

DDN ExaScalerのさまざまな 性能改善について

DDN Storage
Shuichi Ihara
2020/10/09



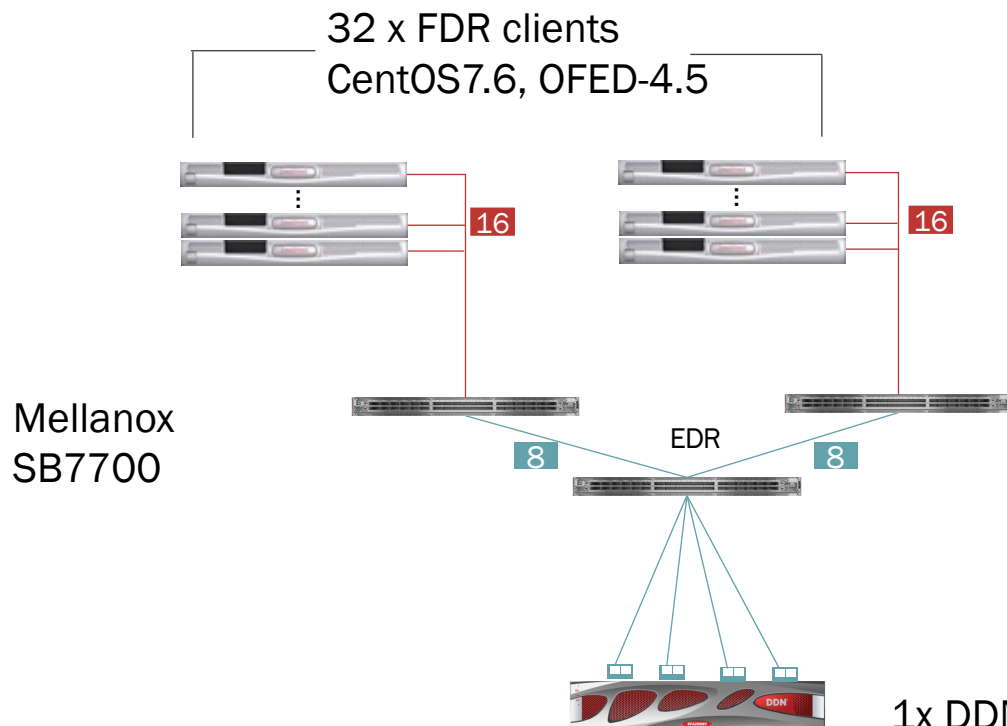
はじめに

- Lustreは継続して性能改善
- 性能比較を紹介
 - Lustre version 2.10.8, 2.12.2 とmaster branch (2.13)
- DDNはこれらの性能改善をExaScaler5.1および5.2に取り込み

本資料でカバーするテストケースおよびワークロード

- オールフラッシュにおける性能比較
 - アグリゲートスループット
 - FPP (File Per Process), Strided SSF (Single Shared File) and Strided SSF Hard
 - 1MB IO Size on FPP and Strided and 47008 Byte IO size for Strided SSF Hard
 - アグリゲートIOPS(4K ランダムリード)
 - シングルOSS性能
- シングルクライアント性能比較
 - シングルスレッド
 - マルチレールを使ったクライアント
 - 4Kランダムリード IOPS
- GPU DIRECTストレージ

ベンチマーク構成



1x DDN AI400NV with 20 x 3.84TB SSD
Mixed 4 x OSS and MDS

• テスト条件

- サーバとクライアントは同じバージョン
- 同じLustreのパラメータ

```
# lctl set_param obdfilter.*.brw_size=16  
# lctl set_param osc.*.max_rpcs_in_flight=16  
# lctl set_param osc.*.checksums=0  
# lctl llite.*.max_read_ahead_mb=2048
```

オールフラッシュにおける性能比較

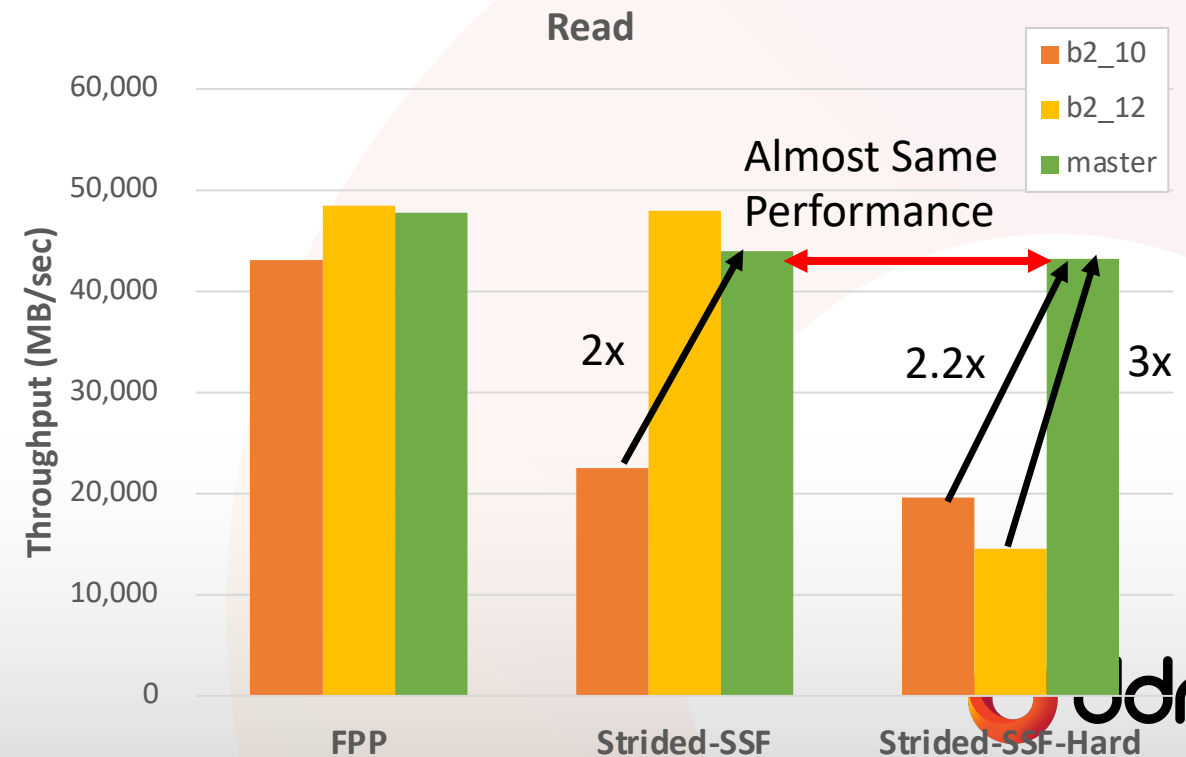
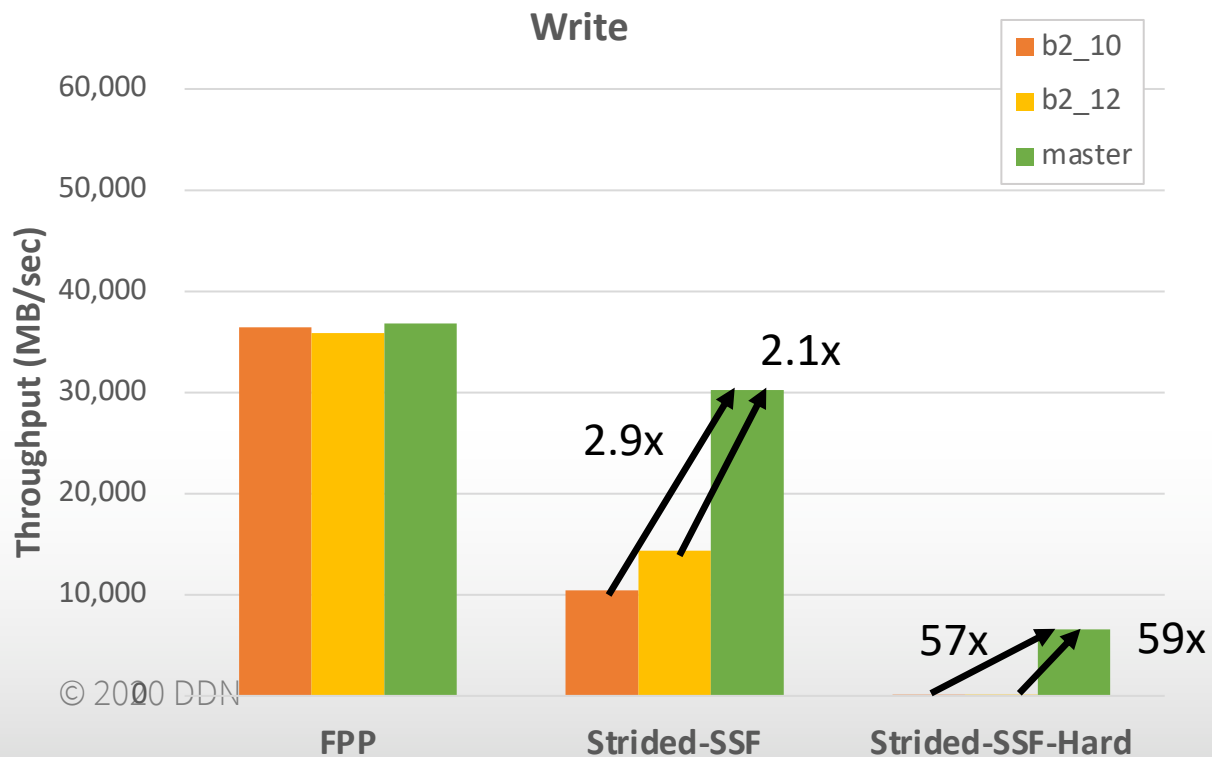
アグリゲートスループット比較

FPP (File Per Process), Strided-SSF (Single Shared File), Strided-SSF-Hard

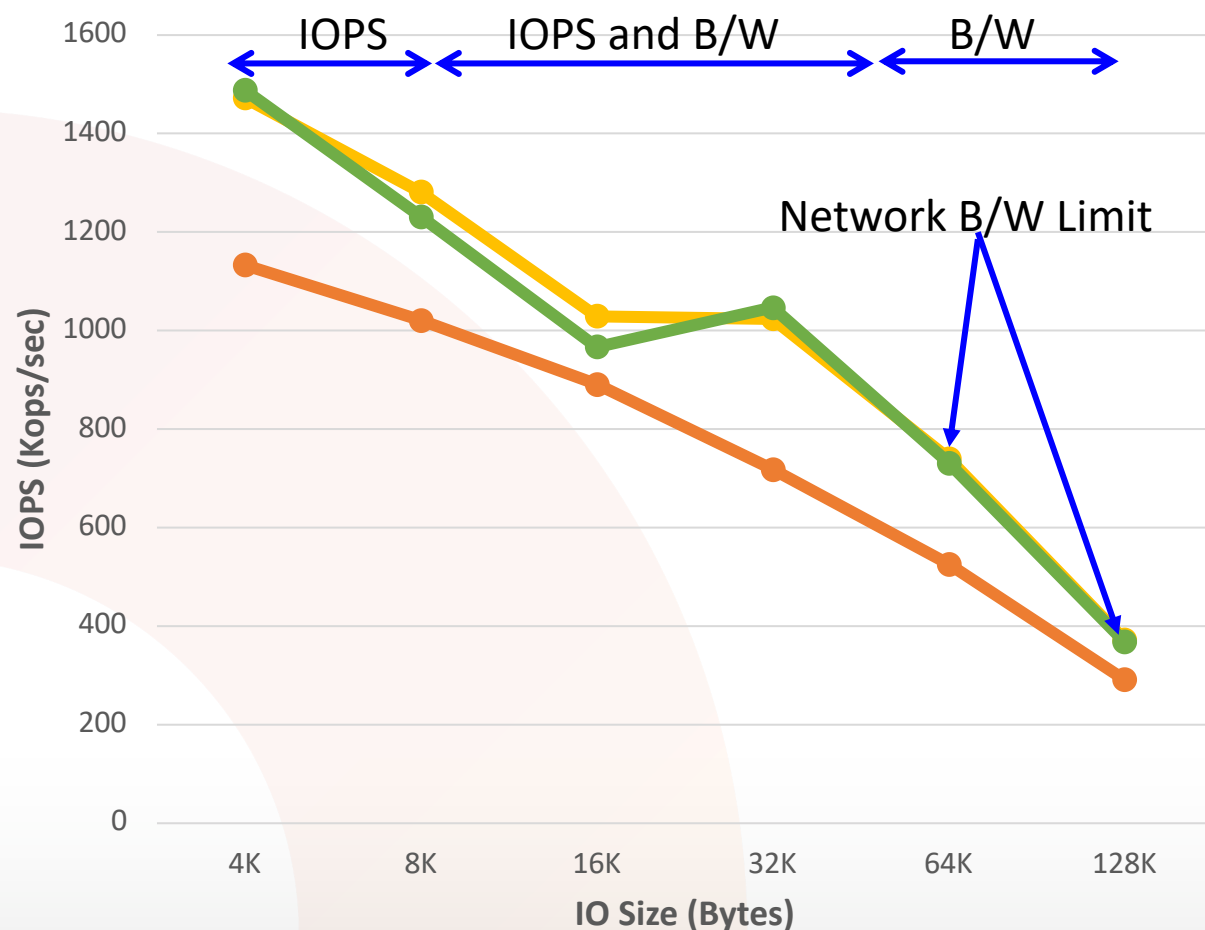
FPP: `ior -w/-r -s 6600 -i 1 -C -Q 1 -g -G 27 -k -e -t 1m -b 1m -F`

Strided-SSF: `ior -w/-r -s 6600 -i 1 -C -Q 1 -g -G 27 -k -e -t 1m -b 1m`

Strided-SSF-Hard: `ior -w/-r -s 132000 -i 1 -C -Q 1 -g -G 27 -ke -t 47008 -b 47008`



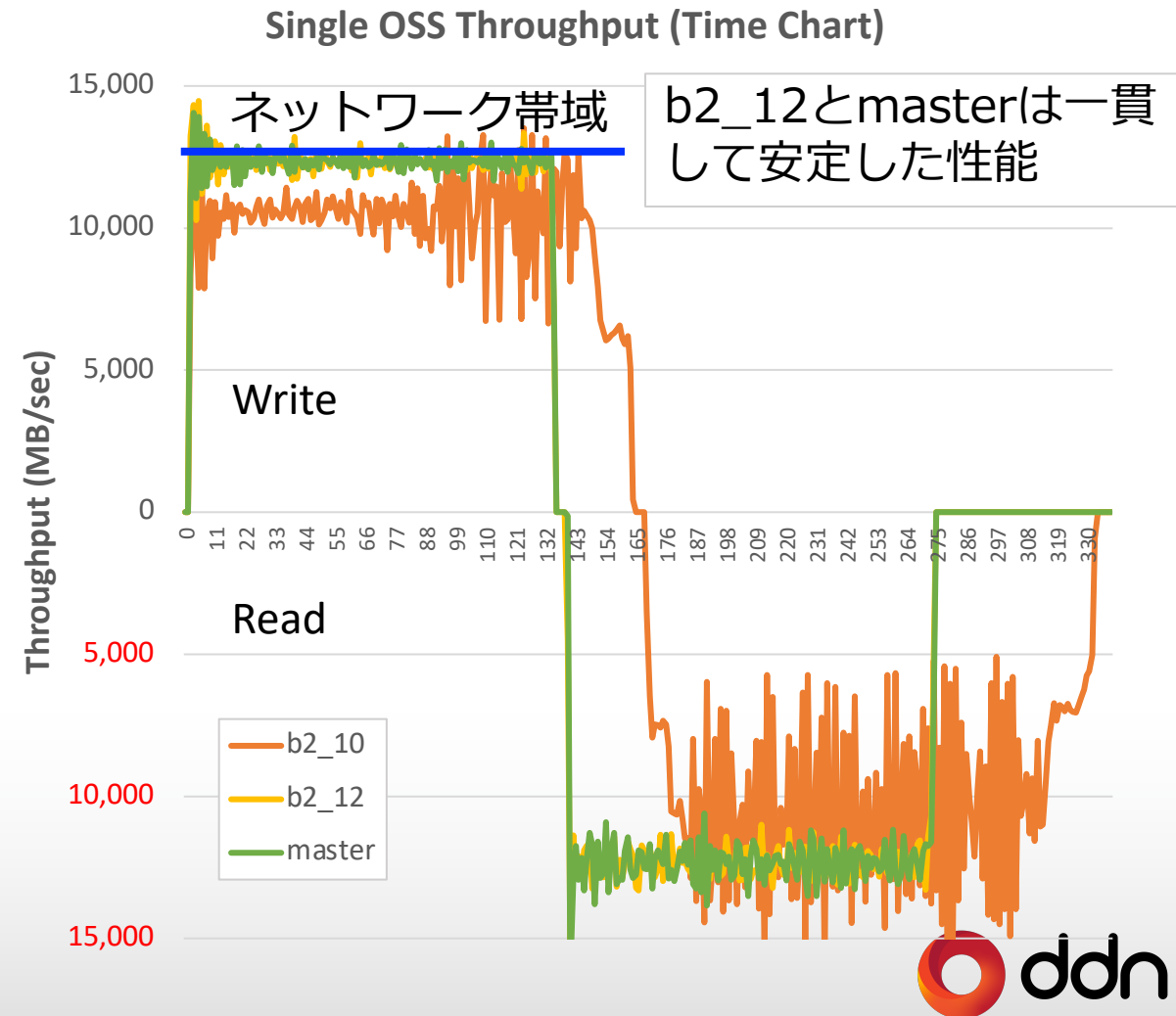
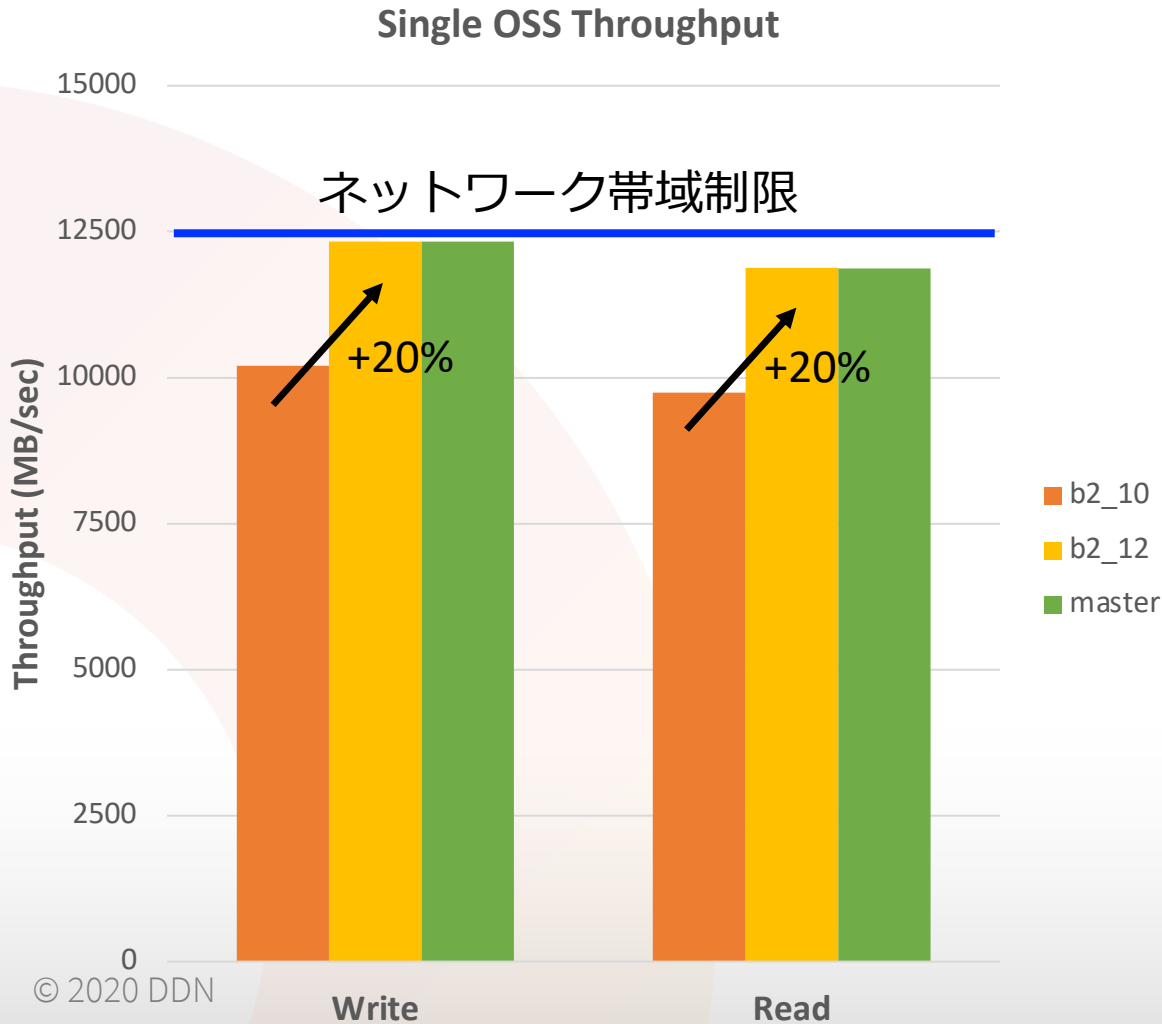
アグリゲートIOPS(4K ランダムリード, O_DIRECT)



- ワークロードはRandom O_DIRECTリード(fio)
 - 異なるIOサイズで比較
- 32クライアント、1536 プロセス
- b2_12とmasterはb2_10よりも30%高速
- b2_12とmasterはネットワーク帯域に達成
 - 4 x IB-EDR帯域 ≈ 48GB/sec (64K IOサイズ)

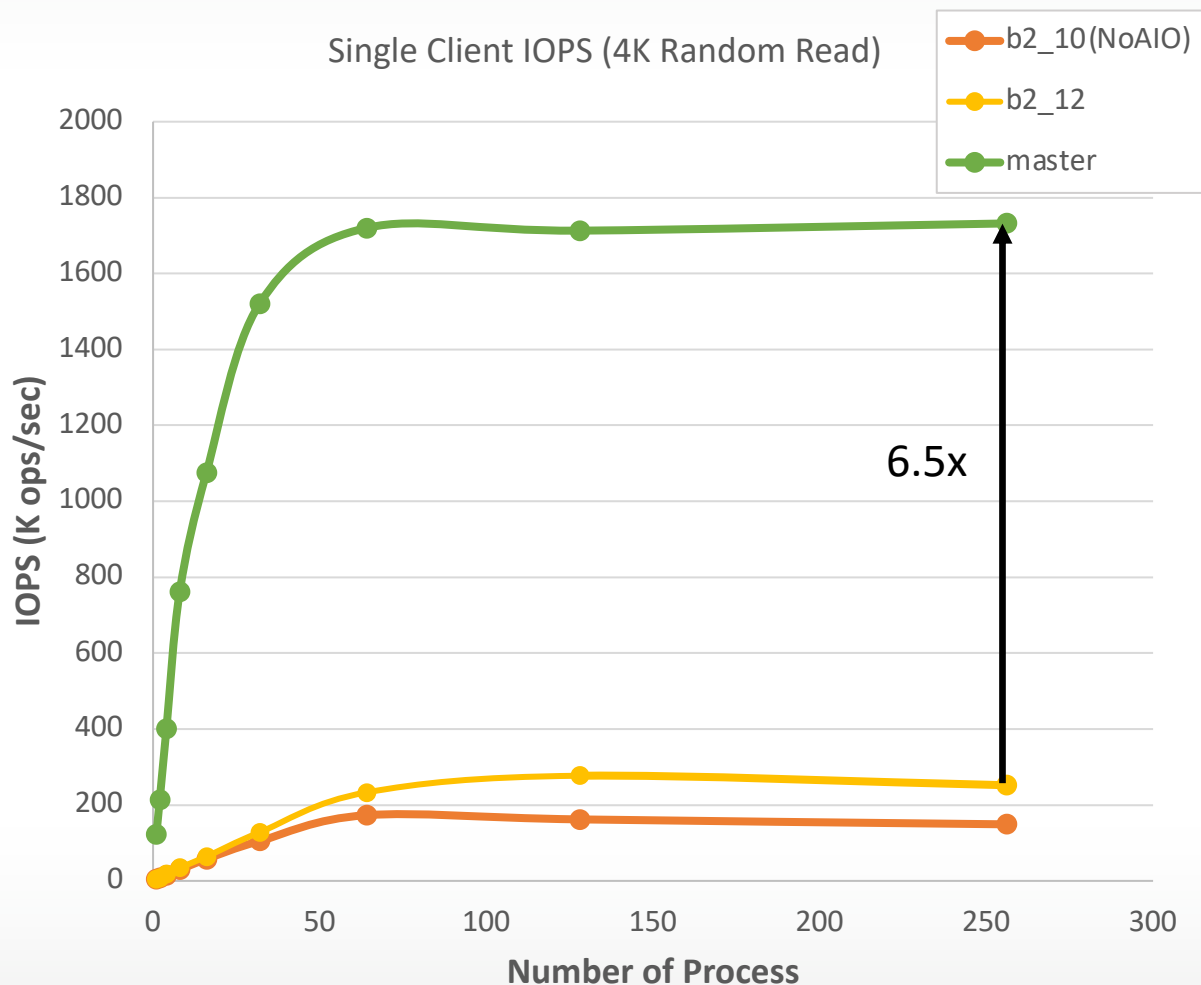
シングルOSS性能

高多重アクセス N:1 (32クライアント、768プロセス)



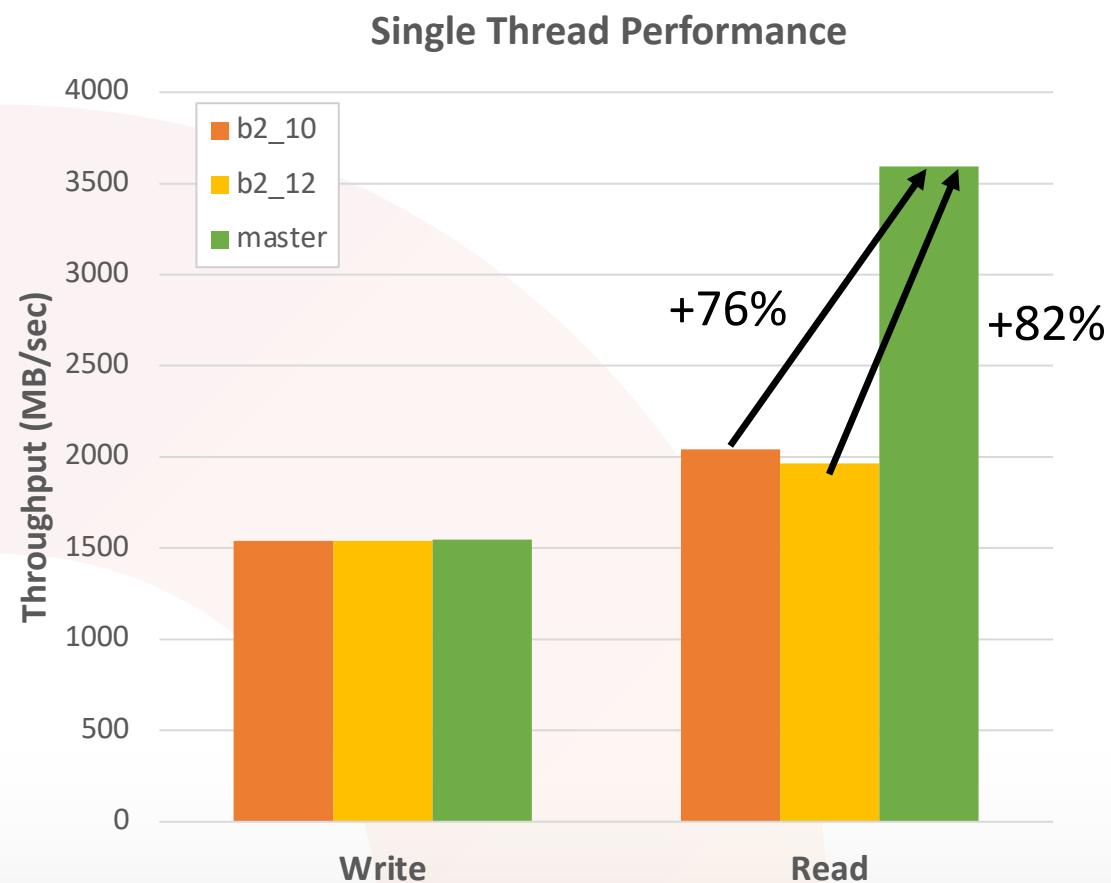
シングルクライアント性能比較

シングルクライアントIOPSスケーラビリティ (4Kランダムリード, O_DIRECT, AIO)



- "fio"を使ってO_DIRECT+AIO READ
 - b2_1 クライアントにて様々なプロセス数を比較
- 10 はO_DIRECT+AIOをサポートしない
- b2_12 はAIO動作するが同期モード
- master branch はO_DIRECT+AIOをフルサポート
- master は1 クライアントにて1.7M 4K Random Readを達成(利用するCPUによってさらに向上)

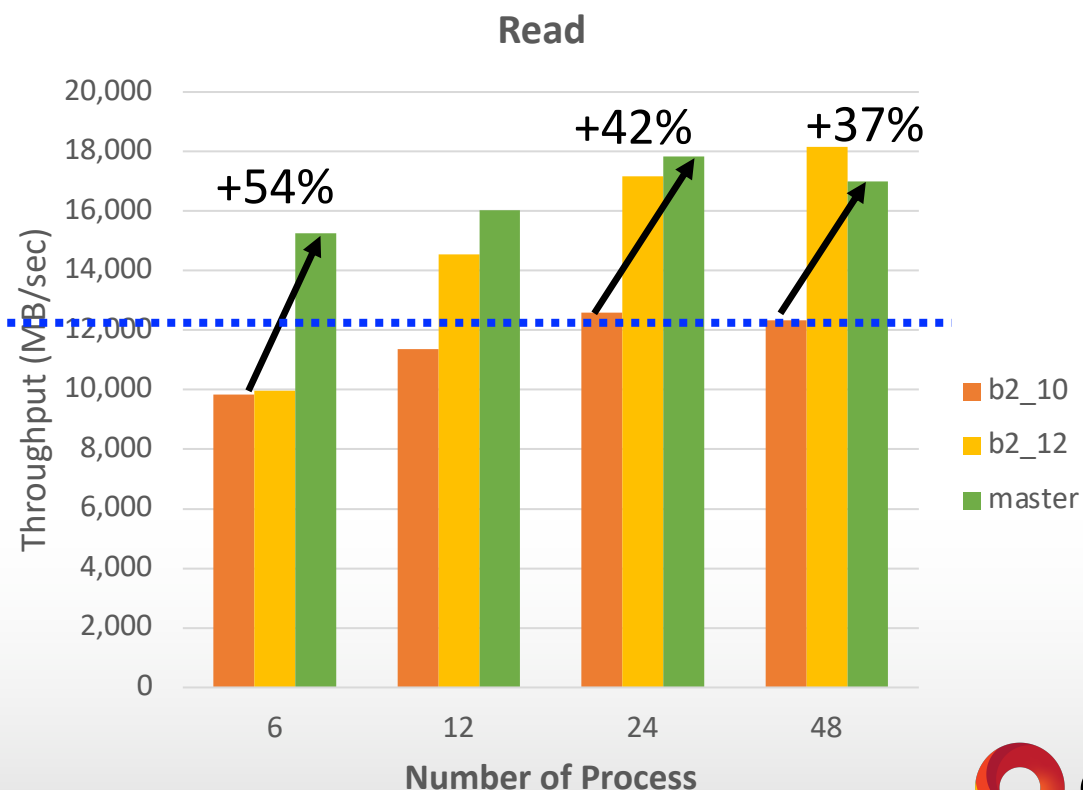
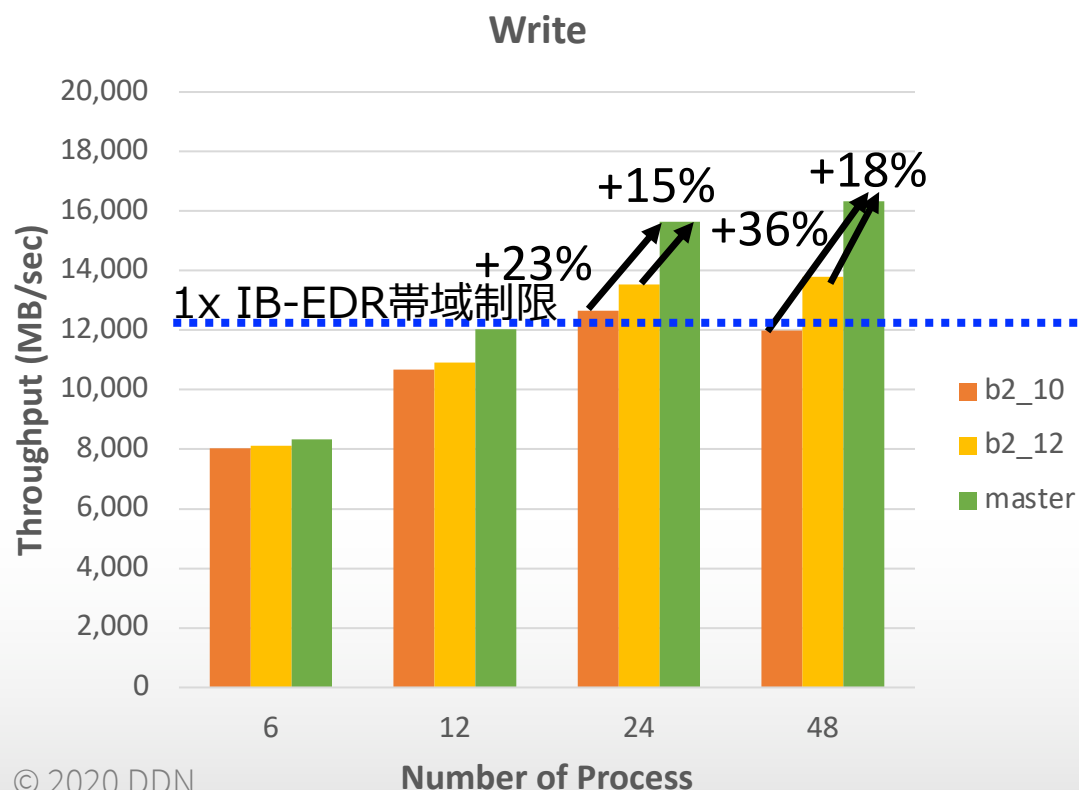
シングルクライアントにおけるシングルスレッドスループット



- 1MBシングルストリーム
IOR -w (-r) -k -t 1m -b 384g -e -vv -o file
- WriteとReadの間でページキャッシュ破棄
- Lustreファイルストライピングの有効
lfs setstripe -c -1 -S 16M
- Readにおいてmasterは2.10/2.12と比べて 76-82%高速
- Writeはどのブランチも同じ性能(利用するCPUに依存)

マルチレール構成のシングルクライアントスループット性能

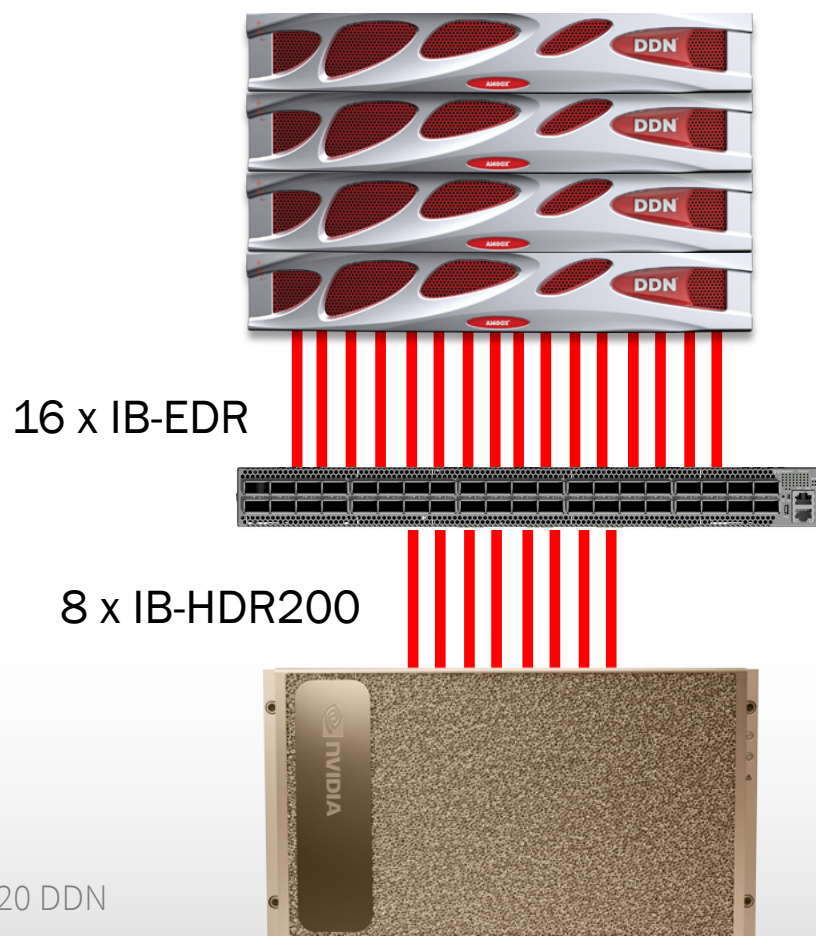
- 高ネットワーク帯域(2 x IB-EDR)のクライアント(LustreのMulti-Rail構成)
- “File Per Process”ワークロード (合計 768GB ファイル)
- クライアント性能のボトルネックが1 x IB-EDR帯域によって見えない



GPUダイレクトストレージ



ベンチマーク構成

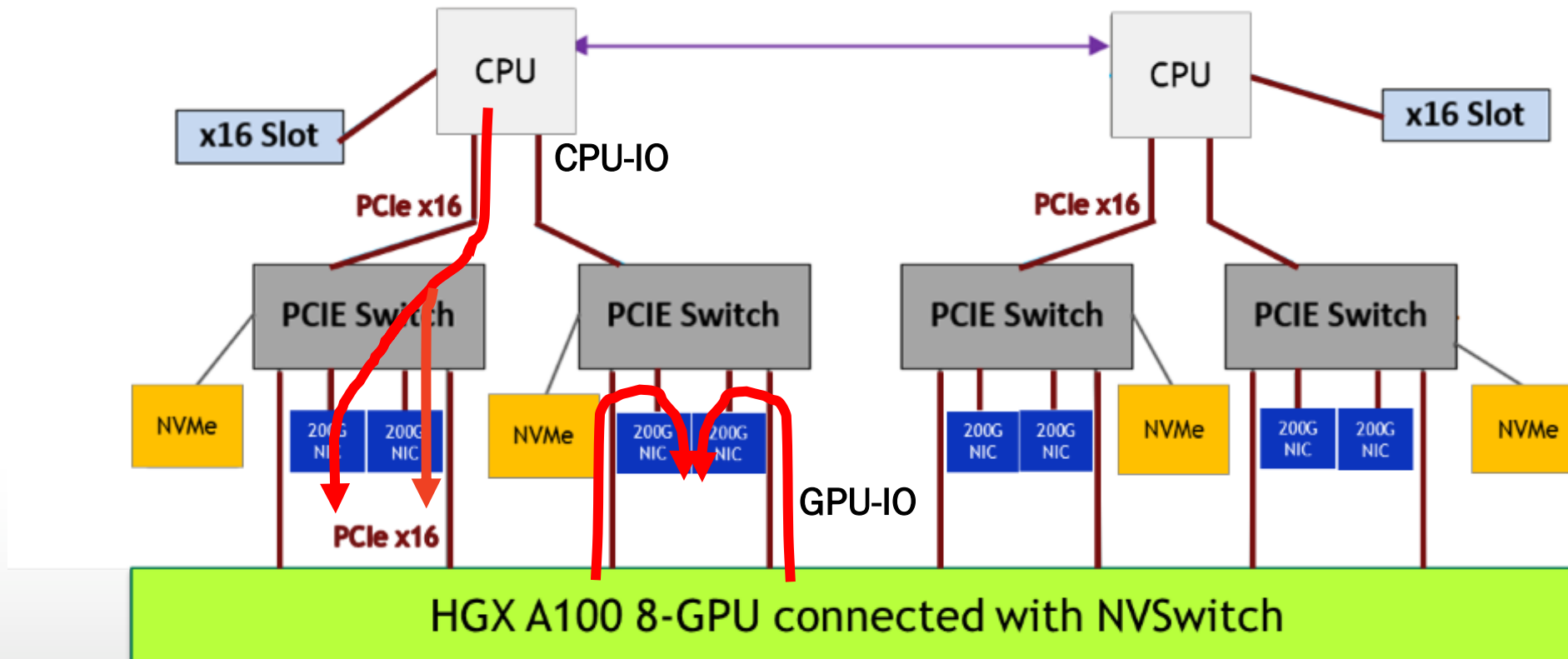


- 4 x DDN AI400
 - 20 x 3.84TB Samsung NVME
 - 4 x EDR
 - Exascaler 5.1.1
- 1 x NVIDIA GDX A100
 - 8 x NVIDIA A100 GPU
 - 8 x HDR200
 - Ubuntu18.04
 - Exascaler 5.1.1
- 1 x Mellanox QM8700 HDR200 Switch

CPU-IO とGPU DIRECTストレージを使ったGPU-IO

GPU DIRECT STORAGE

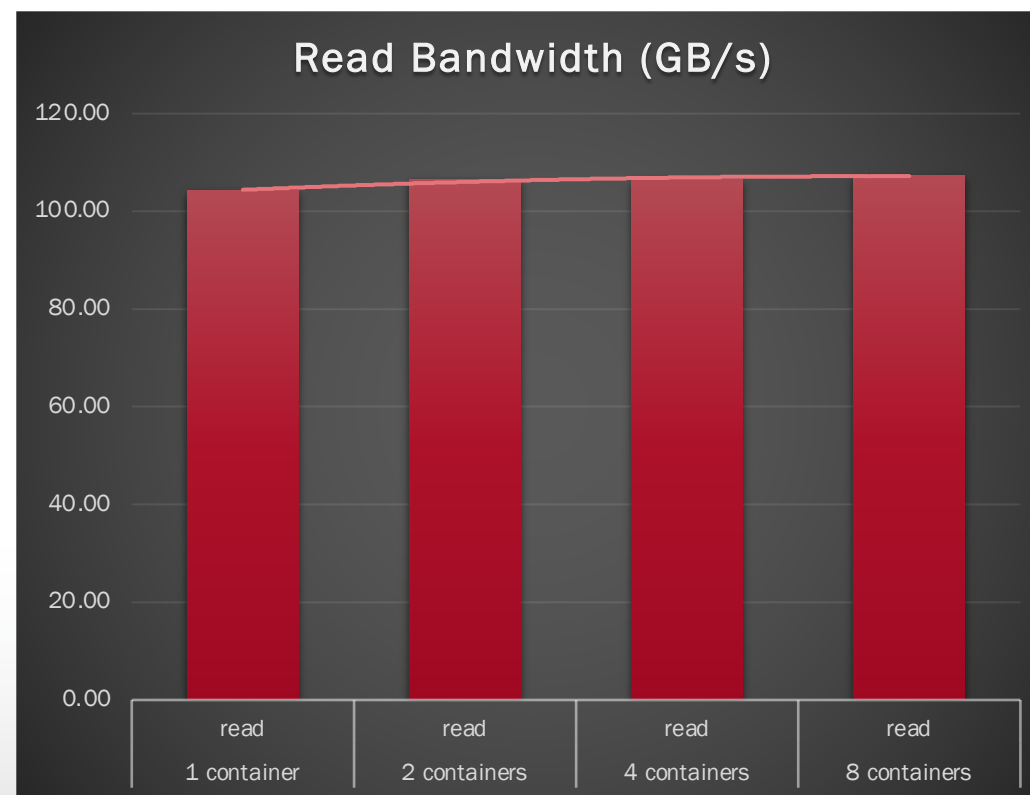
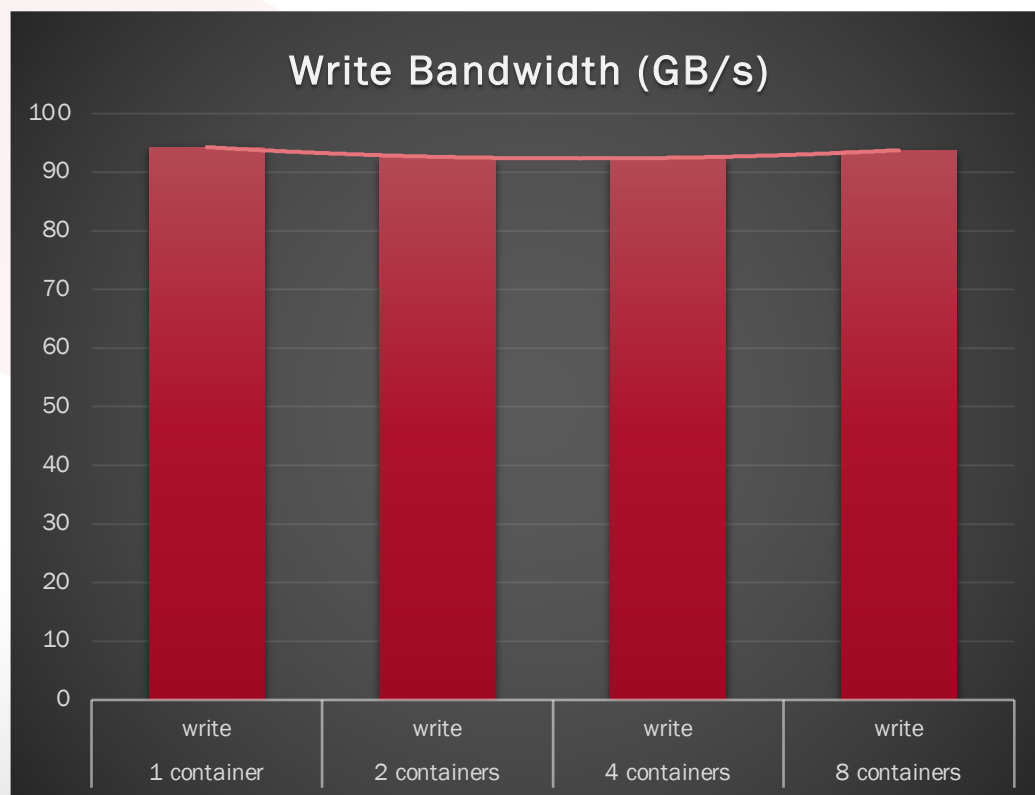
- 不必要なメモリコピーを排除し低レイテンシーを実現
- ホストCPUやメモリサブシステムの消費量が減少
- ストレージとGPU間のBWデータ転送が高速化
- AI、深層学習、HPCアプリケーションの性能向上



ベンチマーク結果 - シーケンシャルスループット(CPU-IO)

- Dockコンテナの中で”fio”を実行

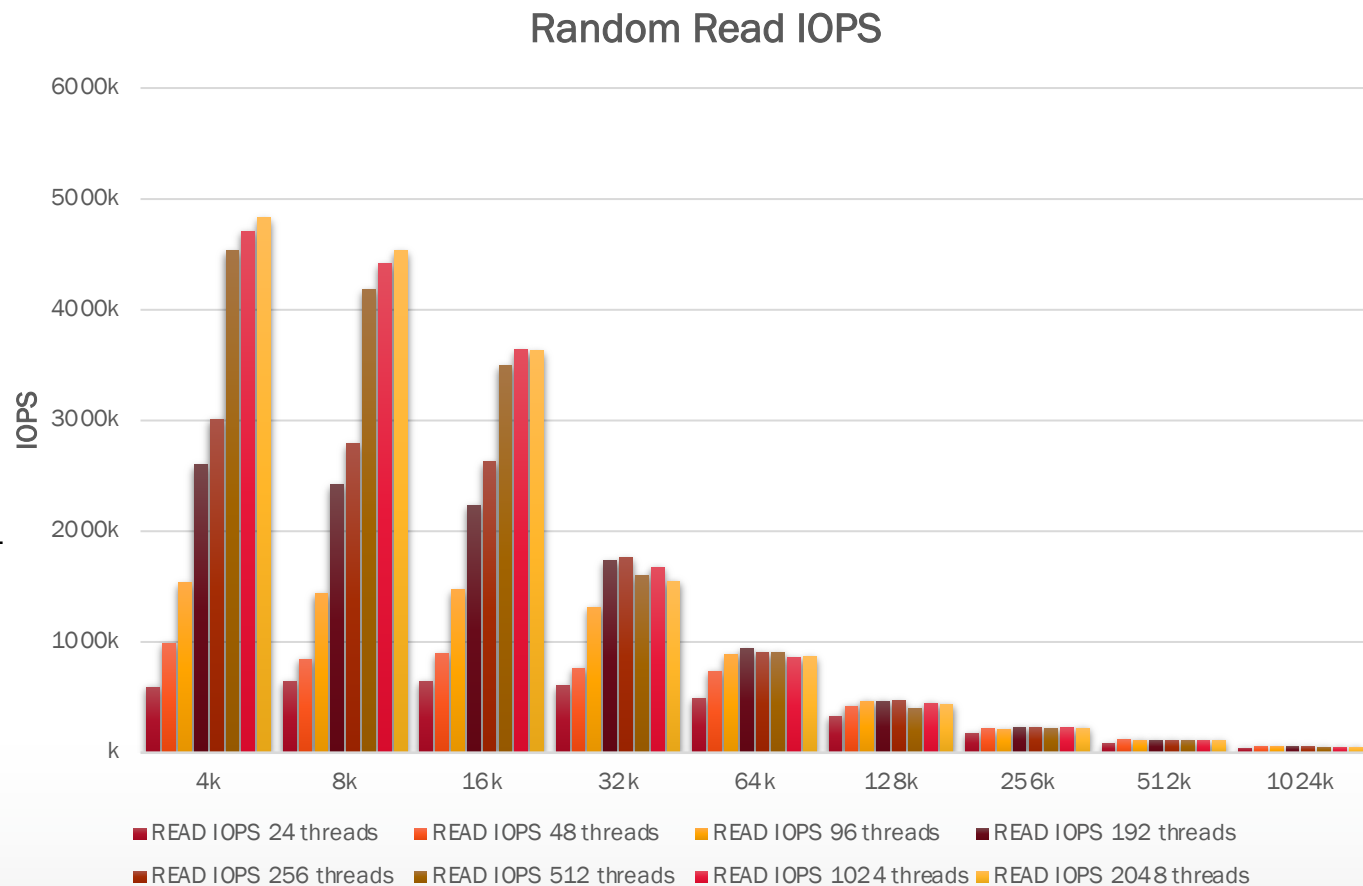
```
# fio --create_serialize=0 --group_reporting=0 --ioengine=libaio --direct=1 --norandommap=0 --fallocate=none --output-format=json,normal --time_based --ramp_time=30s --runtime=1m --readwrite=read --bs=16M --size=10G --iodepth=16 --name=c01file --directory=/lustre/ai400x/client1/bm_bw/cont1/ --numjobs=32
```



最大性能: Write 94GB/sec, Read 107GB/sec

ベンチマーク結果 - ランダムリード IOPS(CPU-IO)

- 異なるIOサイズのランダムリード
 - 4KBから1MBのIOサイズ
 - 複数のIOスレッド数
- "fio" にてワークロードを生成
 - ```
python3 run_fio.py --iteration 3 --docker --iosize 4k 8k 16k 32k 64k 128k 256k 512k 1024k --ncontainer 1 --
iotype randread randwrite --oss_ips $OSSIPS --runtime $RUNTIME --ramptime ${RAMPTIME} --job_size $JOB_SIZE --njobs 24 48 96 192 320 --iodepth 128 --
filepath $BASENAME/client1/bm_io/cont{}/ --nmount 1 --
log_path results_test_iops_1cont_1mount -cleancache
```



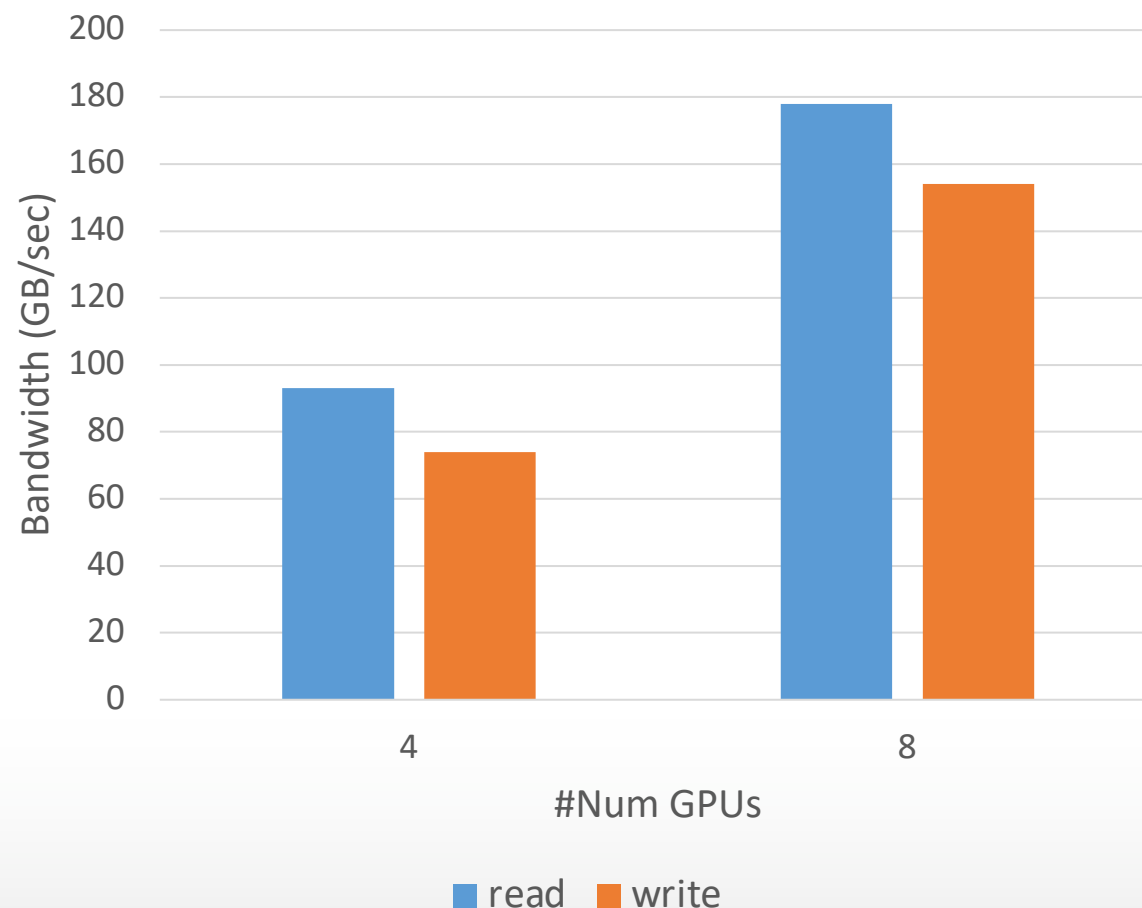
# ベンチマーク結果 - シーケンシャルスループット(GPU-IO)

- “gdsio”にてテスト実施

- 1MB IO Size and 128 Worker threads

```
PATHGDSIO="/home/ddn/raphael/GDS/gds_0_7/gds-beta-v0.7.1/tools/"
FS="/lustre/ai400x/gds_ib0/multi-rail/gpu"
WORKERS=128; IO_TYPE=0; XFER_TYPE=0;
IO_SIZE=1M; DATASET_SIZE=4G
${PATHGDSIO}/gdsio -T 120 ¥
-D ${FS}/md0 -d 0 -n 0 -w $WORKERS -s $DATASET_SIZE
-i $IO_SIZE -x $XFER_TYPE -I $IO_TYPE ¥
-D ${FS}/md1 -d 1 -n 1 -w $WORKERS -s $DATASET_SIZE
-i $IO_SIZE -x $XFER_TYPE -I $IO_TYPE ¥
-D ${FS}/md2 -d 2 -n 2 -w $WORKERS -s $DATASET_SIZE
-i $IO_SIZE -x $XFER_TYPE -I $IO_TYPE ¥
-D ${FS}/md3 -d 3 -n 3 -w $WORKERS -s $DATASET_SIZE
-i $IO_SIZE -x $XFER_TYPE -I $IO_TYPE ¥
-D ${FS}/md4 -d 4 -n 4 -w $WORKERS -s $DATASET_SIZE
-i $IO_SIZE -x $XFER_TYPE -I $IO_TYPE ¥
-D ${FS}/md5 -d 5 -n 5 -w $WORKERS -s $DATASET_SIZE
-i $IO_SIZE -x $XFER_TYPE -I $IO_TYPE ¥
-D ${FS}/md6 -d 6 -n 6 -w $WORKERS -s $DATASET_SIZE
-i $IO_SIZE -x $XFER_TYPE -I $IO_TYPE ¥
-D ${FS}/md7 -d 7 -n 7 -w $WORKERS -s $DATASET_SIZE
-i $IO_SIZE -x $XFER_TYPE -I $IO_TYPE ¥
```

GDS IO results - DGX-A100 - 4xAI400X





ddn