中核機関報告2

東京大学におけるGPU移行の取り組みと先進GPUプログラム開発

下川辺 隆史

東京大学

第1回 HAIRDESCシンポジウム

（計算・データ・学習）融合：（S+D+L）

スパコンのワークロードの多様化：S・D・L

- 計算科学，計算工学：Simulations（S）
- 大規模データ解析：Data（D）
- AI，機械学習：Learning（L）

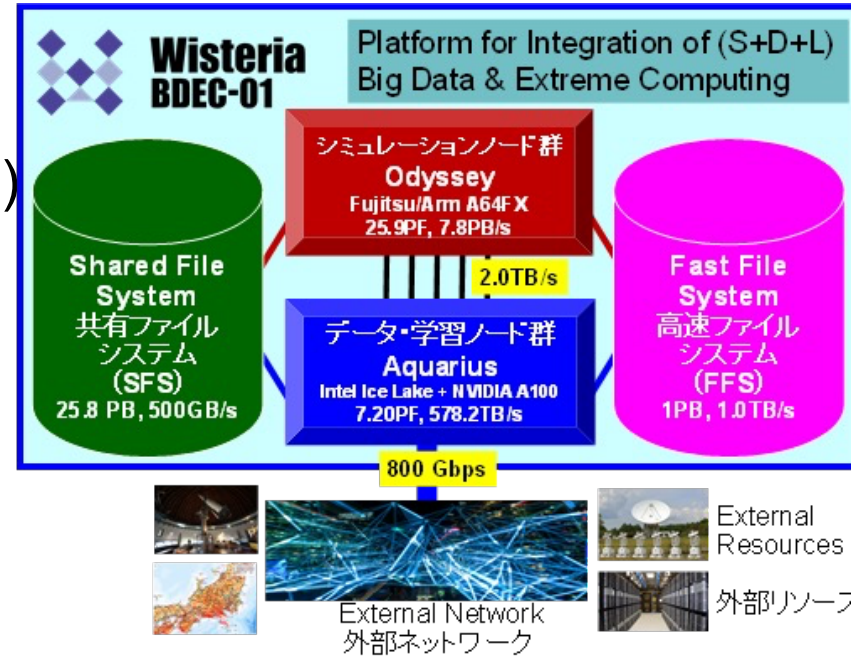
東大情報基盤センターでは，2015年頃から「（S+D+L）融合」による新しい科学の開拓を実現する，ハードウェア，ソフトウェア，アプリケーション，アルゴリズムに関する研究開発を開始

- BDEC計画（Big Data & Extreme Computing）
- 「データ＋学習」による，より高度な「シミュレーション」⇒AI for HPC
- 地球科学関連では自然な発想（すでに実施されている）
- 2021年5月に運用を開始した「Wisteria/BDEC-01」は「BDEC計画」の1号機
  - 「計算・データ・学習（S+D+L）」融合を実現する，世界でも初めてのプラットフォーム

Wisteria/BDEC-01

BDEC:「計算・データ・学習（S+D+L）」融合のためのプラットフォーム（Big Data & Extreme Computing）

- 2021年5月~2027年5月運用終了予定
  - 東京大学柏Ⅱキャンパス
- 33.1 PF, 8.38 PB/sec.，**富士通製**
  - ~4.5 MVA（空調込み），~360m^2
- Hierarchical, Hybrid, Heterogeneous (h3)
- **2種類のノード群**
  - **シミュレーションノード群（S, SIM）：Odyssey**
    - **従来のスパコン**
    - **Fujitsu PRIMEHPC FX1000 (A64FX), 25.9 PF**
      - **7,680ノード（368,640 コア），20ラック，Tofu-D**
  - **データ・学習ノード群（D/L, DL）：Aquarius**
    - **データ解析，機械学習**
    - **Intel Xeon Ice Lake + NVIDIA A100, 7.2 PF**
      - **45ノード（Ice Lake：90基，A100：360基），IB-HDR**
    - **外部リソース（ストレージ，サーバー，センサーネットワーク他）に直接接続**
  - ファイルシステム：共有（大容量）＋高速

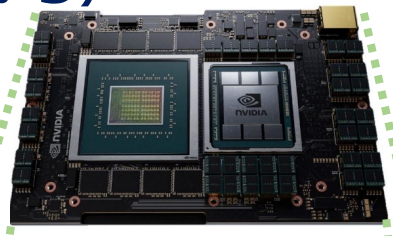


Wisteria
BDEC-01

Miyabi

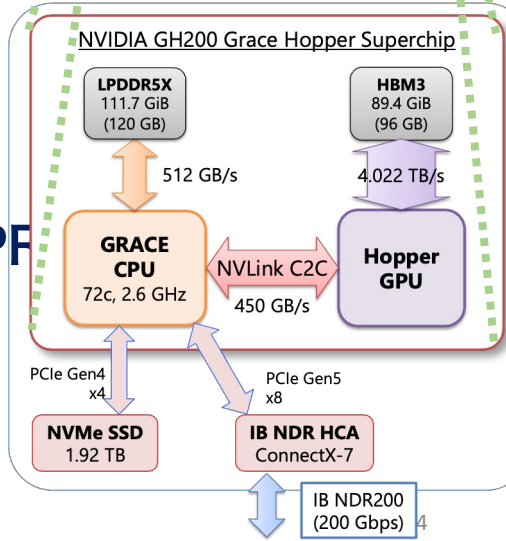
JCAHPC
筑波大学 University of Tsukuba
東京大学 THE UNIVERSITY OF TOKYO

筑波大学と東京大学が共同で運営する 最先端共同HPC基盤施設（JCAHPC）が運用（2025年1月から）



Miyabi-G: 演算加速ノード: NVIDIA GH200

- 計算ノード: NVIDIA GH200 Grace-Hopper Superchip
  - Grace: 72c, 3.45 TF, 120 GB, 512 GB/sec (LPDDR5X)
  - H100: 66.9 TF DP-Tensor Core, 96 GB, 4,022 GB/sec (HBM3)
    - CPU-GPU間はキャッシュコヒーレント
  - NVMe SSD for each GPU: 1.9TB, 8.0GB/sec, GPUDirect Storage
- **合計 (CPU+GPUの合計値)**
  - 1,120 ノード, 78.8 PF, 5.07 PB/sec, IB-NDR 200



Miyabi-C: 汎用CPUノード: Intel Xeon Max 9480 (SPR

- 計算ノード: Intel Xeon Max 9480 (1.9 GHz, 56c) x 2
  - 6.8 TF, 128 GiB, 3,200 GB/sec (HBM2e only)
- 合計
  - 190 ノード, 1.3 PF, IB-NDR 200
  - 372 TB/sec for STREAM Triad (Peak: 608 TB/sec)

Miyabi活用に向けてのGPU移行支援

国内でもGPU搭載スパコンが増えている。2030年頃運用開始予定の「富岳NEXT」もNVIDIA GPU搭載。

Miyabiは東大初のGPU主体のスパコン

- ユーザのGPU移行を支援するため、GPU関連の講習会、GPU移行相談会、ポータルサイトなど整備
  - どなたでも参加可能、無料

GPU移行ポータルサイト

Miyabiに搭載されたGH200は、GPU移植に最適

- CPU、GPUどちらからも相互にメモリ空間を参照可能。
- CPU-GPUのデータ転送がボトルネックとなりにくい。
- 従来はGPU化というと、「すべてのデータをGPUにおく」が鉄則であったが、「遅い部分をまずGPUに移植し試してみる」という方針が有効。

GPU移植・移行の計画

NVIDIA Japanの協力

3,000人以上のOFP利用者：2つの形態

「自己移植（Self Porting）」：様々なオプション

- 「移行ポータルサイト」https://jcahpc.github.io/gpu_porting/
- 1週間のハッカソン（ミニキャンプ），3ヶ月に1回，オンライン・ハイブリッド，Slack併用
- 毎月開催される「相談会」（Zoom，非ユーザーも自由に参加できる）
  - GPU化の相談、プロファイラの相談、ライブラリ相談など
- 各種講習会

「サポート移植（Surpported Porting）」，2022年10月開始

- 多くのユーザーを有するコミュニティコード（19種類），OpenFOAM（NVIDIA）
- 外注のための予算も確保（落札ベンダーが担当）
- 「サポート移植」グループメンバー（主に若手）はハッカソン・相談会にも積極的に参加

基本的にOpenACC/StdPar（Standard Parallelism）推奨

Category	Name (Organizations)	Target, Method etc.	Language
Engineering (5)	FrontISTR (U.Tokyo)	Solid Mechanics, FEM	Fortran
	FrontFlow/blue (FFB) (U.Tokyo)	CFD, FEM	Fortran
	FrontFlow/red (AFFr) (Advanced Soft)	CFD, FVM	Fortran
	FFX (U.Tokyo)	CFD, Lattice Boltzmann Method (LBM)	Fortran
	CUBE (Kobe U./RIKEN)	CFD, Hierarchical Cartesian Grid	Fortran
Biophysics (3)	ABINIT-MP (Rikkyo U.)	Drug Discovery etc., FMO	Fortran
	UT-Heart (UT Heart, U.Tokyo)	Heart Simulation, FEM etc.	Fortran, C
	Lynx (Simula, U.Tokyo)	Cardiac Electrophysiology, FVM	C
Physics (3)	MUTSU/iHallMHD3D (NIFS)	Turbulent MHD, FFT	Fortran
	Nucl_TDDFT (Tokyo Tech)	Nuclear Physics, Time Dependent DFT	Fortran
	Athena++ (Tohoku U. etc.)	Astrophysics/MHD, FVM/AMR	C++
Climate/ Weather/ Ocean (4)	SCALE (RIKEN)	Climate/Weather, FVM	Fortran
	NICAM (U.Tokyo, RIKEN, NIES)	Global Climate, FVM	Fortran
	MIROC-GCM (AORI/U.Tokyo)	Atmospheric Science, FFT etc.	Fortran77
	Kinaco (AORI/U.Tokyo)	Ocean Science, FDM	Fortran
Earthquake (4)	OpenSWPC (ERI/U.Tokyo)	Earthquake Wave Propagation, FDM	Fortran
	SPECFEM3D (Kyoto U.)	Earthquake Simulations, Spectral FEM	Fortran
	hbi_hacapk (JAMSTEC, U.Tokyo)	Earthquake Simulations, H-Matrix	Fortran
	sse_3d (NIED)	Earthquake Science, BEM (CUDA Fortran)	Fortran

GPUハッカソン・ミニキャンプ

ハッカソン・ミニキャンプとは？

- 参加者がコードやデータセットを持ち込み、CUDA、OpenACC、Deep Learning など、GPUに関連した課題に対して、メンターからの助言を受けながら、その課題解決に取り組みます。
- ミニキャンプは8日間、ハッカソンは15日間（講習会実施日はそれぞれ2日、3日）
- ハイブリッドまたはオンライン

メンター

- 課題解決に協力
- JCAHPC（筑波大、東大）、北大、東北大、科学大、名大、京大、阪大、九州大、NVIDIA、富士通、プロメテック・ソフトウェアなどから参加

次回GPUミニキャンプ

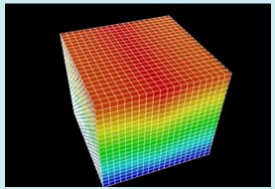
- 年間3回ほど実施予定
- 募集チーム数：約10チーム

HAIRDESCにける
東大での研究開発

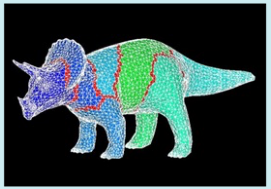
東大：先進的GPUプログラム開発に向けた研究開発

HPCアプリケーション
格子系（構造・非構造），多体系

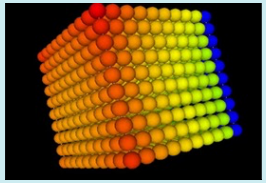
- サンプルコードの作成と性能結果公開
- CPU-GPU同時効率的利用法



構造格子（差分法）



非構造格子
（有限要素/体積法）



多体系（N体問題）

性能可搬性プログラミング環境：
Fortranからの移行

kokkos

- Kokkos（GPU用性能可搬フレームワーク，C/C++専用）への収斂
- Solomon（OpenACC/OpenMP統合ライブラリ）等を併用，段階的にFortranから脱却

OpenACC とOpenMP の両方に対応した通常の実装

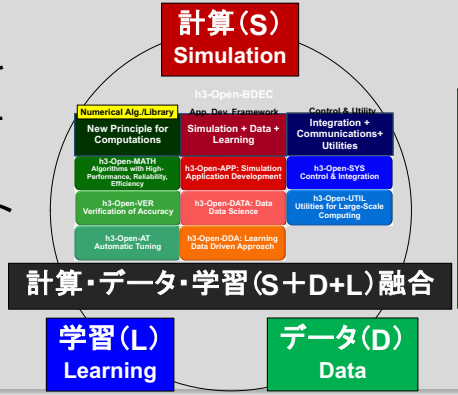
```
#if defined(OFFLOAD_BY_OPENACC)
#if defined(OFFLOAD_BY_OPENACC_KERNELS)
#pragma acc kernels
#pragma acc loop
#else
#pragma acc parallel
#pragma acc loop
#endif
#endif
#if defined(OFFLOAD_BY_OPENMP_TARGET)
#if defined(
    OFFLOAD_BY_OPENMP_TARGET_LOOP)
#pragma omp target teams loop
#else
#pragma omp target teams distribute
    parallel for
#endif
#endif
  for (int i = 0; i < N_i; i++) {
    // loop body A
  }
```

Solomon を用いた等価な実装

```
OFFLOAD()
for (int i = 0; i < N_i; i++) {
  // loop body A
}
```

AI for Science

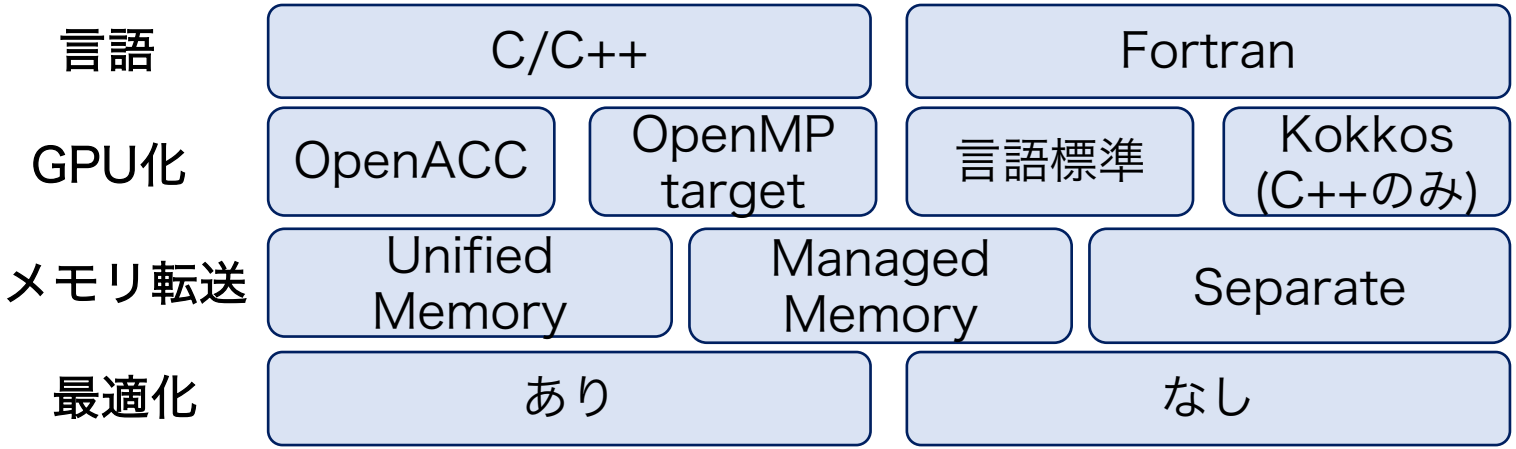
- 「計算・データ・学習（S+D+L）」融合を実現する革新的ソフトウェア基盤「h3-Open-BDEC」
- 量子・HPCハイブリッド連携もサポート



計算科学アプリケーションのGPU化向けコード整備

構造格子系、非構造格子系、多体系のGPUコードの整備

- OpenACC、OpenMP target、言語標準、Kokkos によるGPU化の実装例
- 下記のすべてのバリエーションを作成



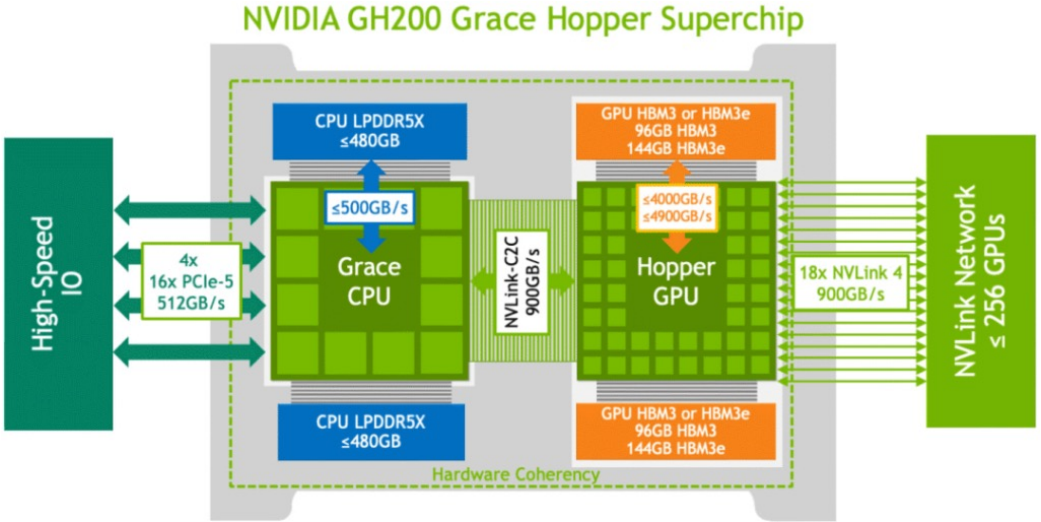
- Wisteria/BDEC-01 Aquarius (NVIDIA A100)、Miyab-G(GH200) で測定
  - 近日、測定結果を公開予定
  - GPU化によって得られる性能の参考値として活用することを想定

GH200の特性の活用

Grace-Hopper相互のメモリ空間を直接参照可，NUMA的な扱い
CPU-GPU間：コヒーレントインタフェース（NVLink-C2C, 450GB/sec/dir）

- PCIe Gen 5の7倍以上の帯域
- CPU-GPUの効率的使い分け可能
- 従来は転送がボトルネック
  - プログラミングも容易
  - AMD MI300Aも同じ方向性

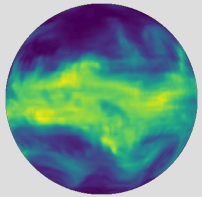
小規模問題，GPUが不得意な計算をCPUが柔軟に処理することも可能



気候コードによるCPU-GPU同時効率的利用

対象アプリ
SP-MIROC

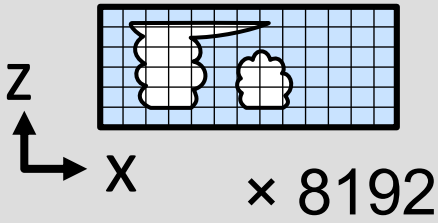
全球気候モデル MIROC6



- 低解像度
- 大規模循環を担当
- 数個のCPUコアで十分

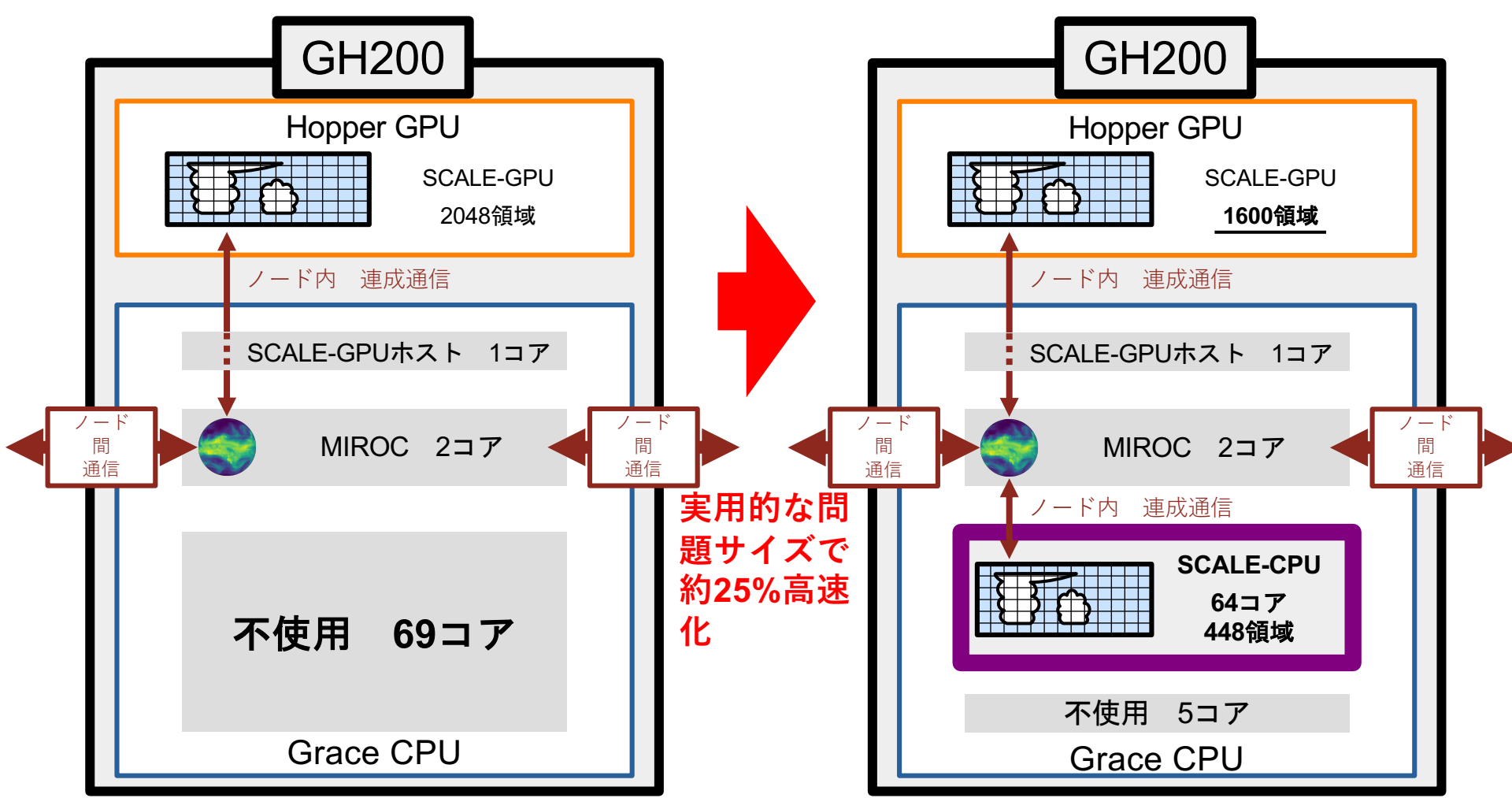
連成

高解像度モデル SCALE-RM v5.3.6（改変版）



- 高解像度
- 雲等の小規模過程を担当
- 8192個の小領域を**ほぼ独立に**計算
- メモリ（キャッシュ）律速

[山崎, 2025, HPC研究会,コンピュータサイエンス領域奨励賞]



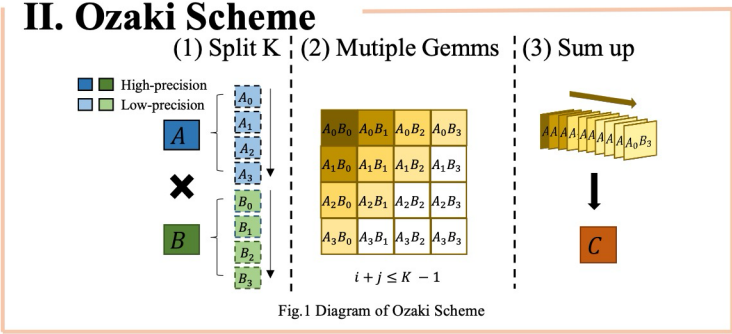
[山崎, 2025, HPC研究会]

Hopper GPUで解くべき問題サイズが減るため、所要時間が短縮

尾崎スキームの Hopper アーキテクチャ向け最適化調査

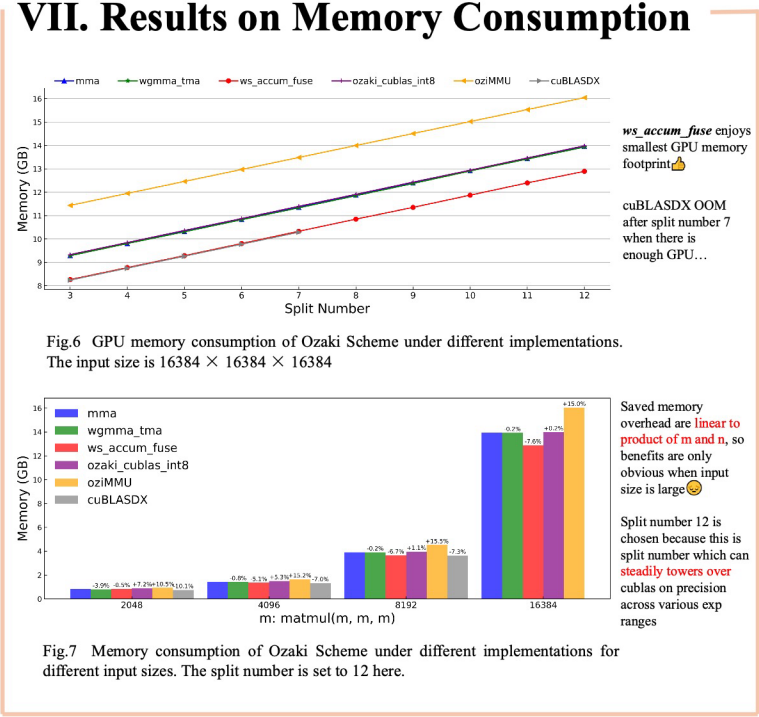
尾崎スキーム

- 低精度計算を活用して高精度の行列乗算を行う方法
  - Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications [K Ozaki et al. 2012]
- AIプロセッサを数値計算に使う方法として注目



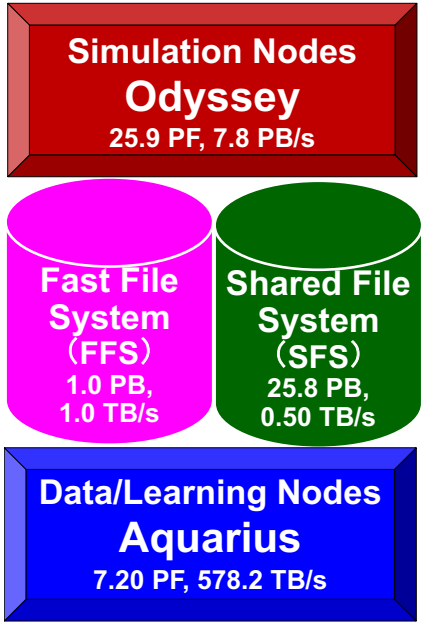
低精度向けAIプロセッサを高速化に利用する手法は今後重要

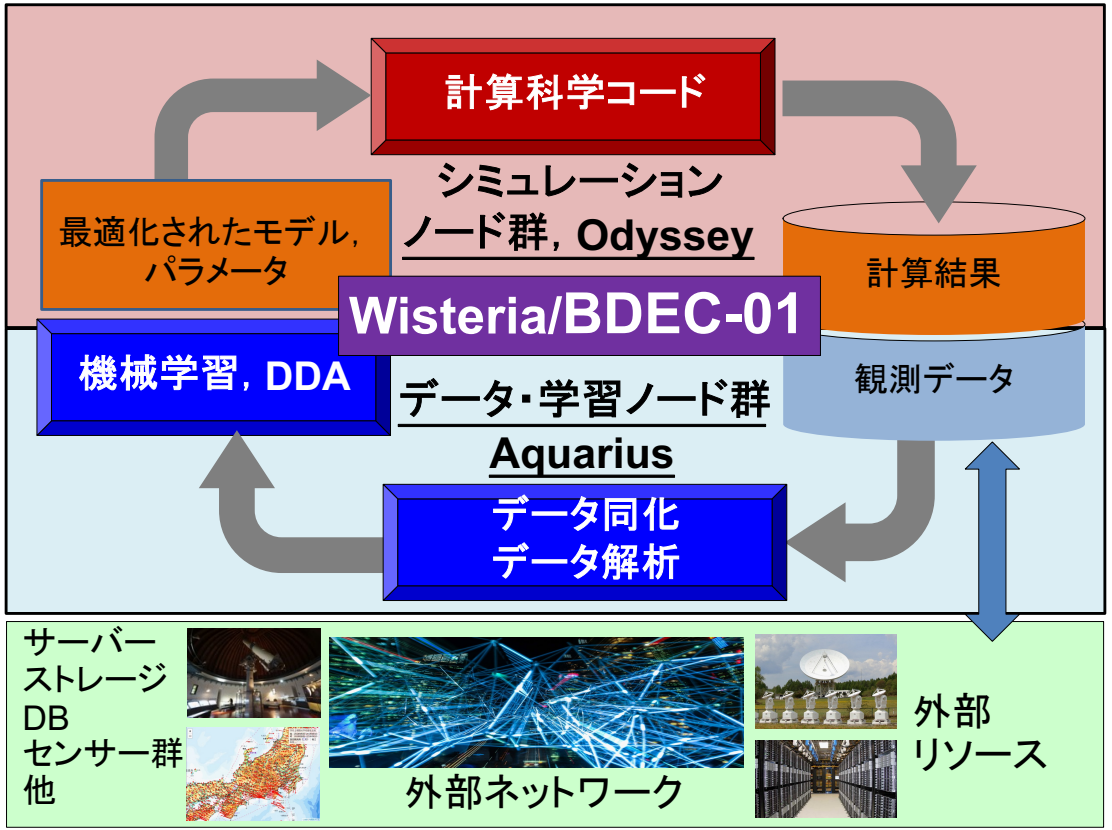
Hopper アーキテクチャに対する高速化手法の提案



[Wang, Shimokawabe, 2026, SCA/HPCAsia 2026, Best Student Poster Award]

Wisteria/BDEC-01 による「計算・データ・学習（S+D+L）」融合の実現と AI for Science 基盤の整備

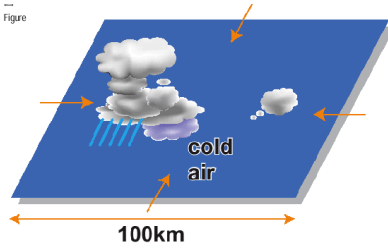




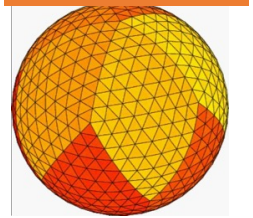
全球雲物理シミュレーション
細分化格子モデル＋粗格子モデル（パラメタリゼーション）
全体の計算量の25%を占める粗格子モデルをAIで置換

- 高精度シミュレーション：Odyssey（NICAM in Fortran）
- 学習・推論：Aquarius（PyTorch: 3-Layer Perceptron

Hierarchical, Hybrid, Heterogeneous
h3-Open-BDEC
Big Data & Extreme Computing

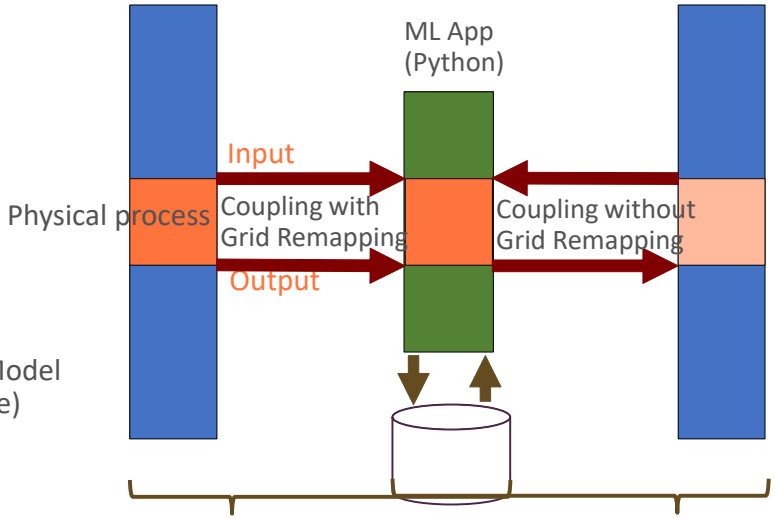


Miyabi-C
Odyssey

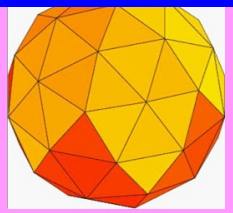


High Resolution Atmospheric Model
(Convection-Resolving Mode)

75%



Miyabi-G
Aquarius



Low Resolution Atmospheric Model
(Convection-Parameterization Mode)

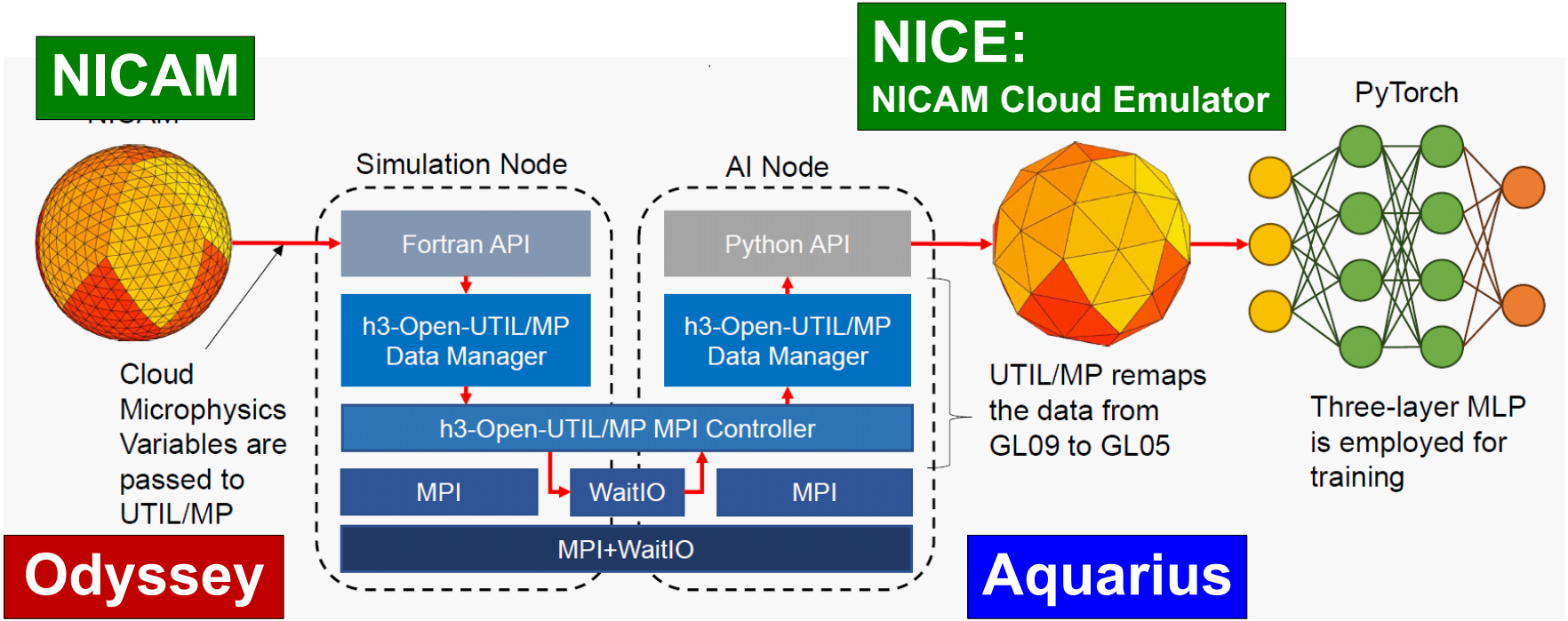
AIへの置換

25% → **~0%**

[荒川，八代（環境研），中島 2022]

Diagram of applying ML to an atmospheric model

AI for Scienceアプリ向けツールの整備



- h3-Open-BDECのMiyabiへの移植、改良
- h3-Open-UTIL/MPのh3-Open-SYS/WaitIOとの連携部分の整備

[Arakawa et al. SCA/HPCAsia 2026 Poster]

性能可搬性プログラミング環境整備

国内はNVIDIA GPUが主流。しかしAMD GPU搭載機もあり。ベンダー中立であることは重要。

- 移植性を確保、Kokkos などが選択肢

現代的なGPU向けプログラミング環境はC++が主流

- NVIDIA (CUDA C++), AMD (HIP), Intel (SYCL) の公式プラットフォームは全てC++ベース。
- 最新機能やライブラリ、ツールはC++向けに優先的に提供
- 複数ベンダーのGPUに対応するKokkos, SYCL等はC++のテンプレートやlambdaで構築

Fortranの今後は？

- Fortran は CUDA Fortran、指示文、言語標準規格での利用のみ。

東大情報基盤センターとしても、Fortranからの移植支援方法を検討中

- 将来的にはFortranからの脱却を目指す
- LLMの活用か？
- Julia?

Solomon: OpenACC/OpenMP統合ライブラリ

簡易にGPU化したい場合: 指示文ベースの実装

- OpenACC: 実質的にNVIDIA GPUのみが対象 (例外: HPE Cray コンパイラ)
- OpenMP target: NVIDIA/AMD/Intel GPUに対応，機能・資料は少ない

プリプロセッサマクロを用いてインターフェースを統合するライブラリを開発

［Miki & Hanawa (2024, IEEE Access)］

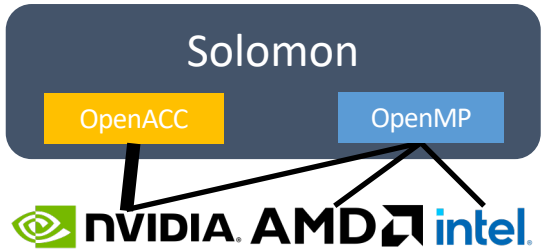
- 対応言語：C/C++（公開中）、Fortran（開発完了、まもなく公開予定）
- 対応しているバックエンド：OpenACC, OpenMP target, OpenMP

OpenACCとOpenMPの両対応した実装

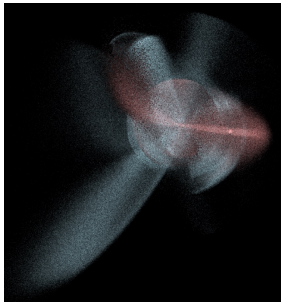
```
#if defined(OFFLOAD_BY_OPENACC)
#if defined(OFFLOAD_BY_OPENACC_KERNELS)
#pragma acc kernels
#pragma acc loop
#else
#pragma acc parallel
#pragma acc loop
#endif
#endif
#if defined(OFFLOAD_BY_OPENMP_TARGET)
#if defined(
    OFFLOAD_BY_OPENMP_TARGET_LOOP)
#pragma omp target teams loop
#else
#pragma omp target teams distribute
    parallel for
#endif
#endif
  for (int i = 0; i < N_i; i++) {
    // loop body A
  }
```

Solomonを用いた等価な実装

```
OFFLOAD()
for (int i = 0; i < N_i; i++) {
  // loop body A
}
```



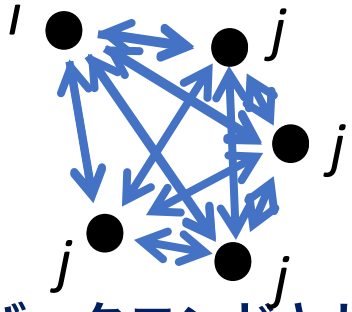
N体計算（重力多体計算）

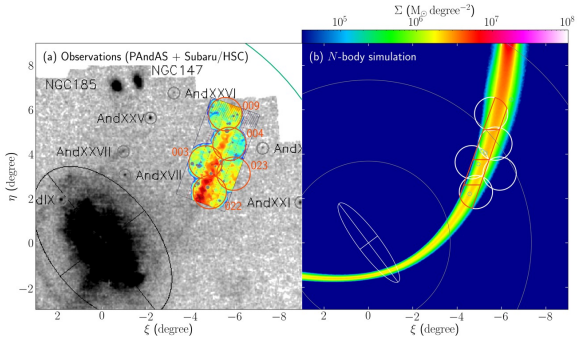


粒子どうしに働く自己重力による系の時間進化を，運動方程式に基づいて計算

- データ量: $\mathcal{O}(N)$
- 重力計算: $\mathcal{O}(N^2)$
- 時間積分: $\mathcal{O}(N)$

$$\boldsymbol{a}_i = \sum_{\substack{j=0 \\ j\neq i}}^{N-1} \frac{G m_j \left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)}{\left(\left|\boldsymbol{x}_j - \boldsymbol{x}_i\right|^2 + \epsilon^2\right)^{3/2}}$$





N体業界的によく使う用語

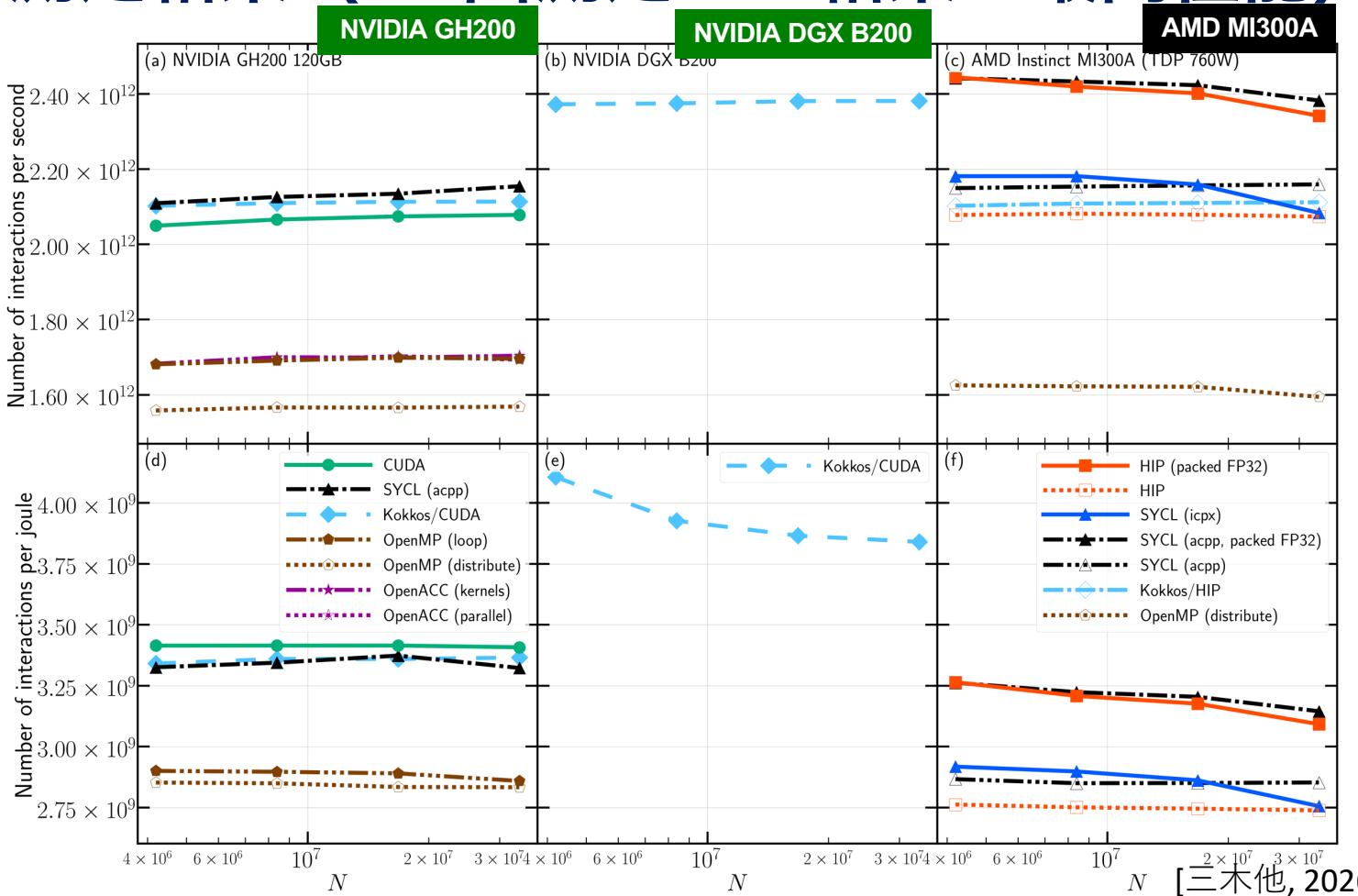
- i-粒子: 重力を受ける粒子
- j-粒子: 重力を及ぼす粒子

直接法のソルバーはツリーコードのバックエンドとして使えるので，高速化しておく価値が高い（また，衝突系N体計算であれば直接法を用いる）

無衝突系であれば全て単精度浮動小数点演算でも十分な精度が得られる

- 本研究では，全ての演算に単精度浮動小数点演算を用いた

性能測定結果（10回測定した結果の最高性能）



[三木他, 2026, HPC研究会]

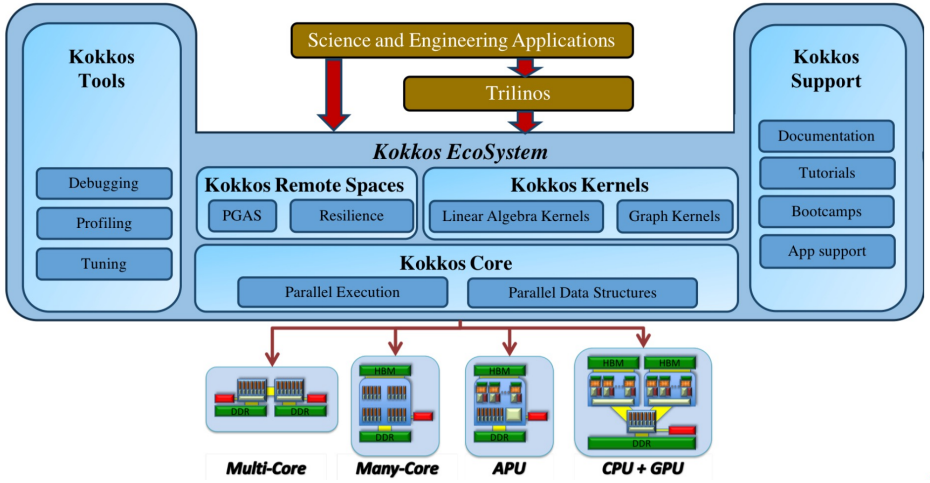
Kokkos の特徴

性能可搬性

- 単一のソースコードでCPU、GPU
  など多様なプロセッサに対応（ベンダー非依存）
- ハードウェア毎の書き換えが不要

生産性の高い並列プログラミング

- C++のラムダ式で並列処理を直感的に記述
- parallel_for (ループ)、parallel_reduce (集計)、
  parallel_ scan (累積和) といった、頻出する並列パターンをサポート



[From Kokkos document]

アーキテクチャの抽象化

- CUDA, OpenMP, HIP, SYCL等をバックエンドとし、ハードウェアの詳細を隠蔽

データレイアウトの最適化

- Kokkos::Viewで多次元配列を柔軟に管理
- データレイアウト（列優先/行優先など）をハードウェア（CPU/GPU）に最適化できる

充実したエコシステム

- デバッガやプロファイラなどのツール群、数学ライブラリなど提供

米DOE国研が開発、CEA（フランス原子力公社）全面採用しFortranコードをC++へ以降

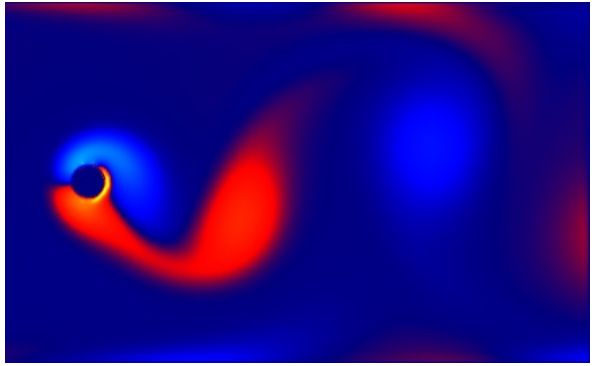
東大はKokkosを提供する高性能計算ソフトウェア財団(HPSF)に加入し、利用推進

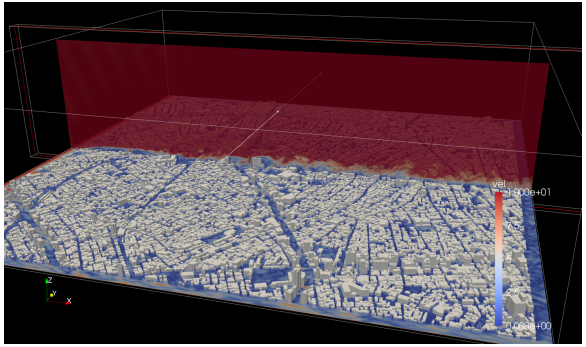
Kokkosの適用事例：流体計算

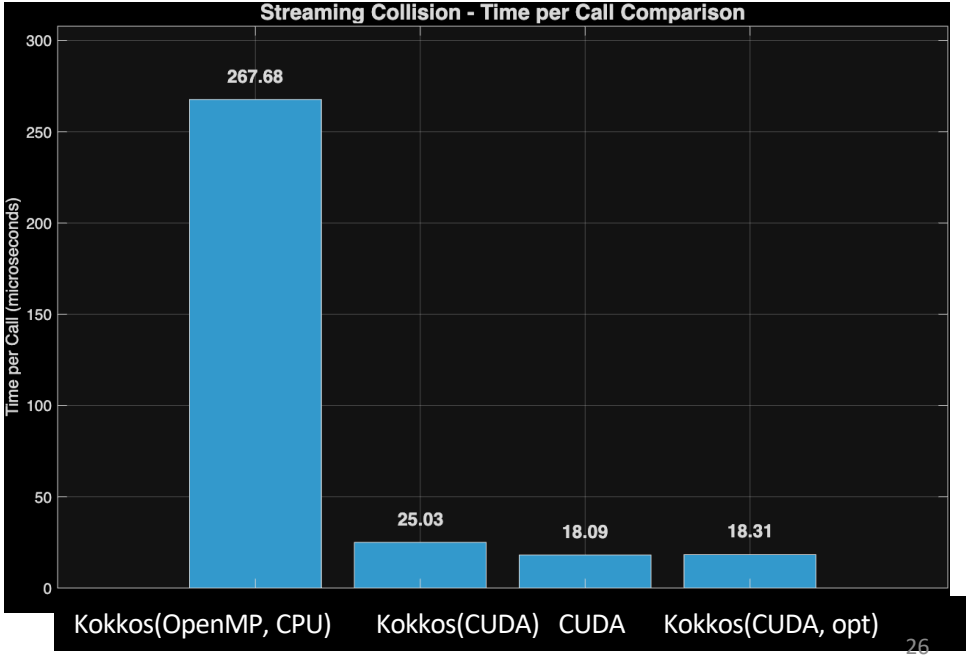
A Discrete-velocity Boltzmann equation

$$f_i(x + c_i\Delta t, t + \Delta t) = f_i(x,t) - \frac{1}{\tau}(f_i(x,t) - f_i^{eq}(x,t)) + F_i$$

streaming　　collision　　forcing term



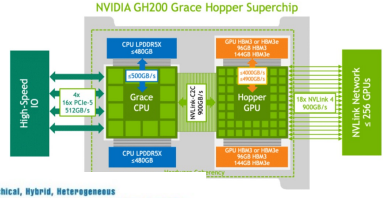




まとめ：東京大学R7年度活動成果

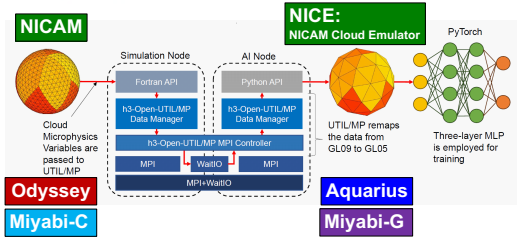
計算科学アプリケーションのGPU化向けコード整備

- 構造格子系、非構造格子系、多体系のGPUコードの整備
  - OpenACC、OpenMP target、言語標準、Kokkosによる GPU化の実装例の作成と性能評価、近日結果を公開予定
  - CPU-GPU同時効率的利用法の検討（Miyabi-G）
- Ozaki Schemeの NVIDIA Hopper アーキテクチャ向け最適化の調査



AI for Scienceアプリケーション向けツールの整備

- Wisteria/BDEC-01上で稼働している、革新的基盤ソフトウェア h3-Open-BDECのMiyabiへの移植、改良
  - h3-Open-UTIL/MPのh3-Open-SYS/WaitIOとの連携部分の整備
  - 全球大気モデルNICAMとpythonプログラムを用いた連成実行実施



性能可搬性プログラミング環境整備

- 指示文ベースのGPUオフロードのための統一インターフェース Solomon（C++対応）のFortran向け実装開発、近日公開予定