan/local", "/mnt/nvme/chogan/bb"};
```

3.3.2 Running

Same as before

3.3.3 Results

```
access,bw(MiB/s),IOPS,Latency,block(KiB),xfer(KiB),open(s),wr/rd(s)
write,514.6819,517.2259,0.0776,131072.0000,1024.0000,0.1132,11.8788
```

3.4 BB + NVMe + RAM

3.4.1 Configuration file

```
capacities_mb = {512, 1536, 4096};
desired_slab_percentages = {
  {1.0},
  {1.0},
  {1.0},
};
```

3.4.2 Running

Same as before

3.4.3 Results

```
access,bw(MiB/s),IOPS,Latency,block(KiB),xfer(KiB),open(s),wr/rd(s)
write,829.3921,836.1266,0.0511,131072.0000,1024.0000,0.1179,7.3482,
```

4 Summary of Results