2021年9月 更新版

Type IIサブシステム向けMPI環境の整備について

概要

- Type IIサブシステムにおけるMPI環境（OpenMPI）の整理を行いました
- CUDAの各バージョンに対応したOpenMPI（とHPC-X）のmoduleを用意しましたのでGPUとMPIを組み合わせて利用したい方はご利用ください

- OpenMPIとHPC-Xについて
  - NVIDIA社が提供するHPC-Xを使うと通常のOpenMPIよりも高速な通信が行えることがあるため、moduleを用意しました。
  - OpenMPIとHPC-Xのどちらが高性能かは通信パターンや利用する通信関数によって決まるため、高い性能を得るには利用者自身で使い分ける必要があります。
  - 各moduleを用いた際の性能の例と傾向分析結果を提供しますので、役立ててください。

ディレクトリ・ファイル構成

- TypeII_サブシステム向けMPI環境の整備について_202109.pdf　この資料と同じもの
- テストプログラムによる基本的な通信性能（ディレクトリ）
  - CPUのみによるMPI通信、cudaMemcpyとCPU間のMPIを組み合わせたもの、CUDA-Aware MPI、NCCLによる通信性能の比較結果（画像）。
  - ディレクトリ名は次ページに対応。
- OSU-MicroBenchmarksによる非同期通信の性能（ディレクトリ）
  - OSU-MicroBenchmarksによる非同期通信の性能比較結果（画像）。CPU同士の通信性能と、GPU同士の通信性能を比べたもの。
  - ディレクトリ名は次ページに対応。

- 各ディレクトリに格納されている性能グラフは7月版のものです。9月版では7月版と比べて推奨されるmodule構成が多少変わりましたが（一部組み合わせで使えるncclバージョンなどが少し違う）、性能を比較した結果、有意な影響はなさそうであったためグラフを更新していません。

利用可能な構成（組み合わせ）一覧

赤字がグラフ画像のディレクトリ名、紫字が利用するmoduleに対応しています。

- CUDA 11.2向け
  - cuda11.2a: module load cuda/11.2.1 openmpi_cuda/4.0.4 nccl/2.8.4
  - cuda11.2h: module load cuda/11.2.1 openmpi_cuda/4.0.4_hpcx2.7.0 nccl/2.8.4
  - cuda11.2x: module load cuda/11.2.1 openmpi_cuda/4.0.5 nccl/2.8.4
- CUDA 11.0向け
  - cuda11.0a: module load cuda/11.0.2 openmpi_cuda/4.0.4 nccl/2.7.8
  - cuda11.0h: module load cuda/11.0.2 openmpi_cuda/4.0.4_hpcx2.7.0 nccl/2.7.8
- CUDA 10.2向け
  - cuca10.2a: module load cuda/10.2.89_440.33.01 openmpi_cuda/4.0.4 nccl/2.7.3
- CUDA 10.1向け
  - cuda10.1a: module load cuda/10.1.243 openmpi_cuda/4.0.4 nccl/2.7.8
  - cuda10.1h: module load cuda/10.1.243 openmpi_cuda/4.0.4_hpcx2.6.0 nccl/2.7.8
- 任意のgcc moduleと組み合わせ可能です。コンパイル対象プログラムにあわせて適切なgccのmoduleを選んでください。（デフォルトではgcc 4.8.5が利用されます。）
  - 例：module load cuda/11.2.1 openmpi_cuda/4.0.4 nccl/2.8.4 gcc/8.4.0

※cuda11.2xとcuda11.2aに目立った性能差はなさそうだが、cuda/11.2.1環境下ではopenmpi_cuda/4.0.5がopenmpi_cudaのdefaultになっている点に注意。（次回定期保守時にopenmpi_cuda/4.0.4をdefaultに変更予定。）

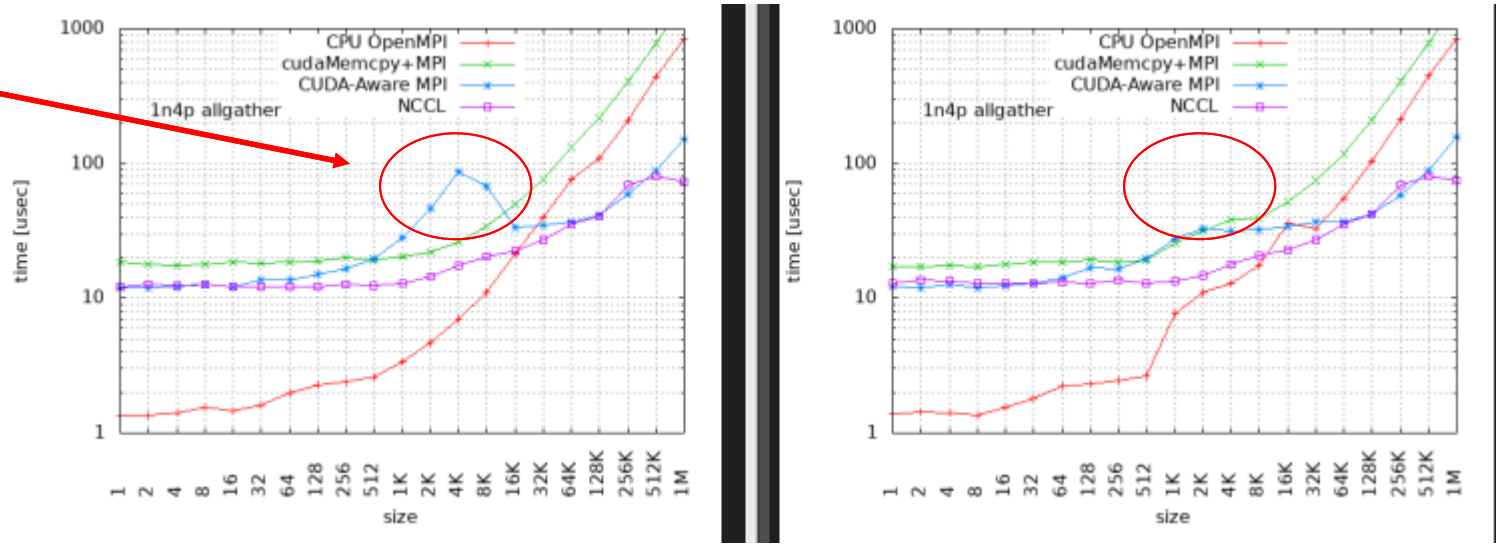
※cuda10.2にHPCX版はありません

利用可能な構成（組み合わせ）一覧：参考

- CUDA 11.2向け 追加テスト
  - cuda11.2a2: module load cuda/11.2.1 openmpi_cuda/4.0.4 nccl/2.8.4
  - cuda11.2h2: module load cuda/11.2.1 openmpi_cuda/4.0.4_hpcx2.7.0 nccl/2.8.4
  - 前ページと同じmodule構成だが、環境変数UCX_RNDV_THRESH=2048を追加した場合の性能。UCX_RNDV_THRESHはMPIの通信方式を切り替える通信サイズを切り替える環境変数。幾つかの通信関数で通信サイズが大きめの場合に性能が低下するのを抑えることができる。
    - =で指定したサイズ以上の通信を行う場合に異なる通信方式を使う、という意味。

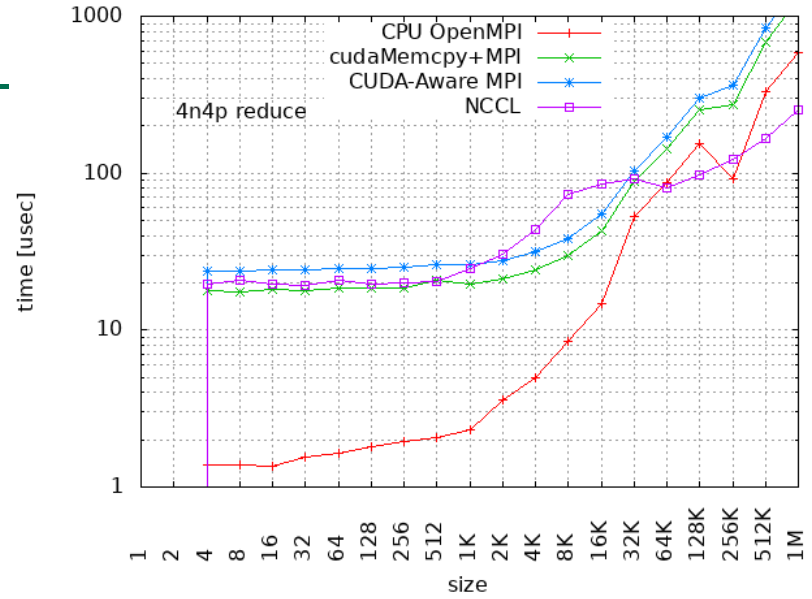
- 値の指定が有効な例
  - あまり多くの例で有効性が示せていないが、今回実験していない通信関数で有効な可能性もある

UCX_RNDV_THRESH=2048を指定した場合

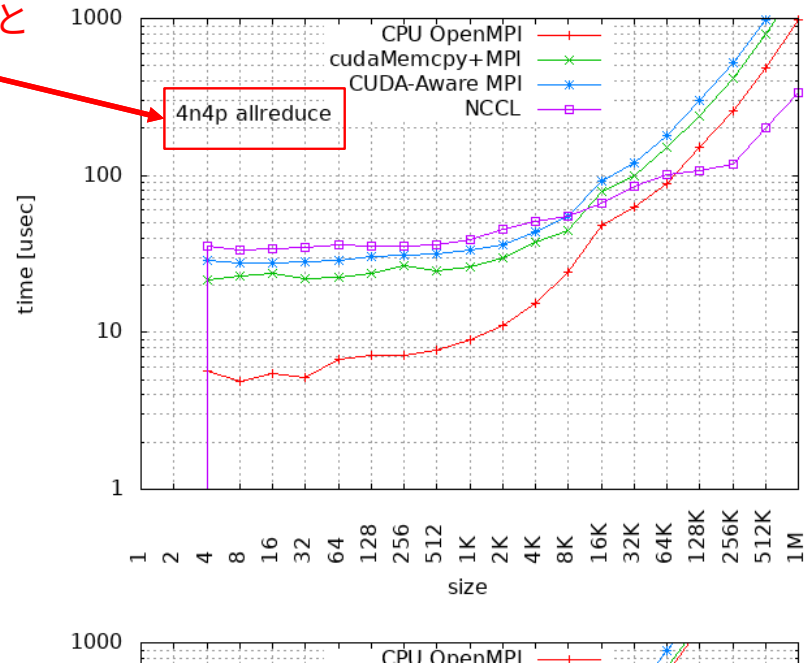


結果グラフの見方

- ディレクトリ名がmoduleの組み合わせに対応
- ファイル名はノード数とプロセス数に対応
  - result_4n4p_all.png
    = 4ノード、ノードあたり4プロセス（4GPU）
- グラフの横軸が示しているのは1通信あたりのByte数
- グラフの縦軸が示しているのは平均実行時間
  - いくつかのグラフは上下の振れ幅が大きい
    - 運悪く外乱があった？
    - ばらつきが大きい実装になっている？
  - ばらつきの度合い・頻度については未調査



ノード数・プロセス数と
通信関数



チェックするべきポイント

- テストプログラムによる基本的な通信性能
  - 実行したいGPU数・ノード数と通信関数ではcudaMemcpy+MPI、CUDA-Aware MPI、NCCLのどれが最速か
    - 常にCUDA-Aware MPIがcudaMemcpy+MPIより高速、とか、常にNCCLが最速、とかであれば良かったのだが、実際に測定してみるとそう単純ではなかった
  - HPCX版と非HPCX版のどちらが高速か
    - 通信方法によって性能の優劣が異なる
    - 全体的にみるとHPCX版はノードあたりGPU数が少ないと高速な傾向？

- OSU-MicroBenchmarksによる非同期通信の性能（ディレクトリ）
  - 通信関数の種類が多いため、使いたいものにあわせて確認してみてください

OpenACCとMPIの組み合わせについて

- 判明していること
  - hpc_sdk/21.2をloadすればHPC SDKに含まれるOpenMPIで問題なく通信できる
    - host_data use_deviceを使うことでupdateなしのMPI通信が可能
  - 具体的な性能はacc2ディレクトリを参照
    - CUDAを用いた場合と比べると若干遅いものが多い気がするが、それほど大きく負けてはいない

- 判明していないこと
  - module版のOpenMPIと組み合わせる余地があるか（性能が向上する可能性があるか）

- その他
  - HPC SDK 21.5もインストールされており、特に制限なく利用することができる
    - コンパイラによる最適化の強化により性能が向上することもあるかもしれない（コードによる）
  - HPC SDK 21.以降ではHPC-Xが同梱されるようになったが、OFEDのバージョンの都合で現在の「不老」では利用不可能

利用可能な構成（組み合わせ）一覧

「7月版」で案内していたもの

赤字がグラフ画像のディレクトリ名、紫字が利用するmoduleに対応しています。

- CUDA 11.2向け
  - cuda11.2a: module load cuda/11.2.1 openmpi_cuda/4.0.4 nccl/2.7.8
  - cuda11.2h: module load cuda/11.2.1 openmpi_cuda/4.0.4_hpcx2.7.0 nccl/2.7.8
  - cuda11.2x: module load cuda/11.2.1 openmpi_cuda/4.0.5 nccl/2.8.4
- CUDA 11.0向け
  - cuda11.0a: module load cuda/11.0.2 openmpi_cuda/4.0.4 nccl/2.7.8
  - cuda11.0h: module load cuda/11.0.2 openmpi_cuda/4.0.4_hpcx2.7.0 nccl/2.7.8
- CUDA 10.2向け
  - cuca10.2a: module load cuda/10.2.89_440.33.01 openmpi_cuda/4.0.4 nccl/2.7.8
- CUDA 10.1向け
  - cuda10.1a: module load cuda/10.1.243 openmpi_cuda/4.0.4 nccl/2.7.8
  - cuda10.1h: module load cuda/10.1.243 openmpi_cuda/4.0.4_hpcx2.6.0 nccl/2.7.8
- 任意のgcc moduleと組み合わせ可能です。コンパイル対象プログラムにあわせて適切なgccのmoduleを選んでください。（デフォルトではgcc 4.8.5が利用されます。）
  - 例：module load cuda/11.2.1 openmpi_cuda/4.0.4 nccl/2.7.8 gcc/7.4.0

※cuda11.2xは従来から提供していたもの。使い続けても特に問題はない。cuda11.2aと目立った性能差はなさそう。

※cuda10.2にHPCX版はありません

利用可能な構成（組み合わせ）一覧：参考

「7月版」で案内していたもの

- CUDA 11.2向け 追加テスト
  - cuda11.2a2: module load cuda/11.2.1 openmpi_cuda/4.0.4 nccl/2.7.8
  - cuda11.2h2: module load cuda/11.2.1 openmpi_cuda/4.0.4_hpcx2.7.0 nccl/2.7.8
  - 前ページと同じmodule構成だが、環境変数UCX_RNDV_THRESH=2048を追加した場合の性能。UCX_RNDV_THRESHはMPIの通信方式を切り替える通信サイズを切り替える環境変数。幾つかの通信関数で通信サイズが大きめの場合に性能が低下するのを抑えることができる。
    - =で指定したサイズ以上の通信を行う場合に異なる通信方式を使う、という意味。

- 値の指定が有効な例
  - あまり多くの例で有効性が示せていないが、今回実験していない通信関数で有効な可能性もある

UCX_RNDV_THRESH=2048を指定した場合