地震の大規模シミュレーション –データ駆動型手法による高度化

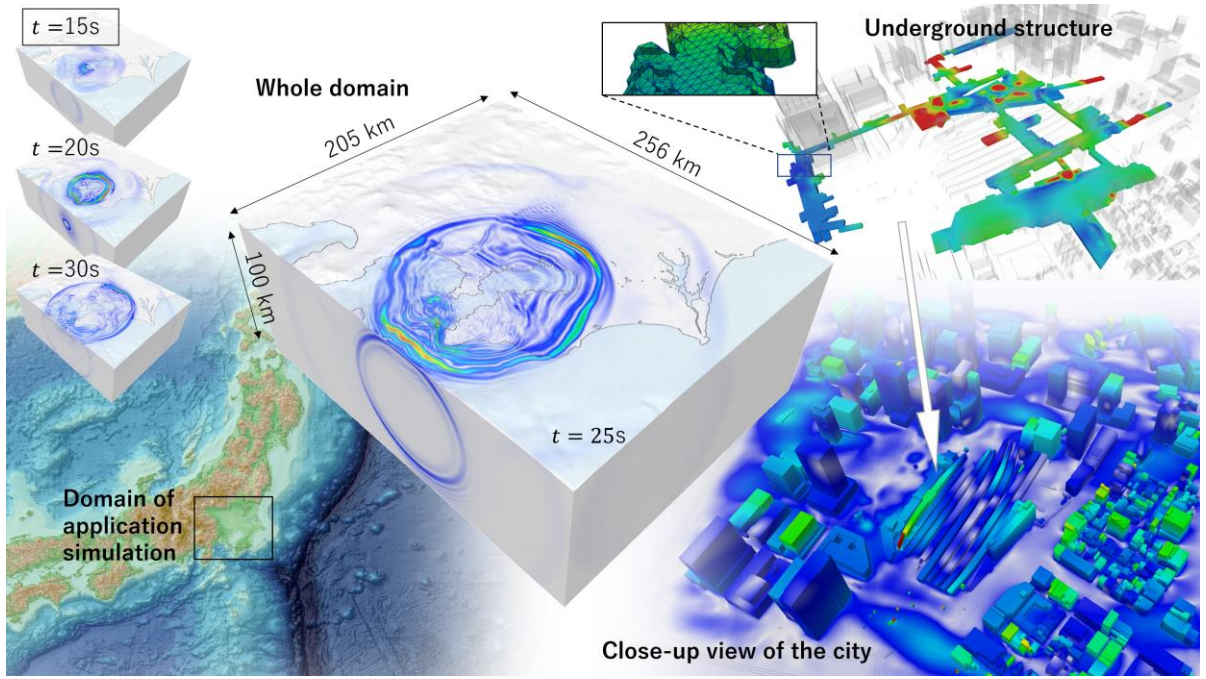
藤田 航平
東京大学 地震研究所 計算地球科学研究センター
東京大学 工学系研究科 社会基盤学専攻

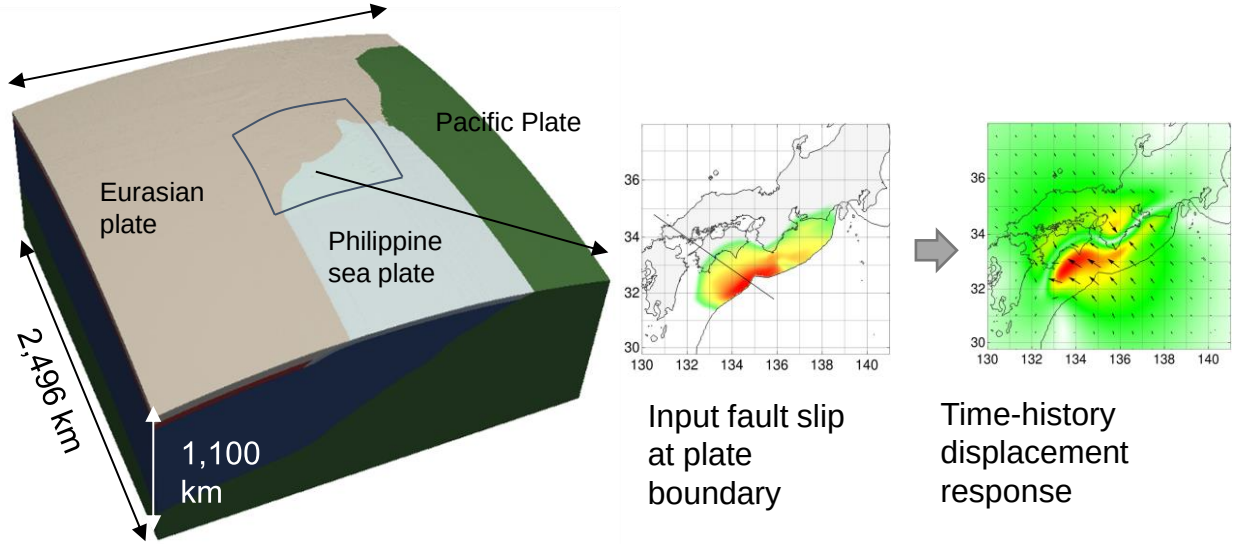
はじめに：地震シミュレーションにおけるデータ駆動型手法の活用

- 地殻変動・地震動・地盤増幅などのPDE (partial differential equation: 偏微分方程式)に基づく時刻歴シミュレーション
  - 高コスト：次世代コンピューター・システムに向けた、高効率（計算時間・必要エネルギー量）な手法が望まれる
- これまでデータ駆動型手法によりPDEに基づく時刻歴シミュレーションを高速化するアルゴリズムを開発してきた
  - 過去の時刻歴データを用いたデータ駆動型手法により、PDEに基づく時刻歴シミュレーションにおいて用いられる反復法ソルバーの初期解を高精度で予測
  - ソルバーの収束までの反復回数が削減されることで、精度を落とさずに高速化を実現

データ駆動型手法による時刻歴シミュレーションの高速化

- 地震波伝播、粘弾性地殻変動、常時微動シミュレーションなど、複数種類のPDEに基づく時刻歴シミュレーションに適用可能





Seismic wave propagation@full Fugaku (152352 nodes)
Up to 25-fold speedup from previous solver without data-driven method
Ichimura et al. HPC Asia 2022 [Best Paper]

Crustal deformation@Fugaku (73728 nodes)
Up to 76-fold speedup with combination with other methods
Fujita et al. ScalaH 2022

データ駆動型手法による時刻歴シミュレーションの高速化

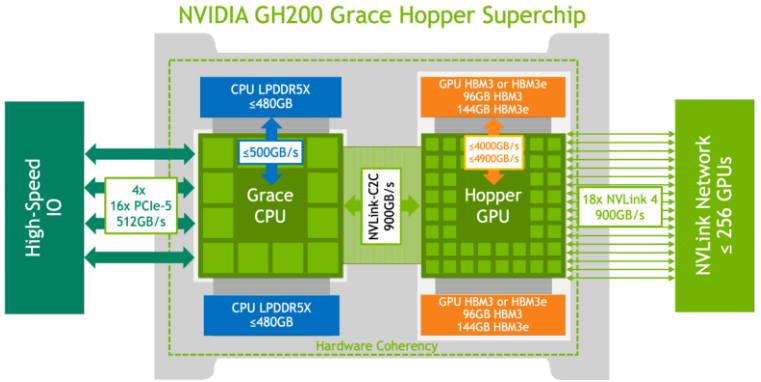
- CPUスパコン、GPUスパコン、また、最近ではCPU-GPU間に高速インターコネクトを持つシステム向けに手法を開発

Fugaku (CPU) → A100 (GPU) → GH200 [single node] (CPU+GPU) → CSCS Alps [GH200 NVL4] (CPU+GPU)

Ichimura et al.: HPC Asia 2022
Fujita et al.: ScalaH@SC22

Murakami et al.:
ICCS 2023

Ichimura et al.:
GTC 2024



Ichimura et al.:
Accepted for
WACCPD@SC24

Data-driven part on memory-rich CPU,
Solver part on high-performance GPU

本日の内容

- データ駆動手法によるPDEに基づく時刻歴シミュレーションの高速化手法の開発@富岳
  - Tsuyoshi Ichimura, Kohei Fujita, Kentaro Koyama, Ryota Kusakabe, Yuma Kikuchi, Takane Hori, Muneo Hori, Lalith Maddegedara, Noriyuki Ohi, Tatsuo Nishiki, Hikaru Inoue, Kasuo Minami, Seiya Nishizawa, Miwako Tsuji, Naonori Ueda, HPC Asia 2022
  - Kohei Fujita, Tsuyoshi Ichimura, Sota Murakami, Takane Hori, Muneo Hori, Lalith Maddegedara, Naonori Ueda, SCALAH@SC22
  - を紹介
- CPUとGPUの同時使用によるPDEベースの波動シミュレーションの高速化@GH200システム
  - Tsuyoshi Ichimura, Kohei Fujita, Maddegedara Lalith, Muneo Hori, GTC2024
  - Tsuyoshi Ichimura, Kohei Fujita, Muneo Hori, Lalith Maddegedara, Jack Wells, Alan Gray, Ian Karlin, John Linford, WACCPD@SC24 (accepted)
  - を紹介

ターゲット問題

- PDEベースの時刻歴シミュレーション
  - 以下の連立方程式を各タイムステップにおいて求解

$$A^n \delta u^n = f^n \qquad (1)$$

  - A^n：既知の疎行列、δu^n：未知ベクトル、f^n：タイムステップnにおける既知ベクトル
  - A^nが大規模な疎行列となるため、反復法を用いて式(1)を並列計算システム上で求解
- 大規模問題を求解可能なアルゴリズムとする
  - Equation-basedアプローチとデータ駆動型アプローチを効率的に活用することを考える
  - 並列性能の担保のため、負荷分散と通信コストも同時に留意

Equation-basedアプローチとデータ駆動型アプローチ

- Equation-basedアプローチ(e.g., multi-grid)
  - 低次モードは低コストで求解できるものの、高次モードの求解は高コストとなる
- データ駆動型アプローチ
  - 学習データの規模と出力の複雑度に依存して計算コストと精度が変化（e.g., 全領域の高次応答を学習するには膨大なコストが必要だが、対象領域が局所的であればコストは小さい）
- これらの方法を効果的に組み合わせることで、解空間に含まれる様々なモードを効率的に求解
  - 最終的な解の精度を保証するスキーム内において、低次モードにはequation-basedアプローチ、局所的な高次モードにはデータ駆動型アプローチを用いて求解

データ駆動型手法による初期解予測

- 過去タイムステップの求解結果を用いて、反復法ソルバーの前段階で高次モードを予測
  - 時間発展行列Aを$X^{n-1} = AX^{n-2}$として推定（ここで$X^{n-1} = [x^{n-1}, x^{n-2}, \ldots, x^{n-s}]$は直近の$s$タイムステップの結果$x^n = \delta u^n - \delta u^n_{adam}$）
  - データ駆動型手法の精度を悪化させるtransientな低次モードはequation-basedな予測手法を用いて除き(δu^n_{adam}はAdams-Bashforth法)、データ駆動型手法によりnon-transientな高次モードを求める:

$$\delta u^n_{ini} = \delta u^n_{adam} + A\left(\delta u^{n-1} - \delta u^{n-1}_{adam}\right)$$

- 過去データの学習と初期解予測はMPI領域分割内で独立して実施
  - 初期解予測においてMPI通信は不要

データ駆動型手法による初期解予測の詳細

- 学習フェーズ
  - sタイムステップの入力データ$X = [x^1, x^2, \dots, x^s]$と出力データ$Y = [y^1, y^2, \dots, y^s]$に対し、$P = XU$が直交基底となるように$s \times s$の上三角行列$U$を修正グラムシュミット法で求める
- 予測フェーズ
  - 入力xを$x = Pc + r$と展開（ここで、$c = P^T x$）
  - 入力xに対する応答を$y \approx YUc = YUP^T x = (YUU^T X^T)x$と推定する
- 直近のsタイムステップのデータを使い、学習・予測を毎タイムステップ実施

データ駆動型予測手法の特性

- 通信を伴わないため、高いスケーラビリティが期待される
- 全てのデータアクセスが連続となるため、高効率で計算可能

- 初期解推定で解ききれなかった誤差を効率的に低減するスケーラブルな反復法ソルバーと組み合わせることで、高速でスケーラブルなシミュレーション手法が可能になると期待
  - 対象行列A^nの特徴に応じて適切なソルバーを選択

対象問題：粘弾性地殻変動解析

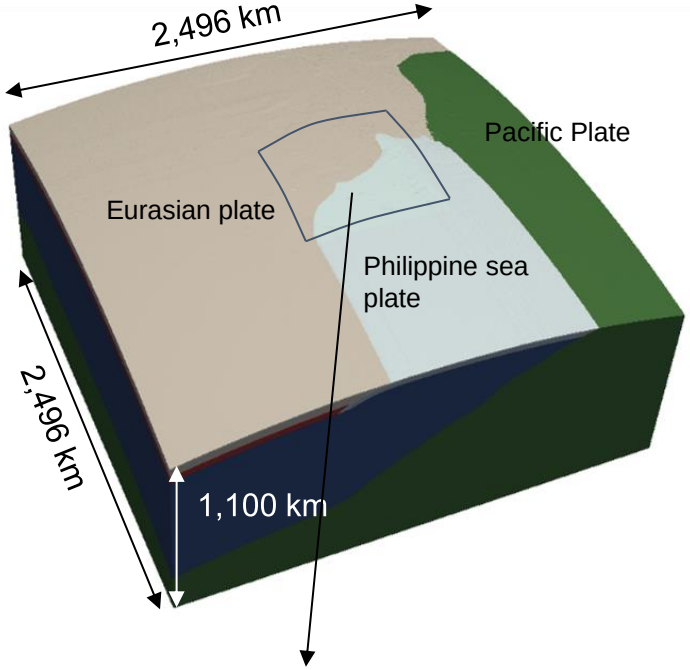
- 断層すべりに対する地殻の粘弾性応答を以下の支配方程式に基づいて計算

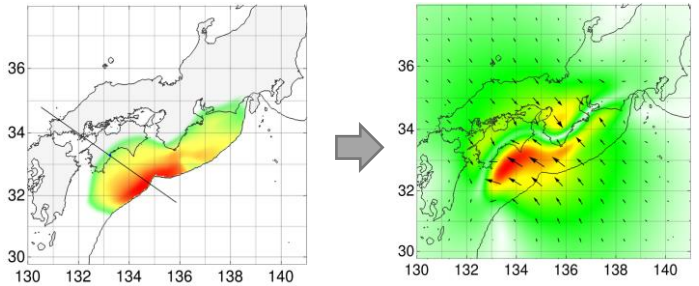
$$\sigma_{ij,j} + f_i = 0,$$

$$\dot{\sigma}_{ij} = \lambda \dot{\epsilon}_{kk} \delta_{ij} + 2\mu \dot{\epsilon}_{ij} - \frac{\mu}{\eta}\left(\sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}\right),$$

$$\epsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)$$

- 有限要素法による離散化・[1]に基づく時間積分により、対象問題は$\boldsymbol{K}\delta\boldsymbol{u}^n = \boldsymbol{f}^n$を各タイムステップで解く問題に帰着
  - ここで$\boldsymbol{K}$は収束性の悪い正定値対象な疎行列となる





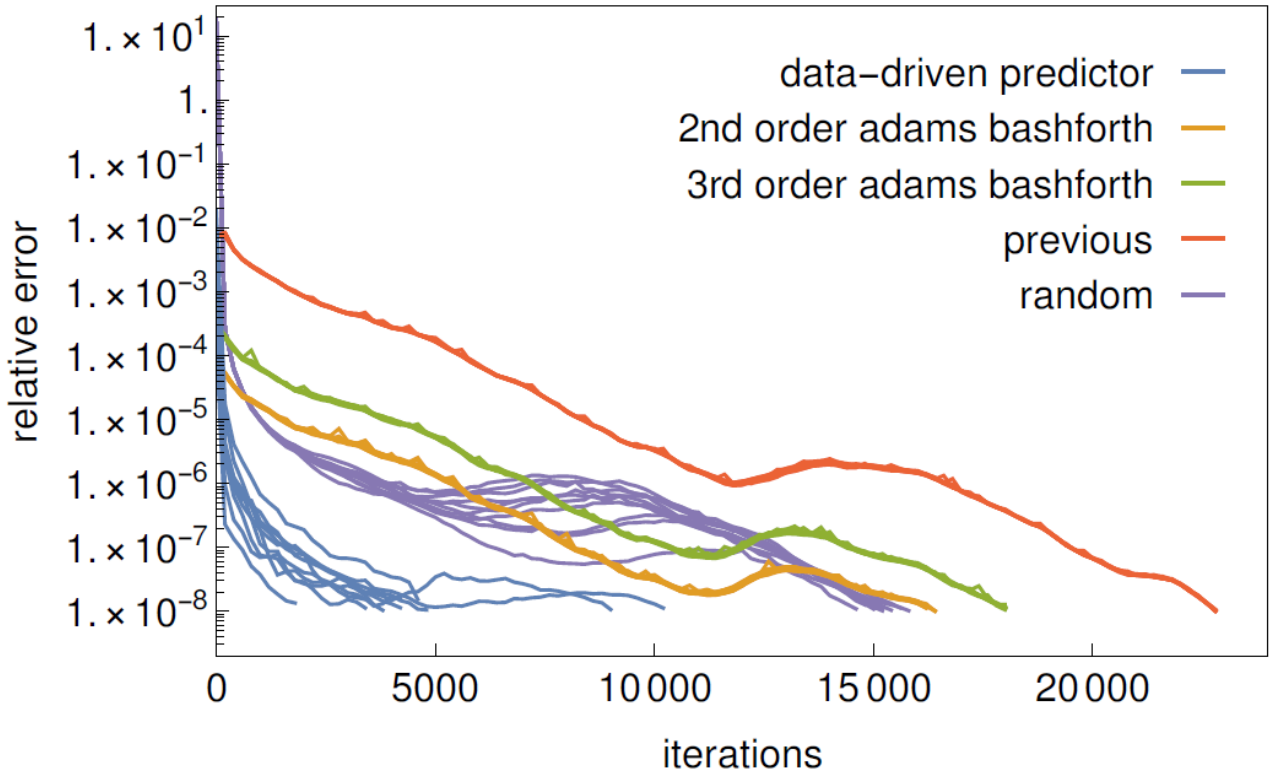
Input fault slip at plate boundary

Time-history displacement response

[1] T. Ichimura, R. Agata, T. Hori, K. Hirahara, C. Hashimoto, M. Hori, and Y. Fukahata, An elastic/viscoelastic finite element analysis method for crustal deformation using a 3-D island-scale high-fidelity model, *Geophysical Journal International*, 2016.

データ駆動型予測手法による初期解の改善

- 標準的な共役勾配ソルバー（PCGE：3x3ブロックヤコビ法による前処理を用いた共役勾配法）の初期解を変更した場合の収束履歴を比較
- 通常の初期解予測手法(2nd order Adams-Bashforth method)と比べ、反復回数が1/3.2に減少



Convergence history of solver for ten time steps

Data-driven predictor (use $s = 16$ steps for the data-driven predictor):

$$\delta u_{ini}^{n} = \delta u_{adam}^{n} + A\left(\delta u^{n-1} - \delta u_{adam}^{n-1}\right)$$

2nd order Adams-Bashforth:

$$\delta u_{ini}^{n} = u^{n-3} - 3u^{n-2} + 2u^{n-1}$$

3rd order Adams-Bashforth:

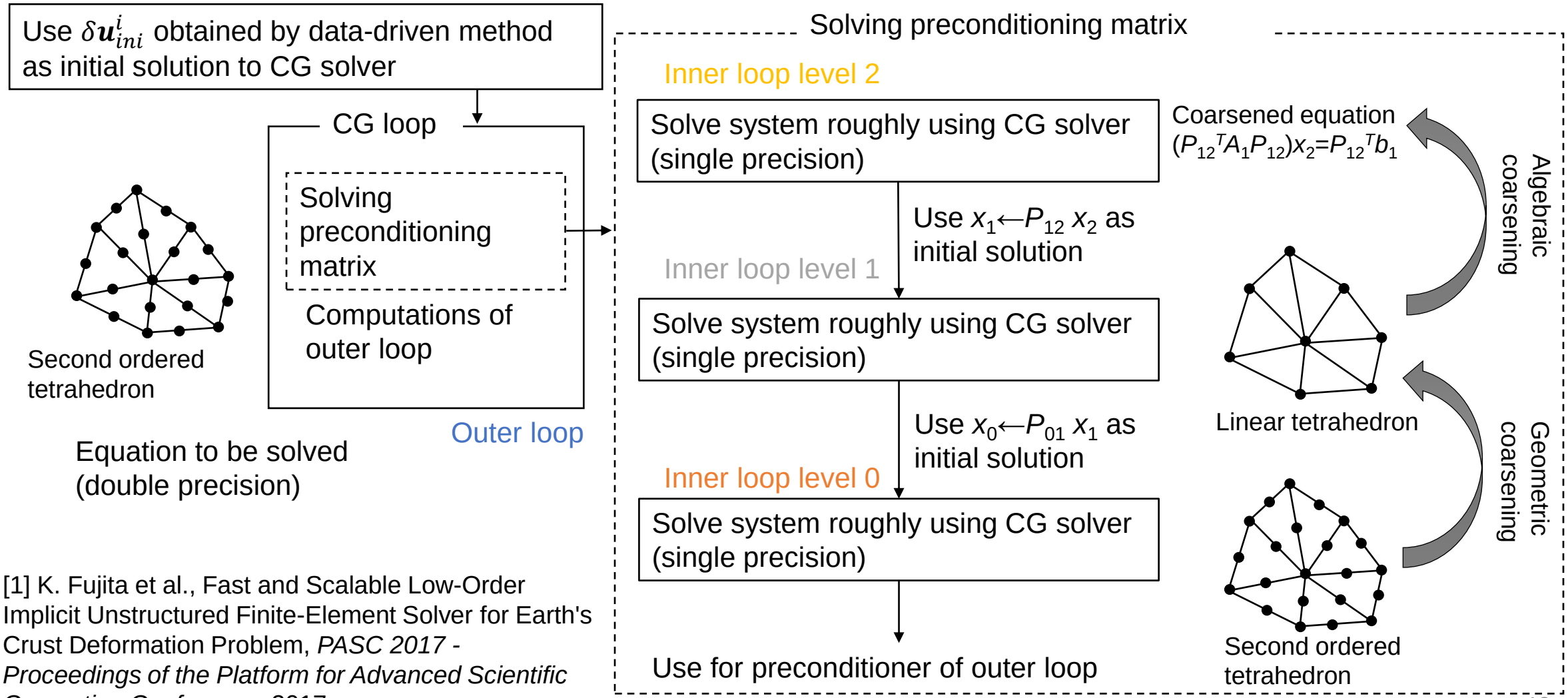
$$\delta u_{ini}^{n} = 7u^{n-1} - 18u^{n-2} + 16u^{n-3} - 5u^{n-4}$$

Previous: $\delta u_{ini}^{n} = u^{n-1} - u^{n-2}$

Random: $\delta u_{ini}^{n} = random(-1{:}\,1)$

反復法ソルバー

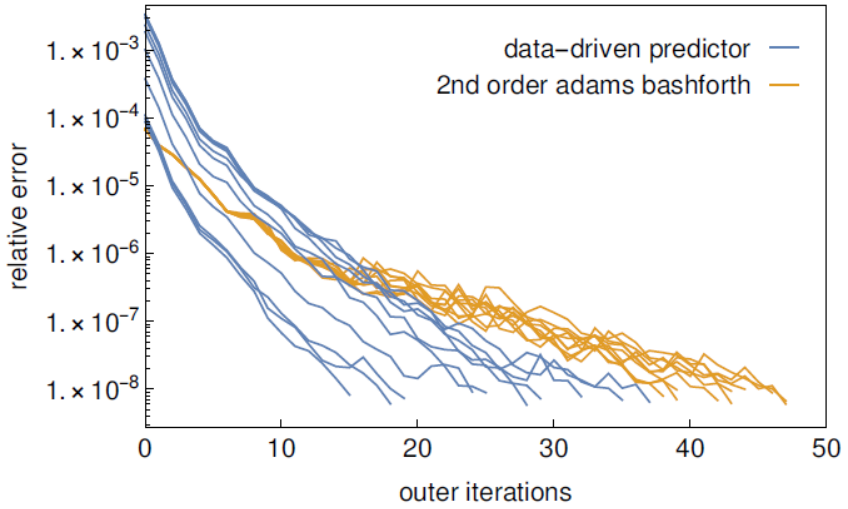
- データ駆動型の初期解予測手法と、低周波成分の求解に適したマルチグリッドソルバー [1]を組みあわせる
- 共役勾配法ソルバーの中で精度混合演算、および、代数・幾何マルチグリッドを活用



[1] K. Fujita et al., Fast and Scalable Low-Order Implicit Unstructured Finite-Element Solver for Earth's Crust Deformation Problem, *PASC 2017 - Proceedings of the Platform for Advanced Scientific Computing Conference,* 2017

データ駆動型初期解予測手法とマルチグリッドソルバーを組み合わせた場合の性能@富岳576ノード

- 初期解予測によりマルチグリッドソルバーの反復数も減少
  - 初期解予測にかかる時間が短いため、通常の初期解予測手法（2nd order Adams Bashforth method）を活用したマルチグリッドソルバー比で15.3/5.7 = 2.7倍の高速化を実現
- 標準的な解析手法と比べて439/5.7 = 77倍の高速化を実現

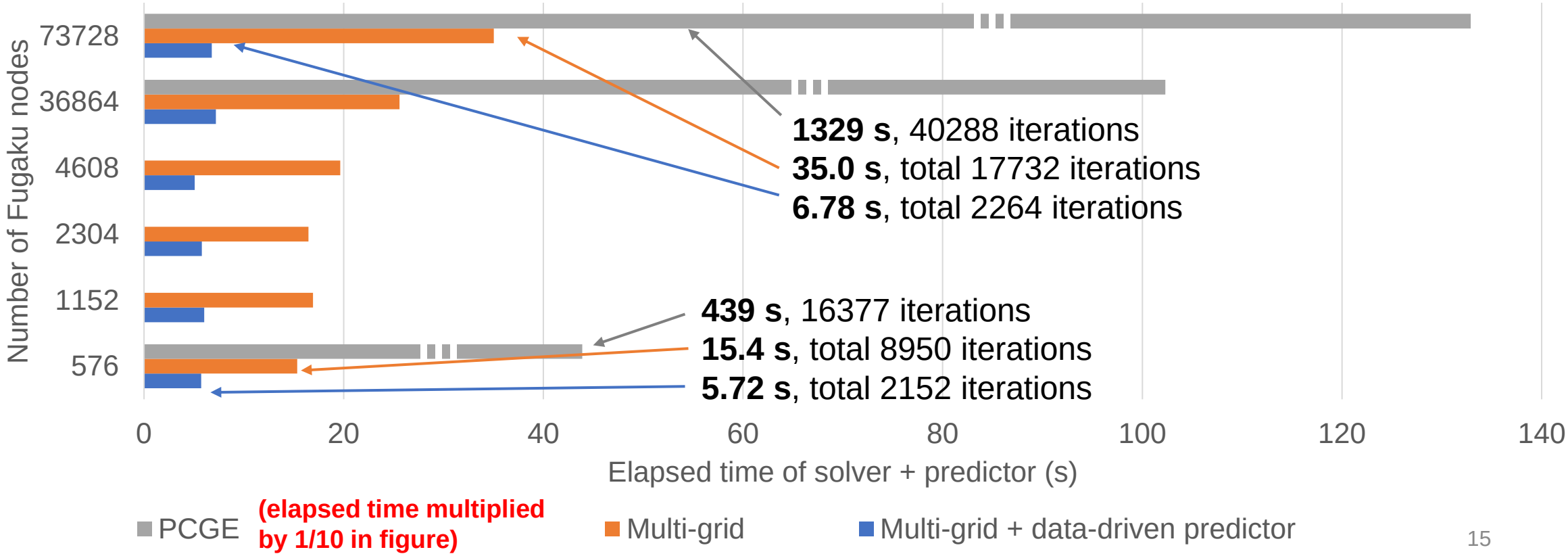


Convergence history of multi-grid solver for ten time steps

Solver	elapsed time (s)			iterations				FLOPS efficiency
	total	solver	predictor	outer	inner loop 0	inner loop 1	inner loop 2	(to FP64 peak)
PCGE	439.2	439.2	-	16377	-	-	-	7.39%
PCGE + developed predictor	137.5	137.2	0.21	5062	-	-	-	7.31%
Multi-grid solver	15.35	15.35	-	42.8	89.7	956	7861	6.83%
Developed solver	5.72	5.39	0.21	26.2	58.6	174	1893	8.71%

Weak scaling

- 富岳73728ノードまでのスケーラビリティを計測
  - 初期解予測手法にかかる時間は並列規模によらず一定 (全モデルで0.21 s)
  - 大規模問題でも反復数の削減効果が得られ、高速化につながっている



データ駆動型手法による時刻歴シミュレーションの高速化

- CPUスパコン、GPUスパコン、また、CPU-GPU間に高速インターコネクトを持つシステム向けに手法を開発

Fugaku (CPU)

A100 (GPU)

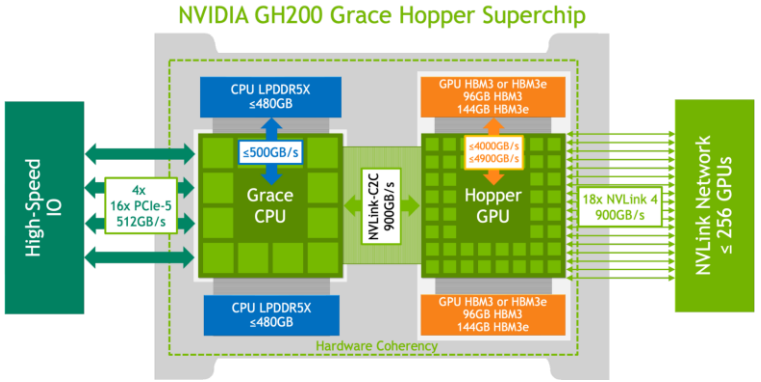
GH200 [single node] (CPU+GPU)

CSCS Alps [GH200 NVL4] (CPU+GPU)

Ichimura et al.: HPC Asia 2022
Fujita et al.: ScalaH@SC22

Murakami et al.: ICCS 2023

Ichimura et al.: GTC 2024



Ichimura et al.: Accepted for WACCPD@SC24

Data-driven part on memory-rich CPU,
Solver part on high-performance GPU

データ駆動型手法による時刻歴シミュレーションの高速化@CPU+GPU system

- 最近の計算能力を利用した計算高速化
  - 大容量CPUメモリ：データ駆動型手法に用いる大規模データの保存が可能
  - 高速GPU＋低プログラミングコスト（OpenACCなど）の開発環境： PDEベースの時刻歴シミュレーションにGPUが活用されている
- PDEに基づく時間発展問題の高速化のためのデータ駆動型手法の利用
  - 過去タイムステップのデータを保存するために大容量メモリが必要
  - GPUメモリは比較的小さいため、問題サイズを維持しながらデータ駆動型手法によってGPUベースの解析を高速化することは難しい
- CPU-GPU間に高速インターコネクトを持つシステムのための手法を開発
  - データ駆動型手法とシステムのCPU-GPU間の高速インターコネクトを活用することで、大容量メモリを搭載したCPUと高性能GPUの両方の性能を引き出す

ターゲット問題

- CPU-GPU間に高速インターコネクトを持つシステムにおいて、多数ケースの時刻歴シミュレーションを解く
- ここで、各時間ステップitにおいて線型方程式 $\boldsymbol{A}\boldsymbol{x}^{it} = \boldsymbol{f}^{it}$ を求解
  - $\boldsymbol{A}$: 正定値対象な疎行列、$\boldsymbol{f}^{it}$: 入力ベクトル、$\boldsymbol{x}^{it}$: 出力ベクトル
  - 大規模問題となるため反復法ソルバーを活用
- 解の精度を担保しつつソルバーを高速化
  - 反復法ソルバーの初期解を以下のように予測

$$\overline{\boldsymbol{x}}^{it} = predictor(\boldsymbol{X}^{it}, \boldsymbol{F}^{it}, \boldsymbol{f}^{it}),$$

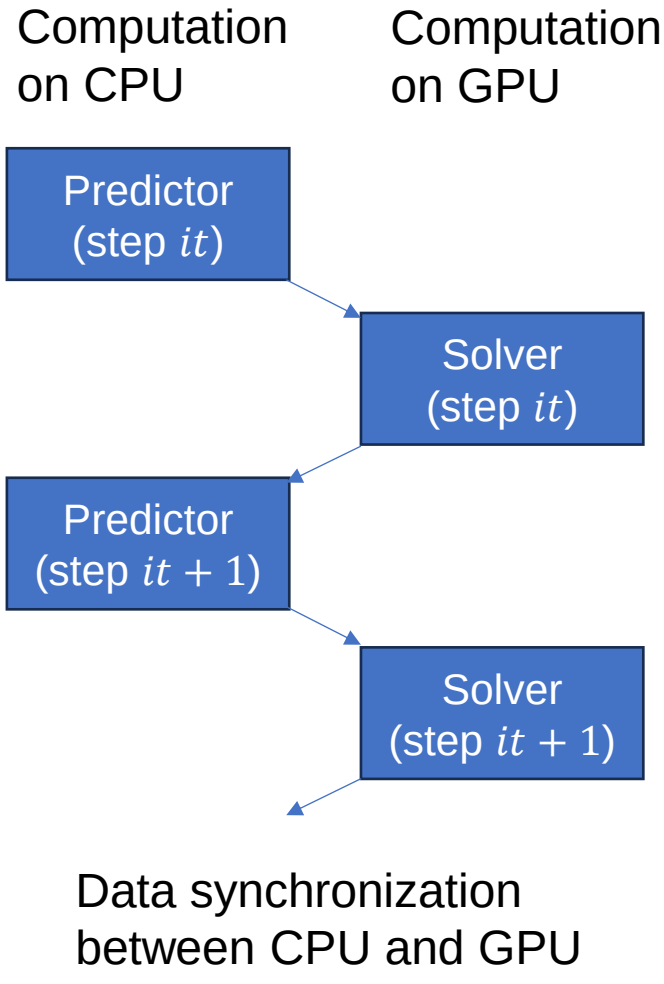
ここで直近のsタイムステップのデータを学習データとしたデータ駆動型手法を活用

$$\boldsymbol{X}^{it} = \left\{\boldsymbol{x}^{it-s}, \boldsymbol{x}^{it-s+1}, \dots, \boldsymbol{x}^{it-1}\right\}, \boldsymbol{F}^{it} = \left\{\boldsymbol{f}^{it-s}, \boldsymbol{f}^{it-s+1}, \dots, \boldsymbol{f}^{it-1}\right\}$$

  - 初期解が高精度で予測されることで、ソルバーの反復数削減と実行時間短縮につながる

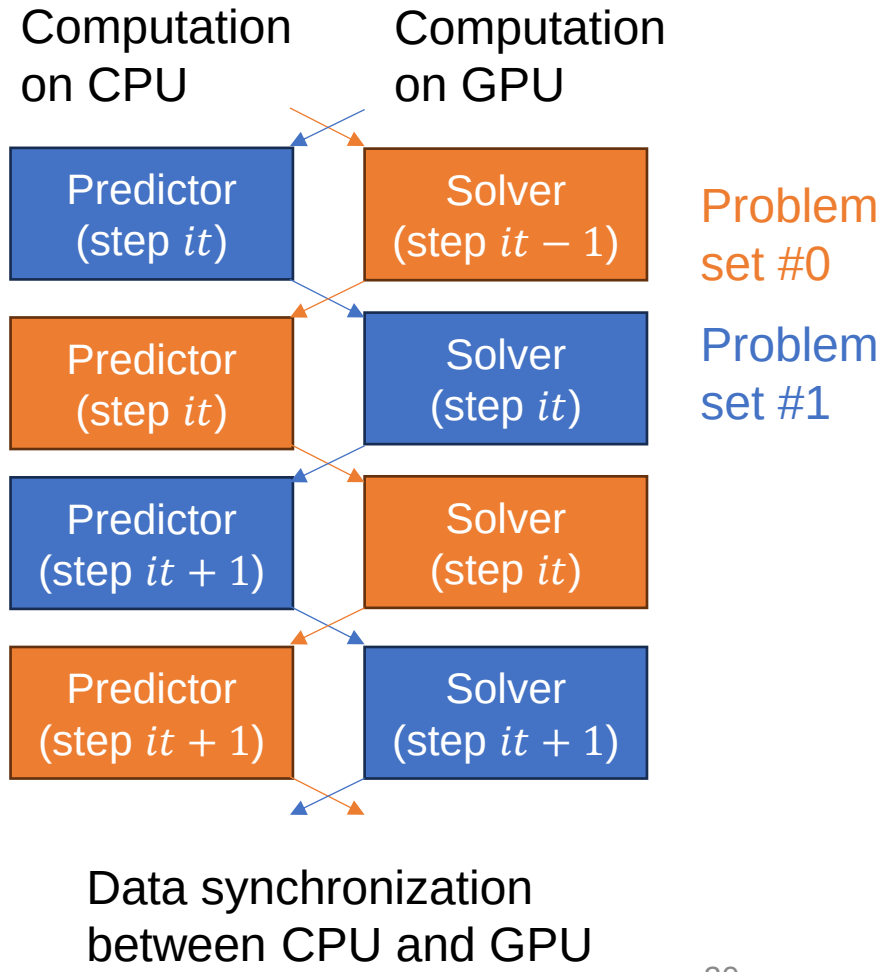
CPU-GPU環境におけるデータ駆動型初期解予測手法の活用

- 高精度な初期解推定には多数の時間ステップのデータが必要
  - データ格納のため大容量メモリが必要
- データ駆動型手法の通常の活用方法
  - CPUメモリ上に学習データを保存し、CPUでデータ学習と予測計算を実行
  - 初期解推定計算の前後でCPUとGPU間でデータを同期



CPU-GPU環境におけるデータ駆動型初期解予測手法の活用

- 提案手法：CPUとGPUを活用し2セットの問題を同時に求解
  - CPU上で一つ目の問題の初期解を推定
  - GPU上で二つ目の問題の連立方程式を求解
  - 初期解推定・方程式求解の前後でCPU-GPU間でデータを同期
- 実行時間中継続してCPUとGPUの同時計算が可能に
  - 初期解推定@CPUと方程式求解@GPUの実行時間は同程度、かつ、高速CPU-GPUインターコネクトによりデータ同期時間は無視できる



Tsuyoshi Ichimura, Kohei Fujita, Maddegedara Lalith, Muneo Hori, GTC 2024

数値実験:ターゲット問題

- 線形動的弾性問題において性能を評価

$$\rho\ddot{\boldsymbol{u}} - (\nabla \cdot \boldsymbol{c} \cdot \nabla) \cdot \boldsymbol{u} = \boldsymbol{f}$$

  - $\boldsymbol{u}$: displacement, ρ: density, $\boldsymbol{c}$: elasticity tensor, $\boldsymbol{f}$: outer force
- 四面体二次要素による空間離散化・Newmark-β法による時間積分により以下に帰着

$$\left(\frac{\boldsymbol{M}}{dt^2} + \frac{\boldsymbol{C}}{dt} + \boldsymbol{K}\right)\boldsymbol{u}^{it} = \boldsymbol{f}^{it} + \boldsymbol{C}\boldsymbol{v}^{it-1} + \boldsymbol{M}\left(\boldsymbol{a}^{it-1} + \frac{4}{dt}\boldsymbol{v}^{it-1}\right)$$

- 各時間ステップitにおいて上記方程式を求解
  - Baseline method: ブロックヤコビ前処理を用いた共役勾配法(メモリ格納型の疎行列ベクトル積手法を活用)
  - 提案手法:データ駆動型手法 [1] によりbaseline methodにあたえる初期解を推定
  - OpenACC (GPU計算部), OpenMP (マルチコアCPU計算部), MPIを用いて実装

[1] Tsuyoshi Ichimura et al., HPC Asia '22, best paper award

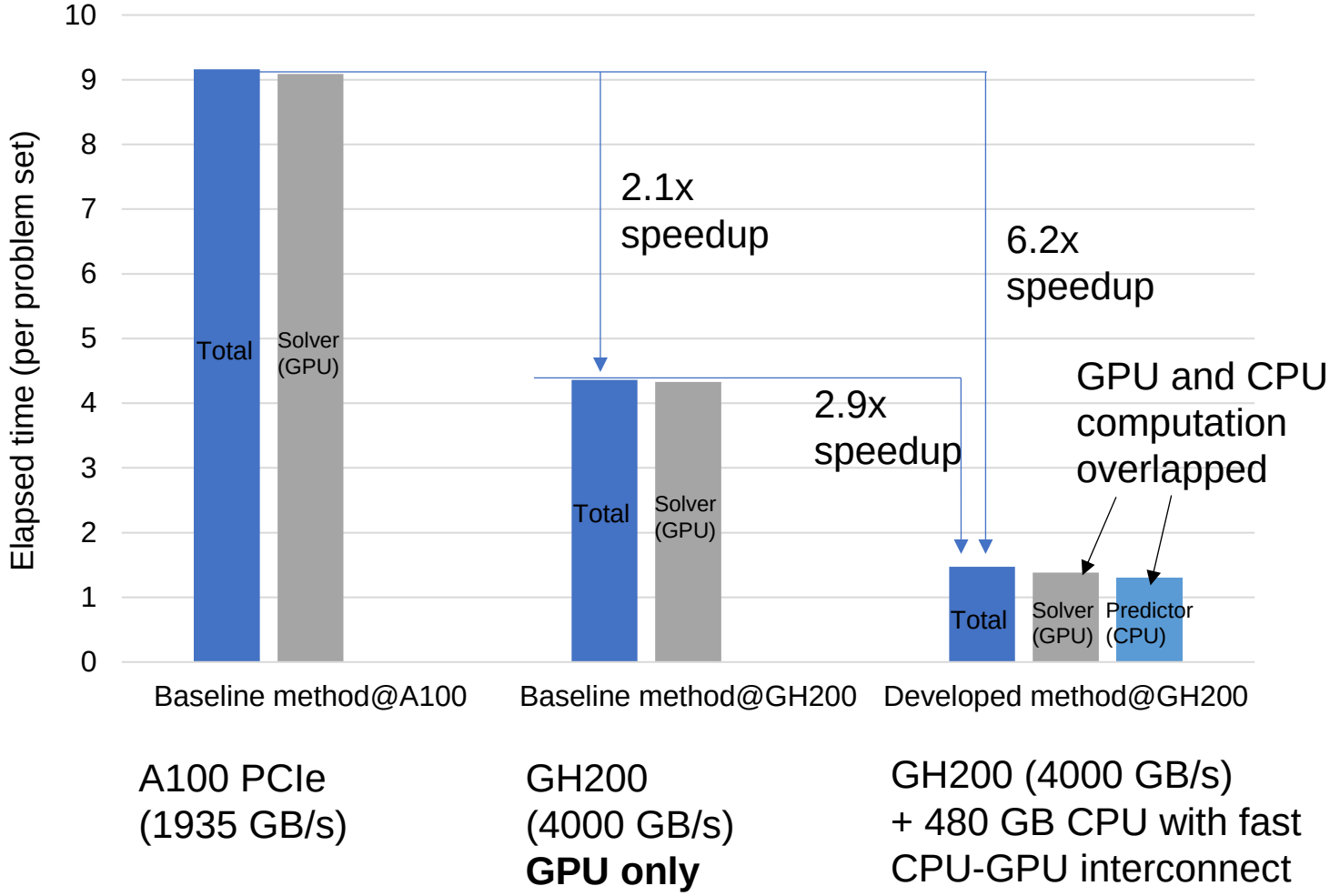
性能測定環境

System	CPU Memory size (bandwidth)	GPU (FP64 peak) Memory size (bandwidth)
Xeon node	4×20-core Xeon Gold 6230 Total of 384 GB (**564 GB/s**) for 4 sockets	NA
A100 node	24-core EPYC 7402P 512 GB (204 GB/s)	A100 PCIe (9.7 TFLOPS) 80 GB (**1935 GB/s**)
Grace Hopper node	72-core ARMv9a Grace 480 GB (384 GB/s)	GH200 (34 TFLOPS) 96 GB (**4000 GB/s**)

- Baseline methodのソルバー反復あたりの実行時間は、各システムのメモリバンド幅にほぼ比例
  - cuSPARSEによる疎行列ベクトル積と同等の性能(メモリバンド幅の50%程度)
  - 計算システム間で公平な比較ができていると期待
- Grace Hopper nodeにおいてはCPUメモリはGPUメモリの5倍の容量を有し、CPU-GPU間は900 GB/sの高速インターコネクトで結合
  - 提案手法により高速解析が期待できる

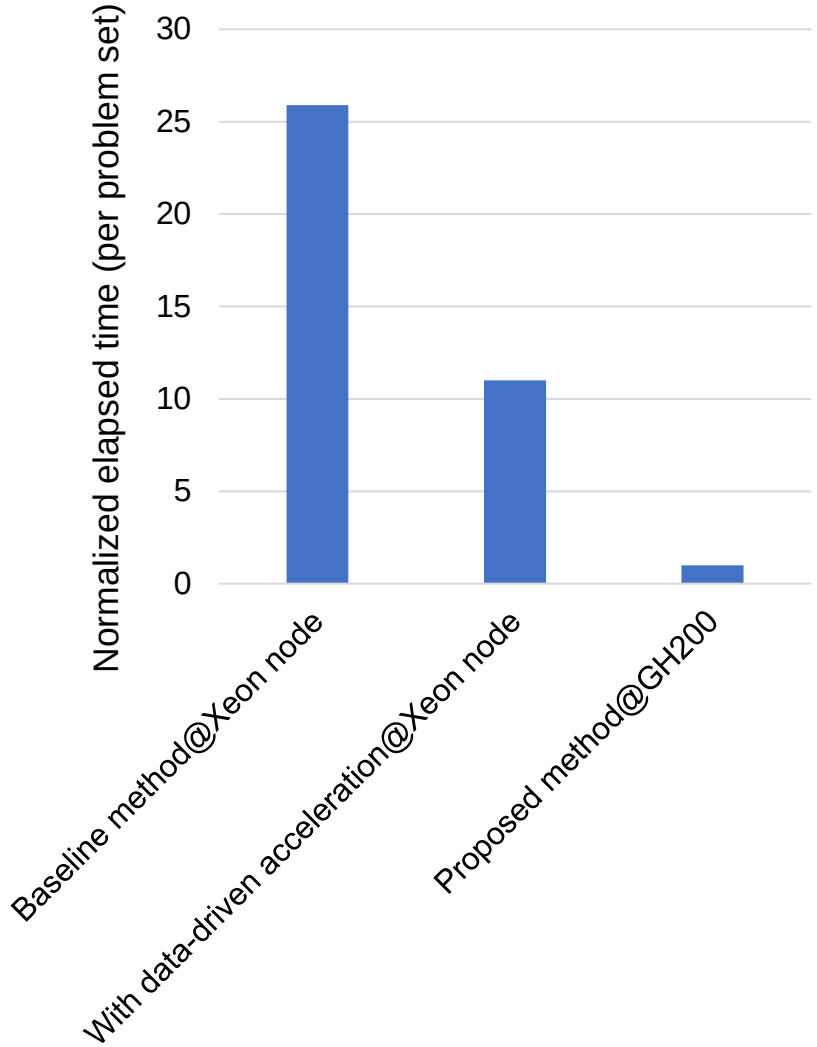
Baseline method@A100と提案手法@GH200での性能比較

- Baseline methodの性能：GPUメモリバンド幅の2.1倍の改善に伴い、GH200においてA100比で2.1倍高速化
- 提案手法@GH200はbaseline method@GH200よりさらに2.9倍高速
  - CPU計算とGPU計算はほぼオーバーラップ
  - データ駆動型手法により反復数が1/3.2になることに伴い計算も高速化
- A100比で合計6.2倍の高速化を実現



CPUシステムとの性能比較

- データ駆動型手法はCPUベースのシステムでも利用できるが、GPUと比べてCPU自体の計算性能は相対的に低い
- 提案手法@GH200 nodeにより
  - Baseline method@Xeon node比で26倍の高速化
  - データ駆動型手法で加速されたソルバー@Xeon node比でも11倍の高速化



Xeon node: 4×20-core Xeon Gold 6230 (total of 564 GB/s for 4 sockets)
GH200: 72-core CPU with 96 GB Grace Hopper GPU (4000 GB/s)

さらなる高速化@GH200

CPU Memory size (bandwidth)	GPU Memory size (bandwidth)
72-core ARMv9a Grace 480 GB (**384 GB/s**)	GH200 96 GB (**4000 GB/s**)

- 高速なSpMV手法(matrix-freeな疎行列ベクトル積手法)と複数ベクトルの同時計算と組み合わせることでさらなる高速化も可能
  - メモリ転送量・ランダムアクセスの削減により、さらに4倍の高速化
  - 実行時間はCPU単体比で86倍、GPU単体比で9.0倍に高速化
  - 使用エネルギーもCPU単体比で1/32に、GPU単体比で1/7.0に削減

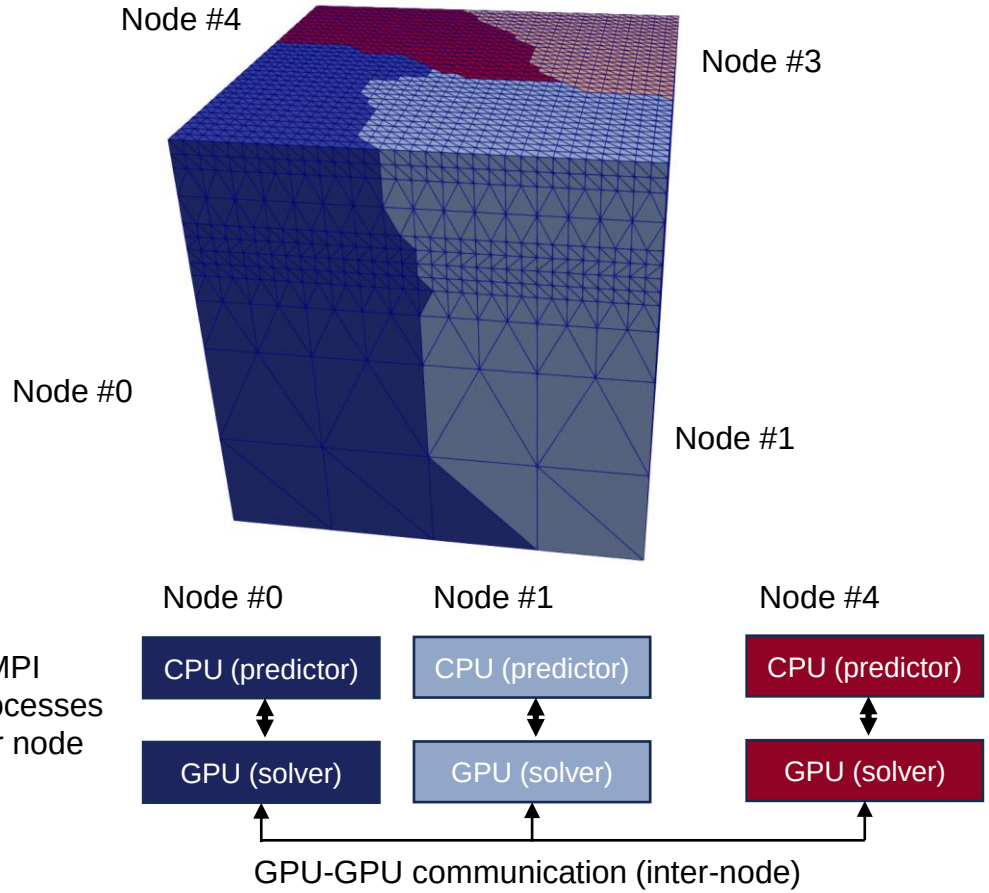
	CPU memory usage	GPU memory usage	Total elapsed time per case	Elapsed time for solver per case	Elapsed time for predictor per case	Solver iterations per time step	Module power (GPU power)	Total energy per time step per case
CPU only	56.9 GB	-	**30.4 s**	30.2 s	No predictor	152	327 W (76 W)	9944 J
GPU only	104 GB	44.9 GB	**3.05 s**	3.03 s	No predictor	152	709 W (608 W)	2163 J
Developed (CPU+GPU)	340 GB	60.5 GB	**0.352 s**	0.336 s	0.310 s	68.8	877 W (652 W)	309 J

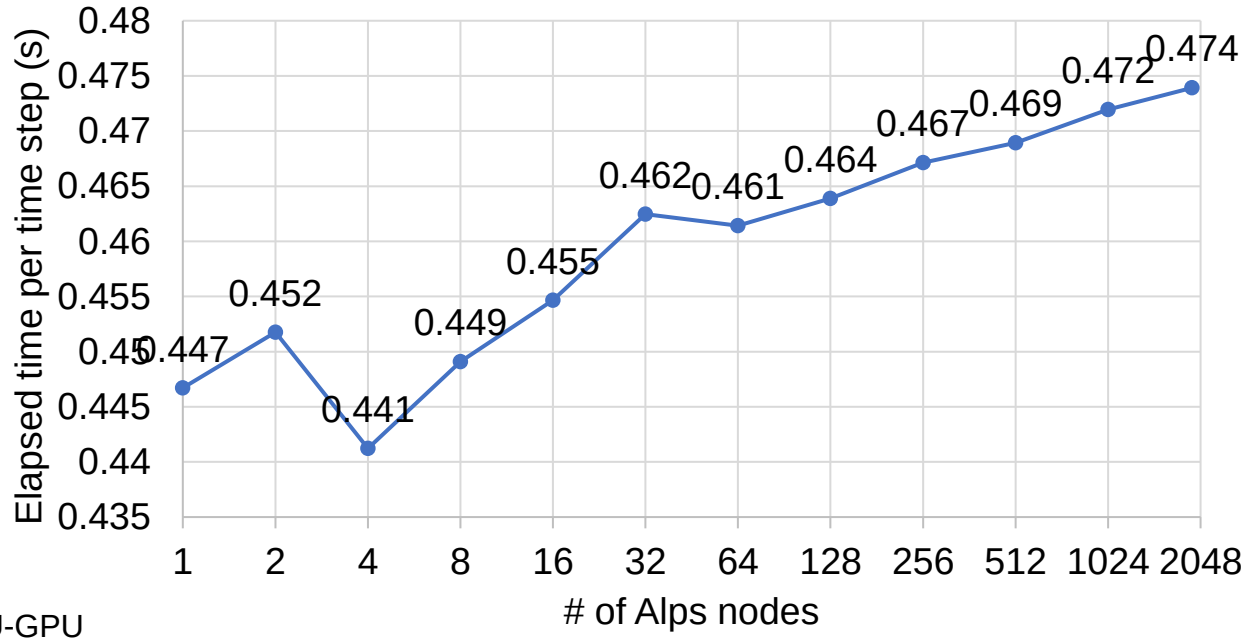
Computed on GPU　Computed on CPU

T. Ichimura, K. Fujita, M. Hori, L. Maddegedara, J. Wells, A. Gray, I. Karlin, J. Linford, **accepted for WACCPD 2024**.
preprint available: https://doi.org/10.48550/arXiv.2409.20380

Weak scaling on Alps@CSCS

- 初期解推定手法：ソルバー間のデータ同期は計算ノード内で閉じているため、大規模システムにおいて高いスケーラビリティが期待
- 1920 Alps nodes (7680 GH200 modules)において94.3% weak scalability





T. Ichimura, K. Fujita, M. Hori, L. Maddegedara, J. Wells, A. Gray, I. Karlin, J. Linford, **accepted for WACCPD 2024**. preprint available: https://doi.org/10.48550/arXiv.2409.20380

まとめ

- データ駆動型手法によりPDEに基づく時刻歴シミュレーションを高速化する手法を開発
  - シミュレーション精度を落とすことなく計算を高速化
  - CPUのみ、GPUのみ、およびCPUとGPUの強結合システムに適用可能
- 異なるタイプのアルゴリズムを組み合わせることで、heterogeneousな計算機アーキテクチャーの効率的な利用が可能となることを示した
  - データ駆動型手法により大容量CPUメモリ・線形ソルバー部で高性能GPUを活用
  - 実行時間と必要エネルギーの双方の削減につながる
- このような柔軟なアルゴリズム開発は次世代のheterogeneousシステムにおいても有用と期待