2023.05.09　第3回気象学会計算科学研究連絡会

気象庁全球モデルにおける スペクトル変換過程のGPU対応について

気象庁 情報基盤部 数値予報課 数値予報開発センター
林田和大

はじめに

- 気象庁が平成30年に策定した「2030年に向けた数値予報技術開発重点計画」では、台風防災への貢献を重点目標の一つとして掲げ、以下を開発の方向性として示している
  - 気象庁全球モデル（以下、GSM）の**水平格子間隔を10km以下へ高解像度化**
  - 高解像度モデルに適した物理過程の開発
  - データ同化の高度化

- 仕様増強には計算資源の大幅な増加が伴う一方、計算機の性能向上のペースが鈍化している
  - ムーアの法則の限界、ノード間通信性能やメモリバンド幅の頭打ち、etc

- 重点計画の達成に向けて、GPUのようなアクセラレータとCPUを組み合わせたヘテロジニアスな構成を持つ将来の計算機上でモデルを高速に実行することを視野に入れる必要がある

はじめに

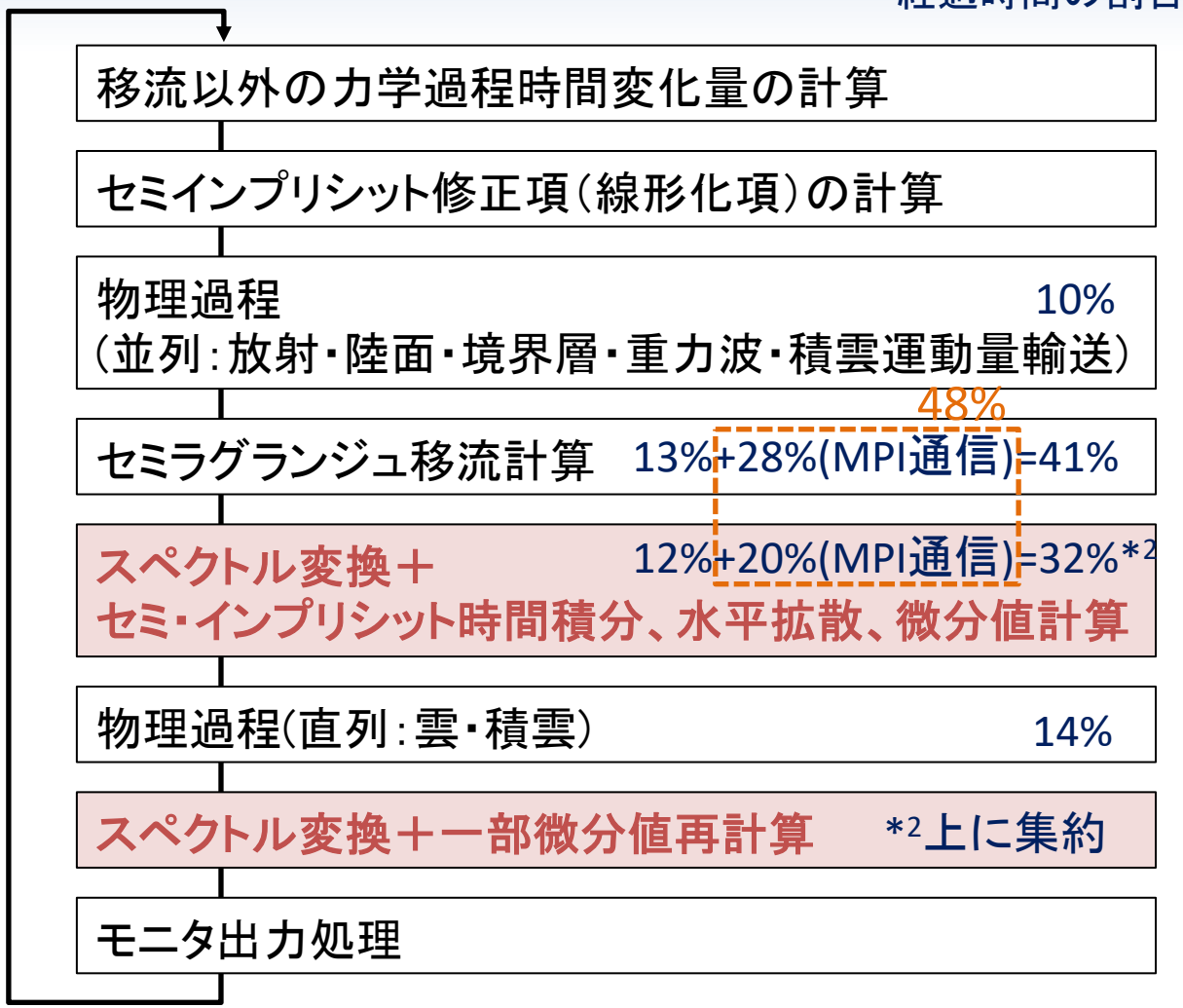
- 気象庁では、GPUを用いた現業数値予報モデルの高速化について基礎調査を実施している
  - モデルでのGPU利用に関する知見・情報を収集したい
  - ヘテロジニアスな構成（GPU等のアクセラレータ+CPU）を持つ計算機がトレンド、将来の計算機への備え

- 本講演ではGSMにおけるスペクトル変換過程のGPU対応で得られた結果及び知見について紹介する

- GSMではGPU利用以外にも、将来の計算機への対応として以下を検討中
  - MPI分割手法の改良による通信量の削減
  - 低精度の浮動小数点演算の活用（単精度化）

GSMについて

GSMにおける時間積分ループ　時間積分ループ内の経過時間の割合*1

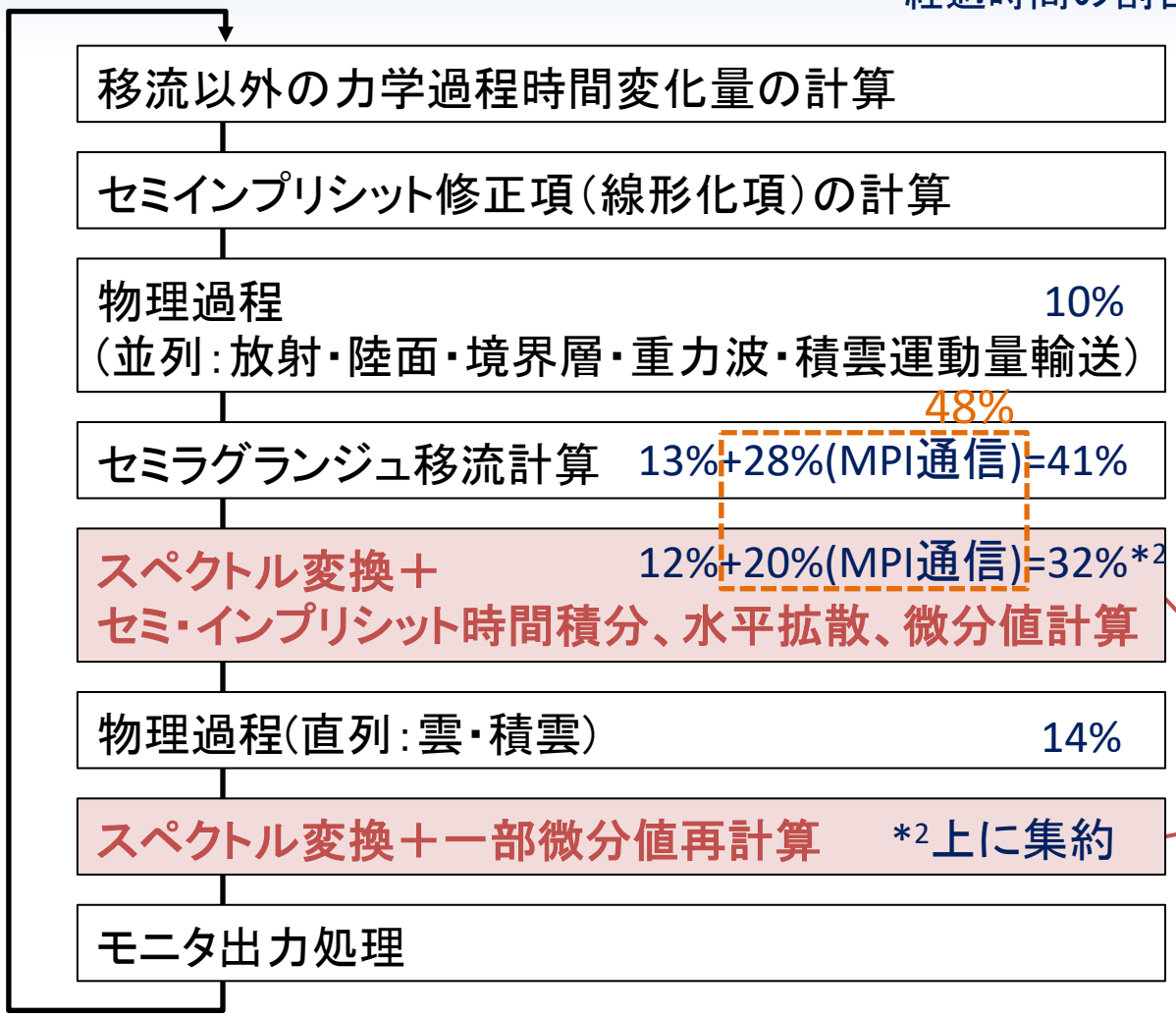


- GSMはセミインプリシット・セミラグランジュ法のスペクトルモデル
- 時間積分ループ内の経過時間のうち、MPI通信が約半分を占める
  - GSMはスペクトル変換とセミラグランジュ法による移流計算において全対全通信が発生する
- スペクトル変換では高速化のために数値計算ライブラリを使っている箇所がある

*1高解像度決定論予報（GSM2303）のある初期値のランク0における経過時間の割合。10%以上を占める過程のみ記載。

GSMについて

GSMにおける時間積分ループ　時間積分ループ内の経過時間の割合*1



- 最終的には時間積分ループ内をすべてGPU化したい
  - ループ前に初期値をCPUからGPUに転送し、GPU内で閉じて計算する
    - 一般的にCPU-GPU間のデータ転送が律速、高速化する上で極力データ転送を減らすことが重要
  - ただし、モニタ出力処理はCPUで行う
- 本発表ではスペクトル変換過程のGPU化を行った結果について紹介する

スペクトル変換過程

【課題】

- MPI通信のGPU化
- 数値計算ライブラリにGPU最適化ライブラリ(NVIDIA数学ライブラリ)の適用

*1高解像度決定論予報（GSM2303）のある初期値のランク0における経過時間の割合。10%以上を占める過程のみ記載。

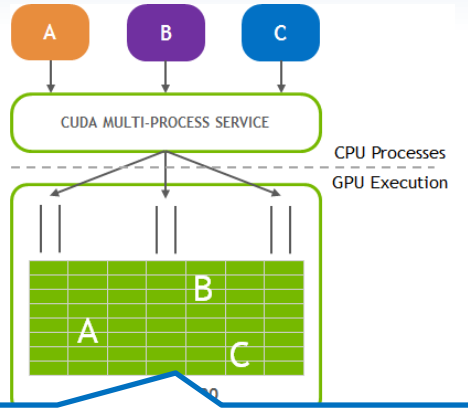
実行環境

本調査で用いた計算ノード	
CPU	Intel Xeon Gold 6226 2.7GHz 12C/24T x2
GPU	NVIDIA Tesla V100-SXM2-32GB 2560FP64コア x4
理論演算性能	倍精度演算33.272TFLOPS （CPU：1.036TFLOPS x2、GPU：7.8TFLOPS x4） （CPU x2：GPU x4=約1:16）
メモリバンド幅	メインメモリ：140.4GB/s x2 デバイスメモリ：900GB/s x4
GPU間接続	NVLink2.0　1GPUから他3GPUへ対してそれぞれ100GB/s、計300GB/s x4（双方向）
CPU-GPU間接続	PCI-Express 3.0 x16　32GB/s（双方向）

・コンパイラはnvfortranを利用
・OpenMPIを利用
・1ノードで実行する
・MPS(Multi-Process Service)を用いて実行する
　—1つのGPU上で複数プロセスを効率的に非同期実行する機能

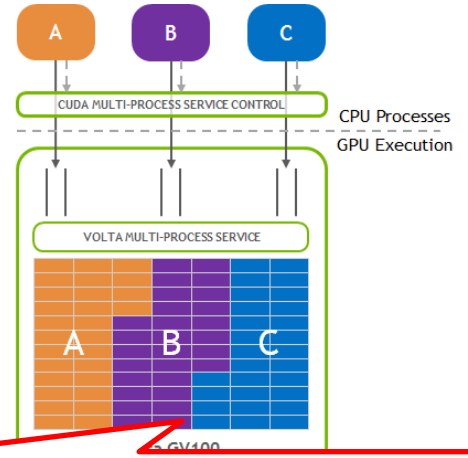
NVIDIA「MULTI-PROCESS SERVICE」
1.1.2. Volta MPS

MPSなし



各プロセスはtime-sliceで実行

MPSあり



メモリ領域を独立し、処理を並列に実行

スペクトル変換過程のGPU化について

- できる限りコード構造を変えずにOpenACCのディレクティブの挿入のみでGPU化することを基本方針とする

- GSMのスペクトル変換単体をGPU化して効果測定したのちに、モデル本体にも実装する

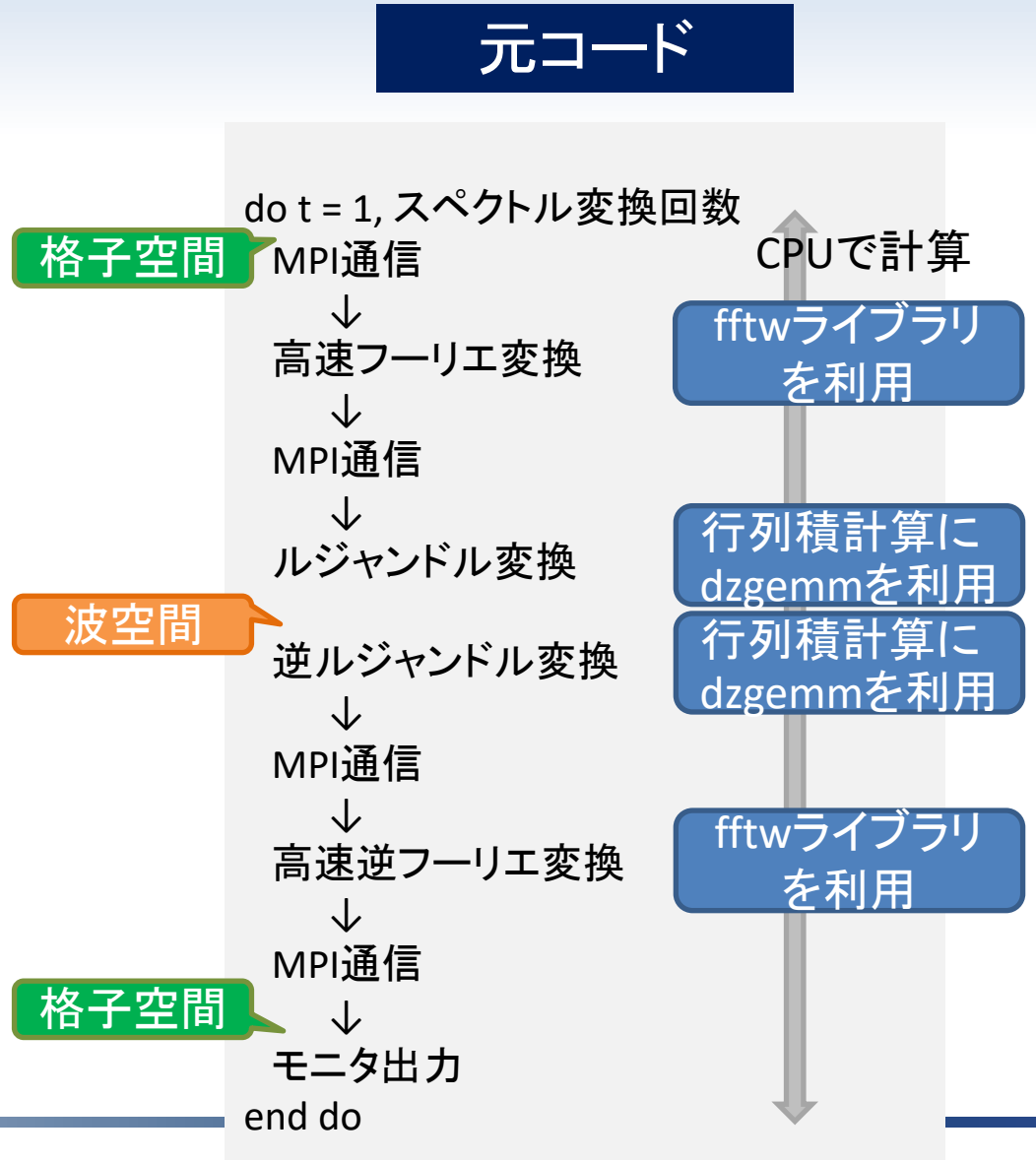
- **NVIDIA数学ライブラリcuFFT、cuBLASを利用**する
  - GSMのスペクトル変換過程では、高速化のために高速フーリエ変換及びルジャンドル変換の行列積計算に数値計算ライブラリを用いている
  - これらのライブラリをGPU上で計算するように最適化されたNVIDIA数学ライブラリに置き換える
  - ライブラリ以外の箇所はOpenACCを用いてGPU化する

- **MPI通信はCUDA Aware-MPIを利用**してGPU化する

スペクトル変換単体の概要

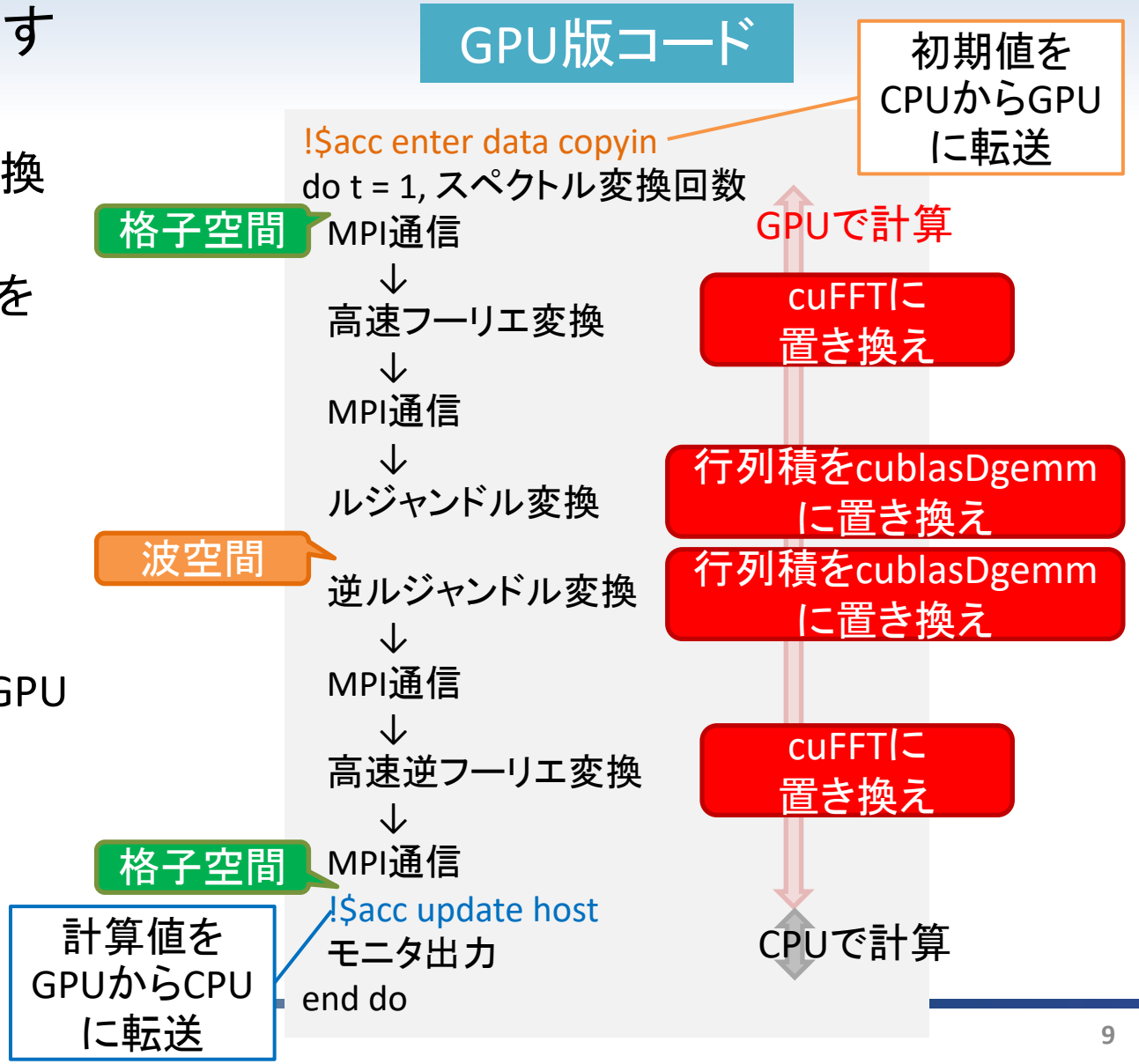
- GSMモデルフレームを用いてスペクトル変換単体のGPU化を確認
  - 波・格子変換を繰り返し実行するプログラム
- 高速フーリエ変換ではfftwライブラリを、ルジャンドル変換の行列積計算にはdzgemmを利用している*

  * ライブラリはIntelMKLを利用
- スペクトル変換に伴い、全対全のMPI通信が発生する
  - フーリエ（逆フーリエ）変換において東西格子（東西波数）の、ルジャンドル（逆ルジャンドル）変換において南北格子（全波数）の並び変えが必要となるため



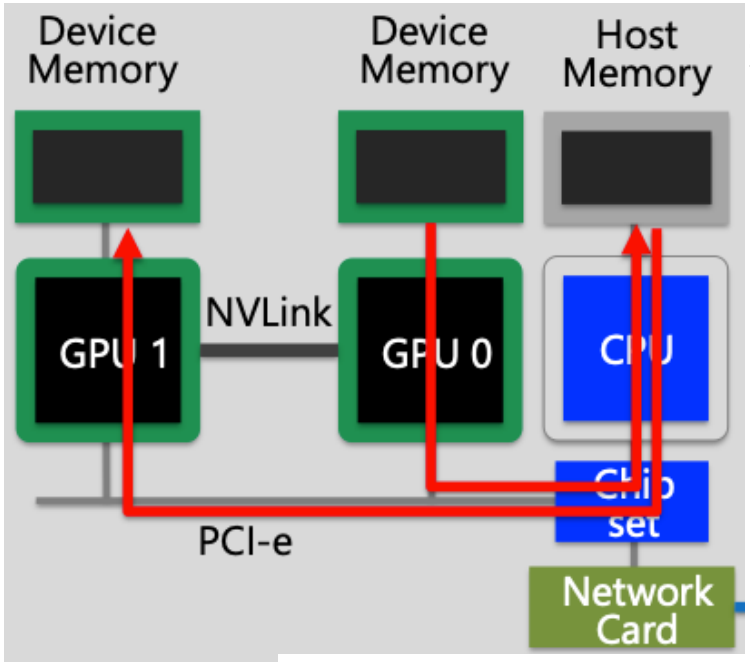
スペクトル変換単体のGPU化

- ループ内はモニタ出力以外、GPU上で計算するよう実装
  - 高速フーリエ変換ではcuFFTを、ルジャンドル変換の行列積計算にはcuBLASを利用する
  - ライブラリ以外の箇所はOpenACCディレクティブを挿入する
  - コード構造は変えない
- CPU-GPU間のデータ転送を最低限にする
  - ループ前に初期値をCPUからGPUへ転送
  - GPU上で更新した計算結果を、モニタ出力前でGPUからCPUに転送
- MPI通信はCUDA Aware-MPIを用いてGPUで実行する（次スライド）



ノード内におけるGPU間のMPI通信

- 従来のMPIではMPI関数の受信領域、送信領域にホストメモリ(CPU側のメモリ)上のアドレスのみ指定可能
  - デバイスメモリ上のデータを使ってMPI通信するには、一度ホストメモリにデータを転送する必要がある
- CUDA aware-MPIはデバイスメモリ(GPU側のメモリ)上のアドレスを指定することが可能になる機能
  - ホストメモリを経由せずに、PCI-eよりも高速なNVLinkを通って、MPI通信が可能



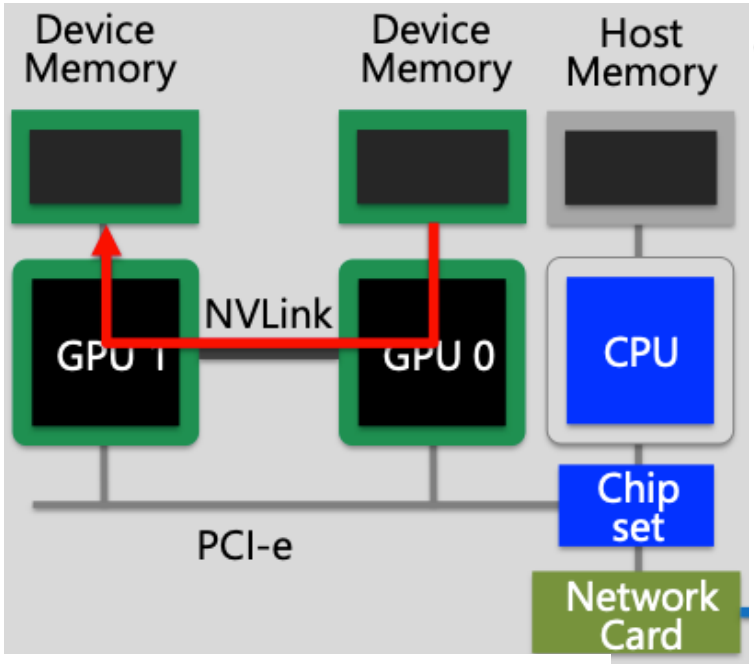
デバイスメモリとホストメモリの間のPCI-eを通る

本調査の計算環境では約3倍帯域幅が違う

PCIe 3.0 x16 ~ 32GB/s
(双方向)

∧

NVLink 2.0 ~ 100 GB/s
(1GPU→他3GPU、双方向)



NVLink経由でGPU間通信が可能

2GPU 搭載の1ノードにおいて、GPU0でsend、GPU1でrecvするMPI通信の模式図
(東京大学 情報基盤センター 第167回 お試しアカウント付き 並列プログラミング講習会)

スペクトル変換単体　実験設定

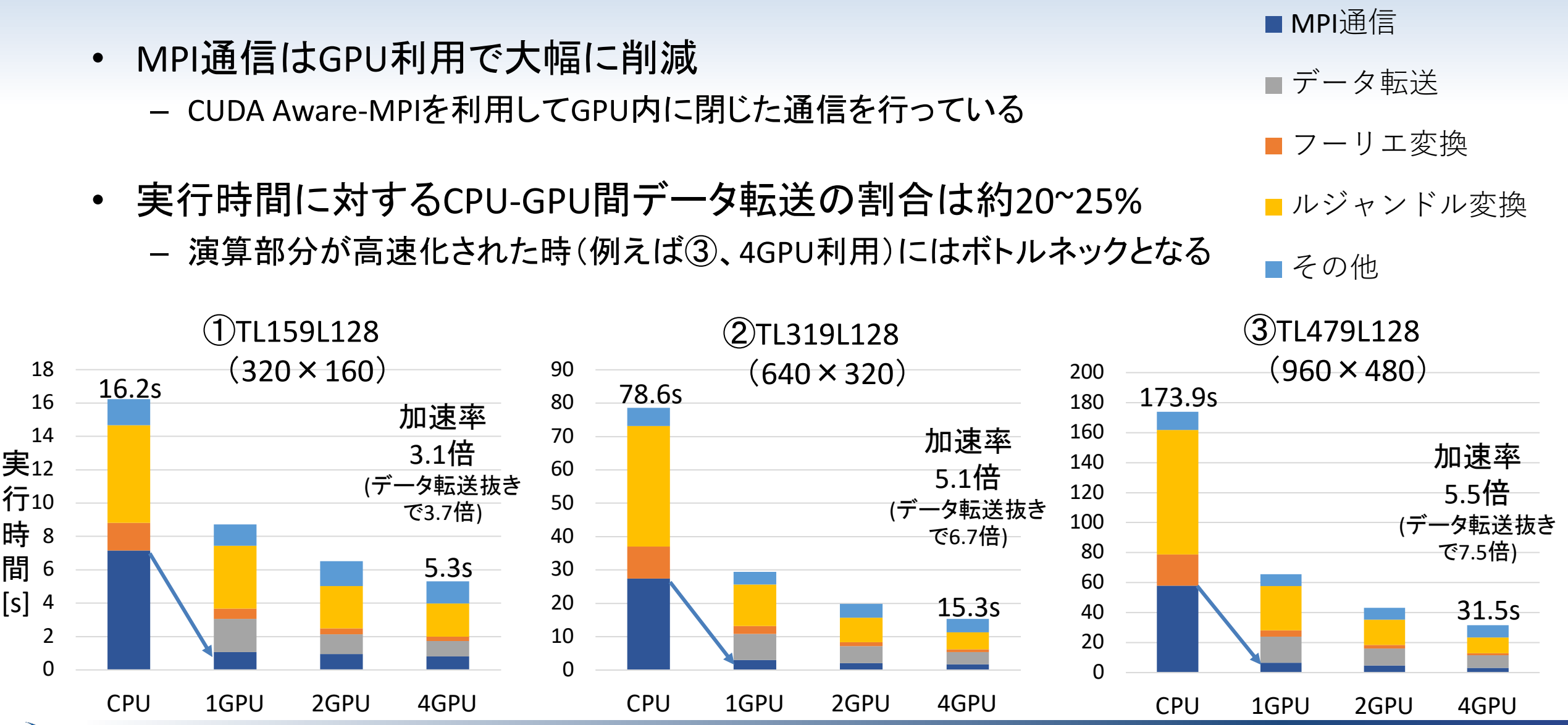
- モデル解像度（問題サイズ）を変えて、CPUのみ/1GPU/2GPU/4GPUで実行し比較する
  - **加速率の問題サイズ依存性**と**GPU数増加による加速率の伸び**を確認する
  - モデル解像度は3種類

設定	モデル解像度	水平格子間隔	問題サイズ＝水平格子点数 (東西×南北)
①	TL159L128	110km	320×160
②	TL319L128	55km	640×320（①の4倍）
③	TL479L128	40km	960×480（①の9倍、②の2.25倍）

- GPU実行と比較するCPU実行のライブラリには、Intel MKLを利用する
  - 高速フーリエ変換ではfftwライブラリを利用
  - ルジャンドル変換の行列積ではdzgemmを利用

スペクトル変換単体　実行時間

- MPI通信はGPU利用で大幅に削減
  - CUDA Aware-MPIを利用してGPU内に閉じた通信を行っている
- 実行時間に対するCPU-GPU間データ転送の割合は約20~25%
  - 演算部分が高速化された時（例えば③、4GPU利用）にはボトルネックとなる

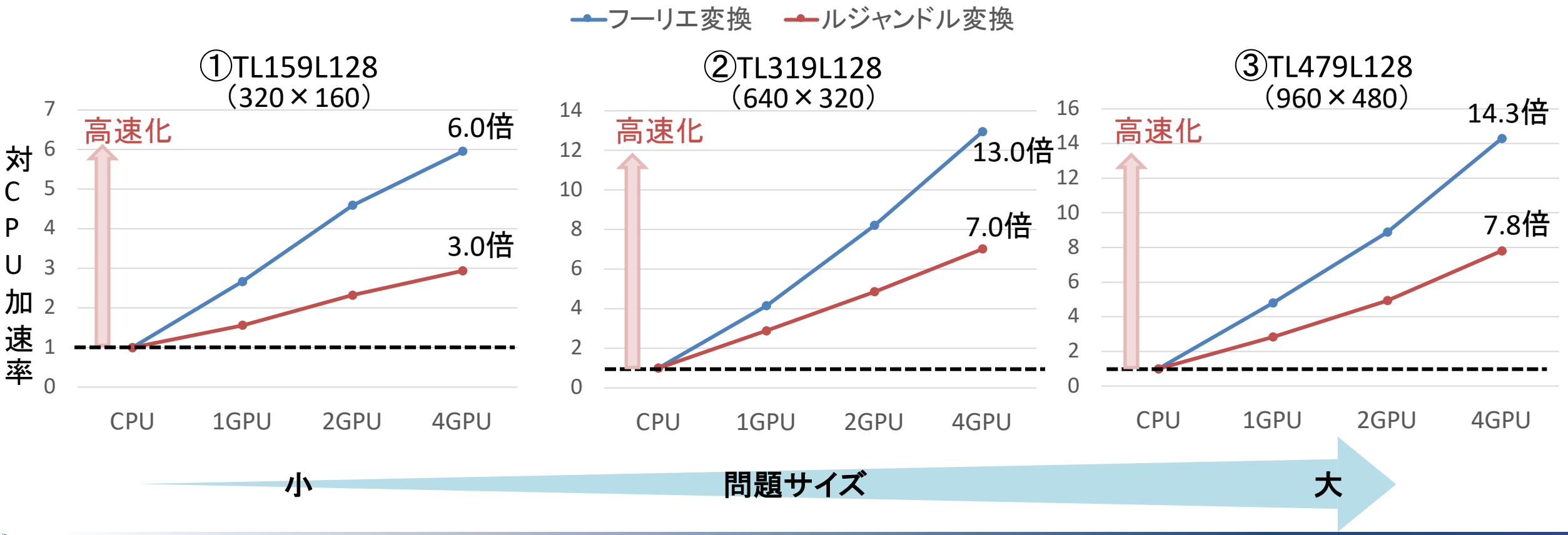


加速率 = CPU(24core)実行時間 / GPU実行時間

スペクトル変換単体　演算部分の対CPU加速率

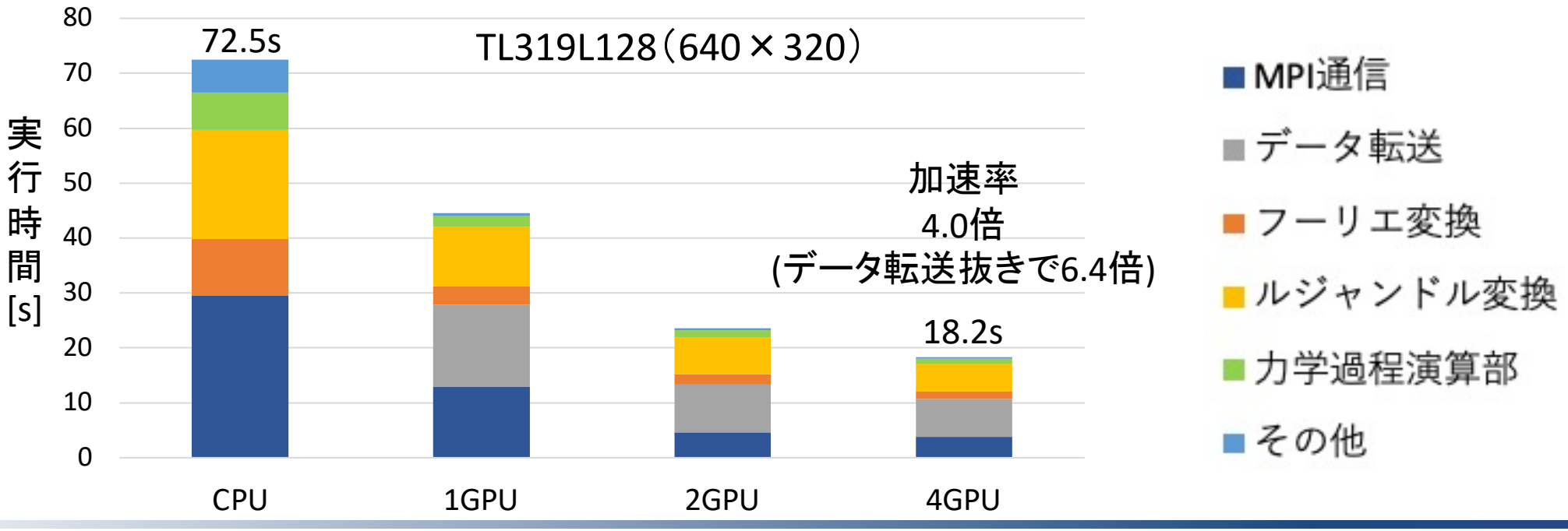
- NVIDIA数学ライブラリを利用したフーリエ変換、ルジャンドル変換で高速化を確認
  - 特に問題サイズが大きいほど、複数GPU利用した場合の加速率*は大きくなる

*加速率 = CPU(24core)実行時間 / GPU実行時間



GSMのスペクトル変換過程のGPU化

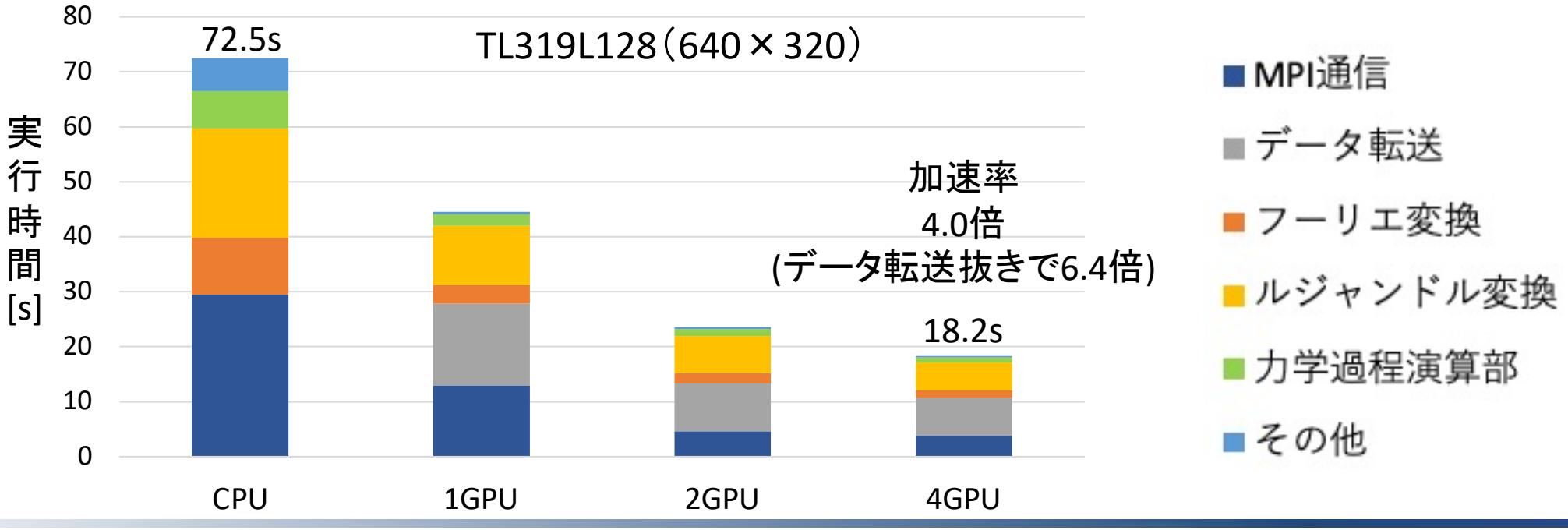
- GSMモデル本体にも実装し、効果を測定した
  - 低解像度決定論予報（TL319L128、東西640格子×南北320格子）で実行、利用するGPU数を変えて12h予報を実行
  - 下図はスペクトル変換過程のみの計測結果



加速率 = CPU(24core)実行時間 / GPU実行時間

GSMのスペクトル変換過程のGPU化

- スペクトル変換単体のGPU化で得られた結果と整合する結果が得られた
  - MPI通信時間の削減、演算部分の高速化、データ転送時間がボトルネック
- CPU実行に対する加速率は4GPU実行で4.0倍（データ転送抜きで6.4倍）
  - 理論演算性能比はCPU:4GPU=約1:16



加速率 = CPU(24core)実行時間 / GPU実行時間

まとめ

- 気象庁では、GPUを用いた現業数値予報モデルの高速化について基礎調査を実施中

- **GSMのスペクトル変換過程をOpenACCでGPU化**し、性能測定を行った
  - MPI通信はCUDA Aware-MPIを利用しGPU化
  - 数値計算ライブラリをNVIDIA数学ライブラリのcuFFT、cuBLASに置き換え
  - ライブラリ以外の箇所はOpenACCディレクティブを挿入

- スペクトル変換単体とモデル本体の両方で整合する結果が得られた
  - MPI通信時間の削減と演算部分の高速化を確認
  - データ転送時間がボトルネック

- モデル本体のスペクトル変換過程でのCPU実行に対する**加速率は4GPU実行で4.0倍**
  - 低解像度決定論予報（TL319L128、東西640格子×南北320格子）での結果
  - **データ転送抜きで6.4倍**
  - 理論演算性能比は**CPU:4GPU=約1:16**

今後の課題

- ノード間におけるGPU間直接通信の性能調査
  - 今回の調査では**ノード内MPI通信をGPU化**したことにより大きく実行時間を削減することができた
    - CPU-GPU間はPCI-Expressで接続、GPU間はNVLINKで接続
    - CPUのメモリバンド幅よりもGPUのメモリバンド幅が6倍強大きい
  - 複数ノードで実行する場合、**ノード間GPU間直接通信（GPUDirectRDMA）**を用いる
    - GPUDirectRDMAで実行する場合、（構成にもよるが）PCI-Expressを通るため今回得られた結果ほどの高速化は得られないのではないか？
    - 実際に測定し、ノード内・ノード間の構成についても議論する必要がある
- GPUによる高速化の測定方法の検討
  - 現在はCPU実行時間に対する加速率を算出して、GPUによる高速化を測定している
    - 加速率がCPUの演算性能に左右されるため、解釈が難しい
    - モデルのGPU化を優先しており、現段階では簡易的な測定しかできていない
  - 多角的な測定方法を検討する必要がある