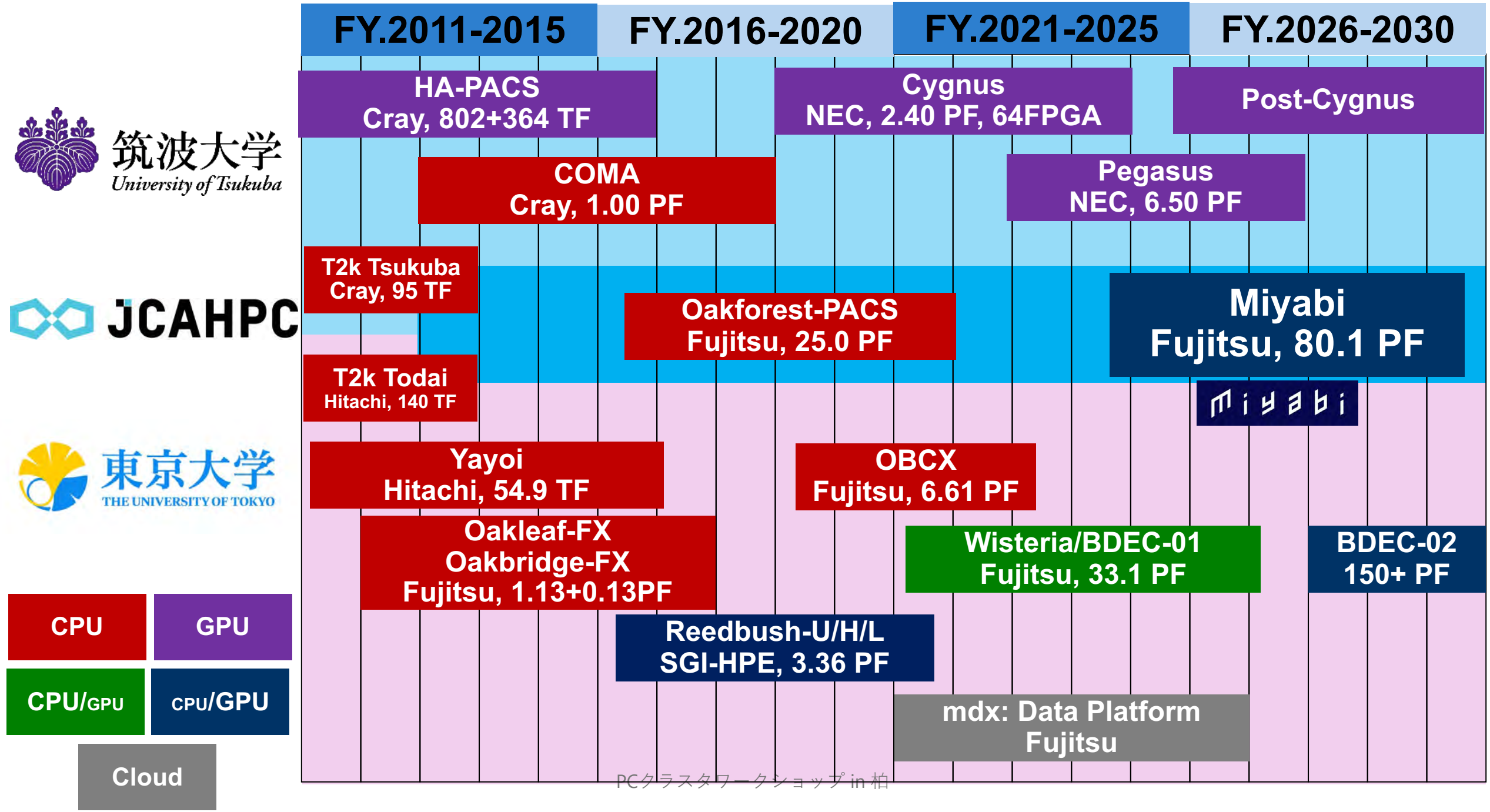
PCクラスタコンソーシアム
ワークショップ@柏
2025年6月27日(金)

JCAHPCスーパーコンピュータ Miyabi-Gにおける東京大学情報基盤センターの取り組み

塙 敏博

東京大学 情報基盤センター / JCAHPC


FY.2011-2015
FY.2016-2020
FY.2021-2025
FY.2026-2030
筑波大学
University of Tsukuba
JCAHPC
東京大学
THE UNIVERSITY OF TOKYO
HA-PACS
Cray, 802+364 TF
Cygnus
NEC, 2.40 PF, 64FPGA
Post-Cygnus
COMA
Cray, 1.00 PF
Pegasus
NEC, 6.50 PF
T2k Tsukuba
Cray, 95 TF
Oakforest-PACS
Fujitsu, 25.0 PF
Miyabi
Fujitsu, 80.1 PF
T2k Todai
Hitachi, 140 TF
Yayoi
Hitachi, 54.9 TF
OBCX
Fujitsu, 6.61 PF
Oakleaf-FX
Oakbridge-FX
Fujitsu, 1.13+0.13PF
Wisteria/BDEC-01
Fujitsu, 33.1 PF
BDEC-02
150+ PF
Reedbush-U/H/L
SGI-HPE, 3.36 PF
mdx: Data Platform
Fujitsu
CPU
GPU
CPU/GPU
CPU/GPU
Cloud

JCAHPC第二世代システム “Miyabi”

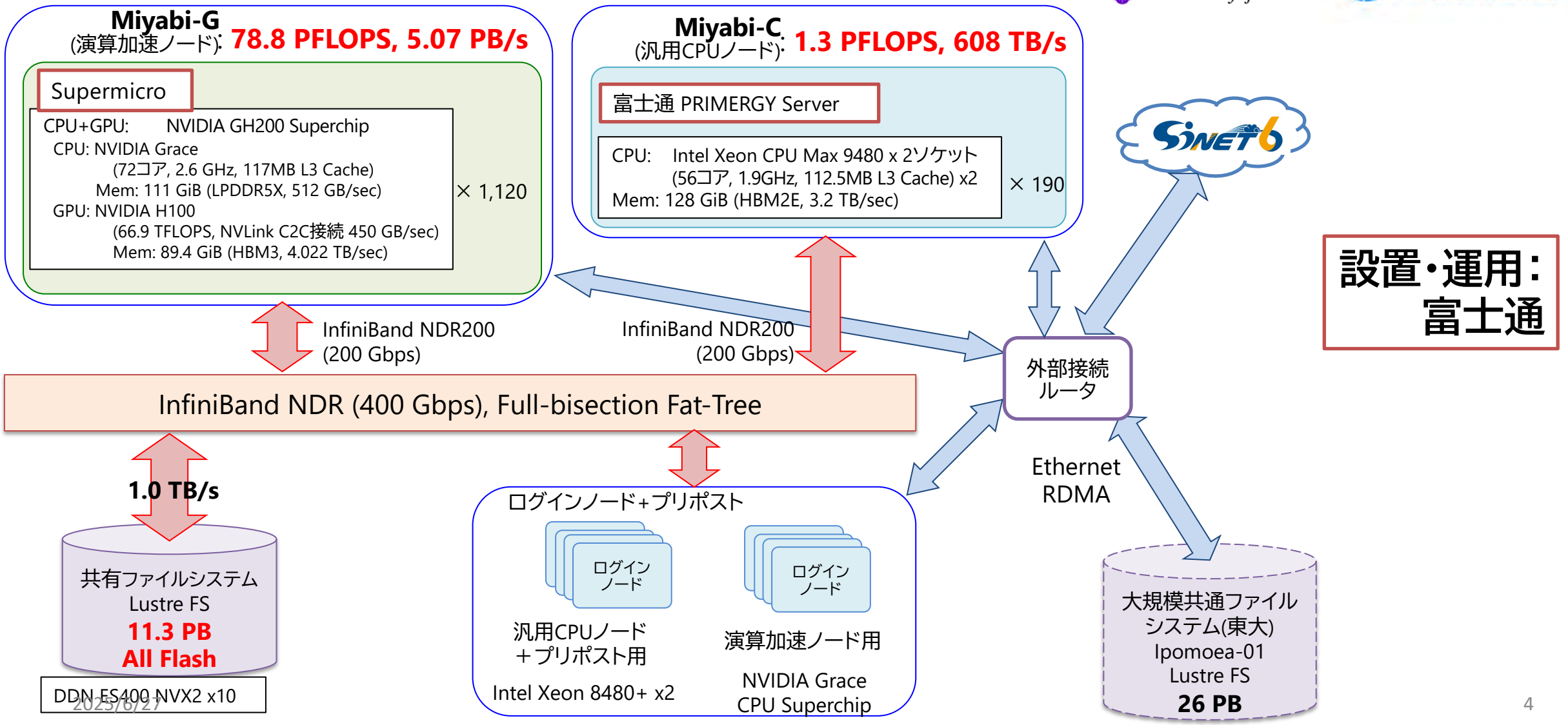
JCAHPC 筑波大学 University of Tsukuba 東京大学 THE UNIVERSITY OF TOKYO

システムの特徴

- システム全体性能の飛躍的な向上のため、演算加速装置としてGPUを主体とするシステムへ
  - 消費電力も削減 ➔ 電力当たり性能の劇的な向上
  - CPU-GPU間が高速リンクで密結合され、既存アプリのGPU化が容易に
- GPU化が困難なアプリケーションのため、汎用CPUのみの計算ノードも導入
- ストレージの高性能化に向けてAll Flashを導入
- 従来のHPCに加えて、AI for Scienceを支える計算基盤

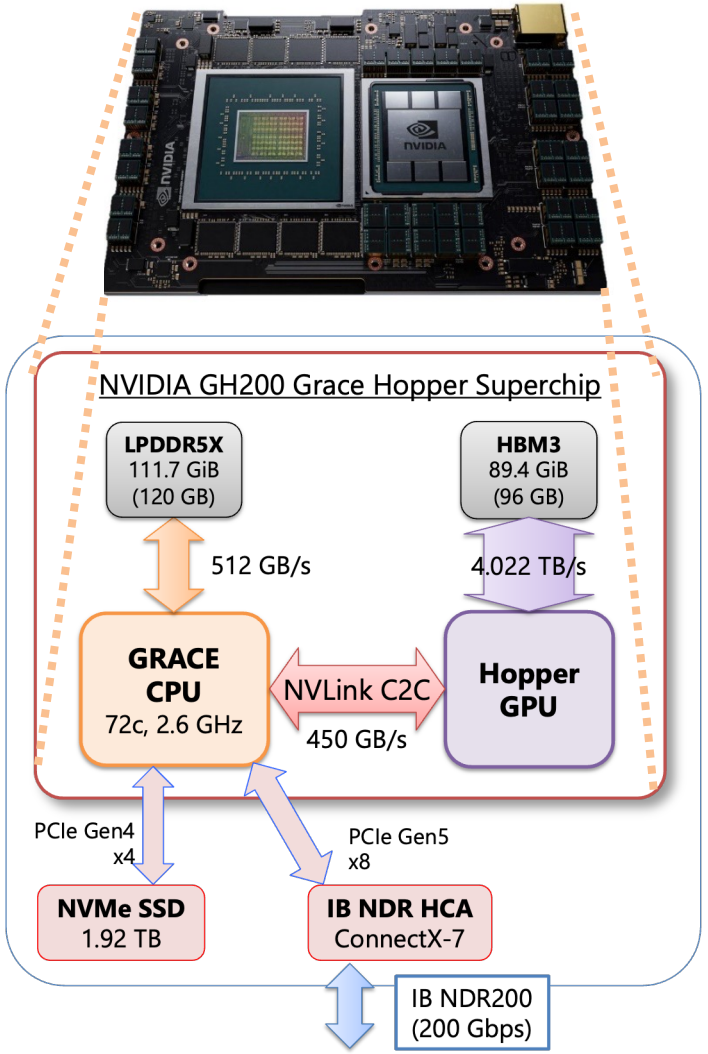
Miyabi (OFP-II)の概要 (1/2)
2025年1月運用開始

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



Miyabi(OFP-II)の概要 (2/2)

- **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)**



65th TOP500 List (June, 2025)

R_{max}: Performance of Linpack (TFLOPS)
R_{peak}: Peak Performance (TFLOPS), Power: kW

http://www.top500.org/

	Site	Computer/Year Vendor	Cores	R_{max} (PFLOPS)	R_{peak} (PFLOPS)	GFLOPS/W	Power (kW)
1	**El Capitan, 2024, USA** DOE/NNSA/LLNL	HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS	11,039,616	1,742.00 (=1.742 EF)	2,746.38 63.4 %	58.99	29,581
2	**Frontier, 2021, USA** DOE/SC/Oak Ridge National Laboratory	HPE Cray EX235a, AMD Optimized 3rd Gen. EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11	9,066,176	1.353.00	2,055.72 65.8 %	54.98	24,607
3	**Aurora, 2023, USA** DOE/SC/Argonne National Laboratory	HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel	9,264,128	1,012.00	1,980.01 51.1 %	26.15	38,698
4	**JUPITER Booster, 2025, USA** EuroHPC/FZJ	EVIDEN, BullSequana XH3000, GH Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, RedHat Enterprise Linux	4,801,344	794,40	930.00 85.3 %	60.62	13,088
5	**Eagle, 2023, USA** Microsoft	Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR	2,073,600	561.20	846.84 66.3 %		
6	**HPC 6, 2024, Italy** Eni S.p.A.	HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, RHEL 8.9	3,143,520	477.90	606.97 66.3 %	56.48	8,461
7	**Fugaku, 2020, Japan** R-CCS, RIKEN	Fujitsu PRIMEHPC FX1000, Fujitsu A64FX 48C 2.2GHz, Tofu-D	7,630,848	442.01	537.21 82.3 %	14.78	29,899
8	**Alps, 2024, Switzerland** Swiss Natl. SC Centre (CSCS)	HPE Cray EX254n, NVIDIA Grace 72C 3.1GHz, NVIDIA GH200 Superchip, Slingshot-11	2,121,600	434.90	574.84 75.7 %	61.05	7,124
9	**LUMI, 2023, Finland** EuroHPC/CSC	HPE Cray EX235a, AMD Optimized 3rd Gen. EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11	2,752,704	379.70	531.51 71.4 %	53.43	7,107
10	**Leonard, 2023, Italy** EuroHPC/Cineca	EVIDEN, BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64GB, Quad-rail NVIDIA HDR100	1,824,768	241.20	181.49 78.7 %	32.19	7,494
15	**ABCI 3.0, 2025, Japan** AIST	HPE Cray XD670, Xeon Platinum 8558 48C 2.1GHz, NVIDIA H200 SXM5 141 GB, Infiniband NDR200, Rocky Linux 9	479,232	145.10	130.44 80.0 %	59.29	3,596
22	**CHIE-3, 2024, Japan** SoftBank, Corp.	NVIDIA DGX H100, Xeon Platinum 8480C 56C 2GHz, NVIDIA H100, Infiniband NDR400, Ubuntu 22.04.4 LTS	297,840	91.94	138.32 66.5 %		
24	**CHIE-2, 2024, Japan** SoftBank, Corp.	NVIDIA DGX H100, Xeon Platinum 8480C 56C 2GHz, NVIDIA H100, Infiniband NDR400, Ubuntu 22.04.4 LTS	297,840	89.78	138.32 64.9 %		
27	**ABCI-Q, 2025, Japan** AIST	Fujitsu, Supermicro SYS-221GE-TNHT-LCC, Intel Xeon Platinum 8558 48C 2.1GHz, NVIDIA H100 SXM5 80GB, Infiniband NDR, Rocky Linux 9.4	315,210	74.58	99.35 75.1 %	40.67	1,834
36	**FPT, 2025, JAPAN** FPT AI Factory	HPE, HGX H200, Xeon Platinum 8558 48C 2.1GHz, NVIDIA H200 SXM5 141 GB, Infiniband NDR400, Ubuntu 22.04.5 LTS	146,304	49.85	67.44 73.9 %		
37	**Miyabi-G, 2024, Japan** JCAHPC	Fujitsu, Supermicro ARS 111GL DNHR LCC, Grace Hopper Superchip 72C 3GHz, Infiniband NDR200, Rocky Linux	80,640	46.80	72.80 64.3 %	47.59	983
46	**TSUBAME 4.0, 2024, Japan** Institute of Science Tokyo	HPE Cray XD665, AMD EPYC 9654 96C 2.4GHz, NVIDIA H100 SXM5 94 GB, Infiniband NDR200	172,800	39.62	61.60 64.3 %	48.55	816
73	**Wisteria/BDEC-01 (Odyssey), 2021, Japan** U.Tokyo	Fujitsu PRIMEHPC FX1000, A64FX 48C 2.2GHz, Tofu D	368,640	22.12	25.95 85.2 %	15.07	1,468

Green 500 Ranking (Jun. 2025)

http://www.top500.org/

	TOP 500 Rank	System	Accelerator	Cores	HPL Rmax (Pflop/s)	Power (kW)	GFLOPS/W	Level
1	259	JEDI, EuroHPC/Julich, Germany	NVIDIA GH200	19,584	4.50	67	72.733	1
2	148	ROMEO-2025, ROMEO HPC Center - Champagne-Ardenne, France	NVIDIA GH200	47,328	9.86	160	70.912	1
3	484	Adastra2, GENCI-CINES, France	AMD Instrinct MI300A	16,128	2.53	37	69.098	1
4	183	Isambard-AI phase1, U. Bristol, UK	NVIDIA GH200	34,272	7.42	117	68.835	1
5	255	Otus (GPU Only), U. Paderborn, Germany	NVIDIA H100 80GB	19,440	4.66		68.177	1
6	66	Capella, TU Dresden ZIH, Germany	NVIDIA H100 94GB	85,248	24.06	445	68.053	3
7	304	SSC-24 Energy Module, Samsung, Korea	NVIDIA H100 80GB	11,200	3.82	69	67.251	2
8	85	Helios GPU, Cyfronet, Poland	NVIDIA GH200	89,760	19.14	317	66.948	2
9	399	AMD Ouranos, Atos, France	AMD Instinct MI300A	16,632	2.99	48	66.464	1
10	412	Henri, Flatiron Institute, USA	NVIDIA H100 80GB	8,288	2.88	44	65.396	?
19	8	ALPS, CSCS, Switzerland	NVIDIA GH200	2,121,600	434.90	7,124	61.047	3
42	46	TSUBAME 4.0, Science Tokyo	NVIDIA H100 94GB SXM5	172,800	39.62	816	48.565	3
46	37	**Miyabi-G, JCAHPC**	**NVIDIA GH200**	**221,952**	**46.80**	**983**	**47.588**	**3**
63	267	Pegasus, University of Tsukuba, Japan	NVIDIA H100 80GB	27,000	4.34	130	41.123	2
89	262	Wisteria/BDEC-01 (Aquarius), The University of Tokyo, Japan	NVIDIA A100 40GB	42,120	4.425	183	24.06	2

素朴な疑問：GH200のGFLOPS/W

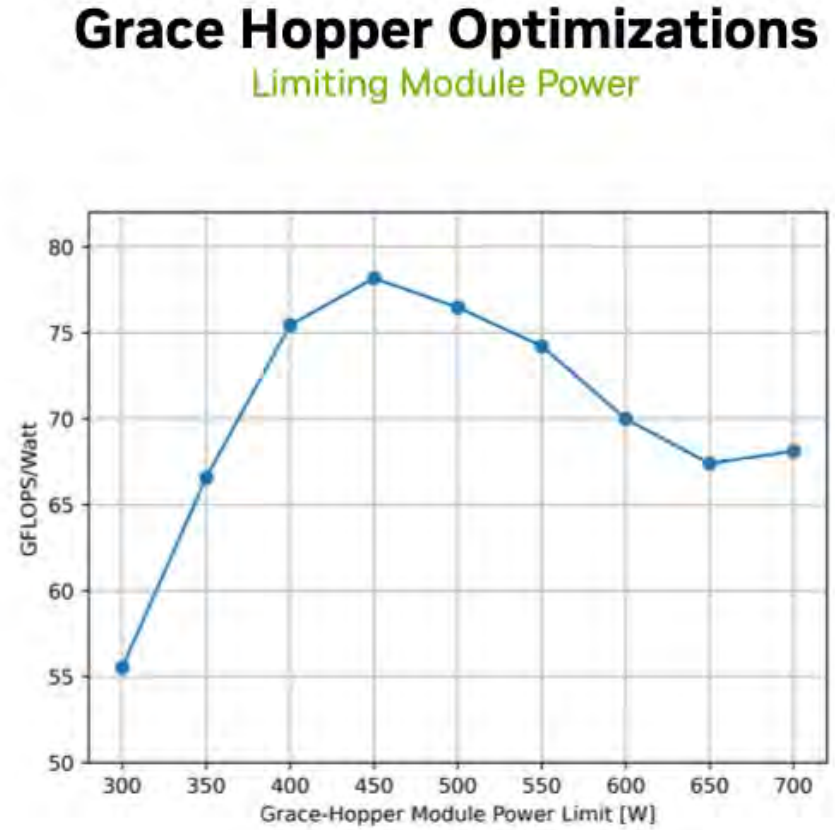
- 同じGH200なのになぜここまでGFLOPS/Wに差があるのか

System	GFLOPS/W
JEDI@Julich	72.733
ROMEO	70.912
Isambard-AI	68.835
Helios@Poland	66.948
ALPS@CSCS	61.047
Venado@LANL	59.287
CEA-HE	52.165
Miyabi-G@JCAHPC	**47.588**

ノード構成の違い: CG4 vs CG1

- GH200のTDP (Thermal Design Power): **1000 W**
- 4ソケット/ノード=CG4では電力供給等の制約からソケットごとに電力キャップ
  ➔ ALPS (HPE Cray-EX254n): **630 W / ソケット**
    - 写真: 2ノード分, 8ソケット
      https://www.servethehome.com/8x-nvidia-grace-hopper-superchips-arm-in-a-blade-hpe-cray-ex254n-at-gtc-2024/
  - 他のCrayシステム, Bull/Atos/Evidenも同様だと思われる
  - 前掲のシステムはMiyabi以外全てCG4
- CG1であるMiyabi-Gでは安定度も考慮して1000➔**900 W**に設定
  - HPLは1000Wで測定

実行時の電力制約を厳しくする方がHPL効率は上がるが，，



Patrick Atkinson and Alan Gray, S72544: Optimizing HPC and AI for Energy and Performance, GTC 2025

しかし，Miyabiの運用では敢えて450W ~ 500Wのような制約をかけるわけにはいかない!!

- The energy optimization that gives the most gain is limiting the total Grace Hopper module power.
- If we have a maximum module power of 700W, we achieve 68 GFLOPS/Watt
- Limiting the module power to 500W, we achieve 78 GFLOPS/Watt

本研究の目的

- GH200における電力効率向上
- ただしユーザージョブの実行時間が延びるのは避けたい

➔ 両立させる方法はないか

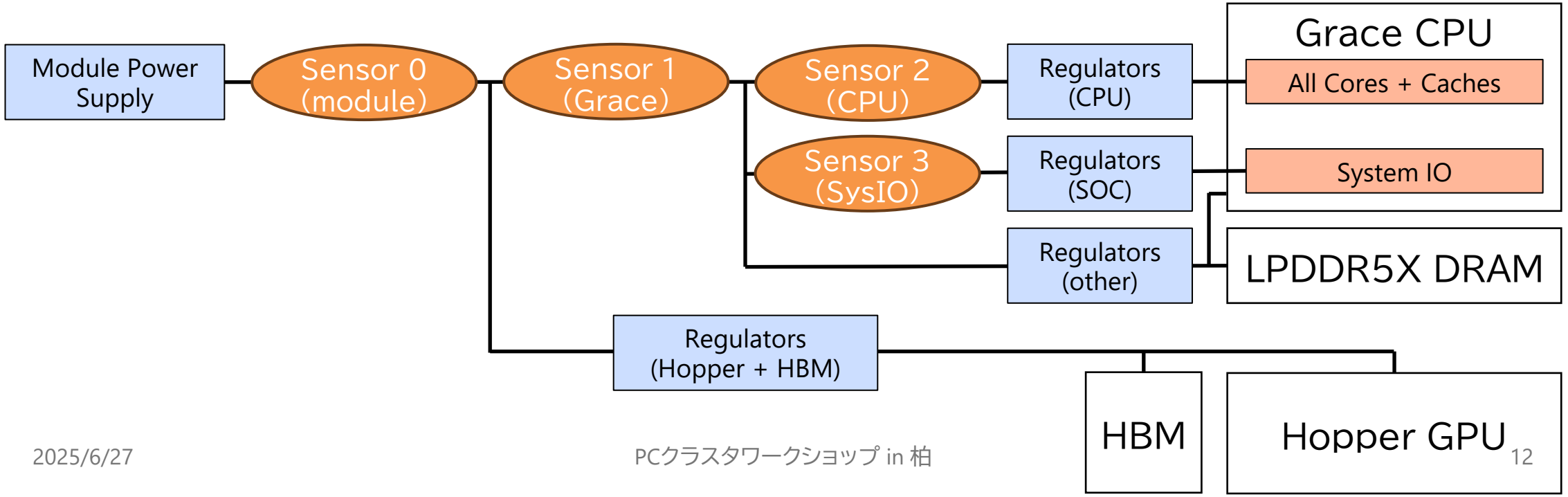
- CPU Frequency Governorがある程度使えるのでは？

- 関連研究
  - Julita Corbalan, “EAR: Energy management framework for HPC”
    - Barcelona Supercomputer Center, LRZ SuperMUC
  - Anna Yue et al., “EVeREST: An Effective and Versatile Runtime Energy Saving Tool for GPUs,” PPoPP’25

  ➔ 今後試してみる予定

GH200における電力測定ポイント

- Armコアにおけるセンサードライバにより読み取り
- モジュール電力: `/sys/class/hwmon/hwmon1/device/power1_average`
- Grace電力: `/sys/class/hwmon/hwmon2/device/power1_average`
  - 同様に `hwmon3` ➔ `Sensor2`, `hwmon4` ➔ `Sensor3`



GH200のGPUにおける電力設定＋測定(?)

```
$ nvidia-smi -q -d POWER

    Driver Version                          : 550.163.01
    CUDA Version                            : 12.4

    GPU 00000009:01:00.0
        GPU Power Readings
            Average Power Draw              : 66.99 W
            Instantaneous Power Draw        : 120.13 W
            Current Power Limit             : 700.00 W
            Requested Power Limit           : 700.00 W
            Default Power Limit             : 900.00 W
            Min Power Limit                 : 100.00 W
            Max Power Limit                 : 900.00 W
        Power Samples
            Duration                        : 2.36 sec
            Number of Samples               : 119
            Max                             : 68.06 W
            Min                             : 65.84 W
            Avg                             : 66.89 W
        GPU Memory Power Readings
            Average Power Draw              : 12.94 W
            Instantaneous Power Draw        : N/A
        Module Power Readings
            Average Power Draw              : 120.40 W
            Instantaneous Power Draw        : 120.13 W
            Current Power Limit             : 900.00 W
            Requested Power Limit           : 900.00 W
            Default Power Limit             : 1000.00 W
            Min Power Limit                 : 200.00 W
            Max Power Limit                 : 1000.00 W
```

GPU電力制限値

モジュール電力制限値

GH200における電力制限

- モジュール電力制限
  - 900 Wに制限：`nvidia-smi -pl 900 -sc 1`
  - CPUとGPU間でバジェット(この場合900W)を適宜分け合う
  - Miyabi-Gでは 900 W, defaultは 1000 W
- GPU電力制限
  - 700 Wに制限：`nvidia-smi -pl 700 -sc 0`
  - Miyabi-Gでは 700 W, defaultは 900 W
- CPU電力制限
  - 250 Wに制限:
    `echo 250000000 > /sys/class/hwmon/hwmon2/device/power1_cap`
  - Miyabi-Gでは 300 W, defaultも 300 W

CPU Frequency Governor

- モジュール電力のバジェットが決まっているとき, Grace CPUの電力によってHopper GPUの電力が制限される
  ➔ CPU実行の際に省電力にすることが重要

Governorの種類	
performance	常に許容される最高周波数を使用 (Miyabi-Gの設定)
ondemand	現在のシステム負荷、直近のCPU使用率から、CPU の周波数を設定
conservative	ondemand に似ているが、CPU 負荷が発生した瞬間に最高周波数に切り替えるのではなく、CPU 速度を徐々に増加および減少
schedutil	kernel のスケジューラと密接に関連して周波数の切り替え 負荷も監視しているが、タスクの種類によっては常に最高周波数で実行
userspace	管理者権限のないユーザでも周波数の切り替えを可能にする。ここでは扱わない。
powersave	実際には実用にならないほどの低い周波数にセットされるため、ここで は扱わない

Miyabi-GにおけるHPL実行 (GPU)

- `HPL_OOC_MODE=1, N=149504`
- 49.5 TFLOPS
- 34.33 kJ (モジュール)

➔ 平均電力 763.6 W

➔ 64.85 GFLOPS/W

HPL @ GH200, performance (900W-700W)

- HPL_OOC_MODE=1, N=141312
- 50.58 TFLOPS
- 27.73 kJ

➔ 平均電力 745.1 W

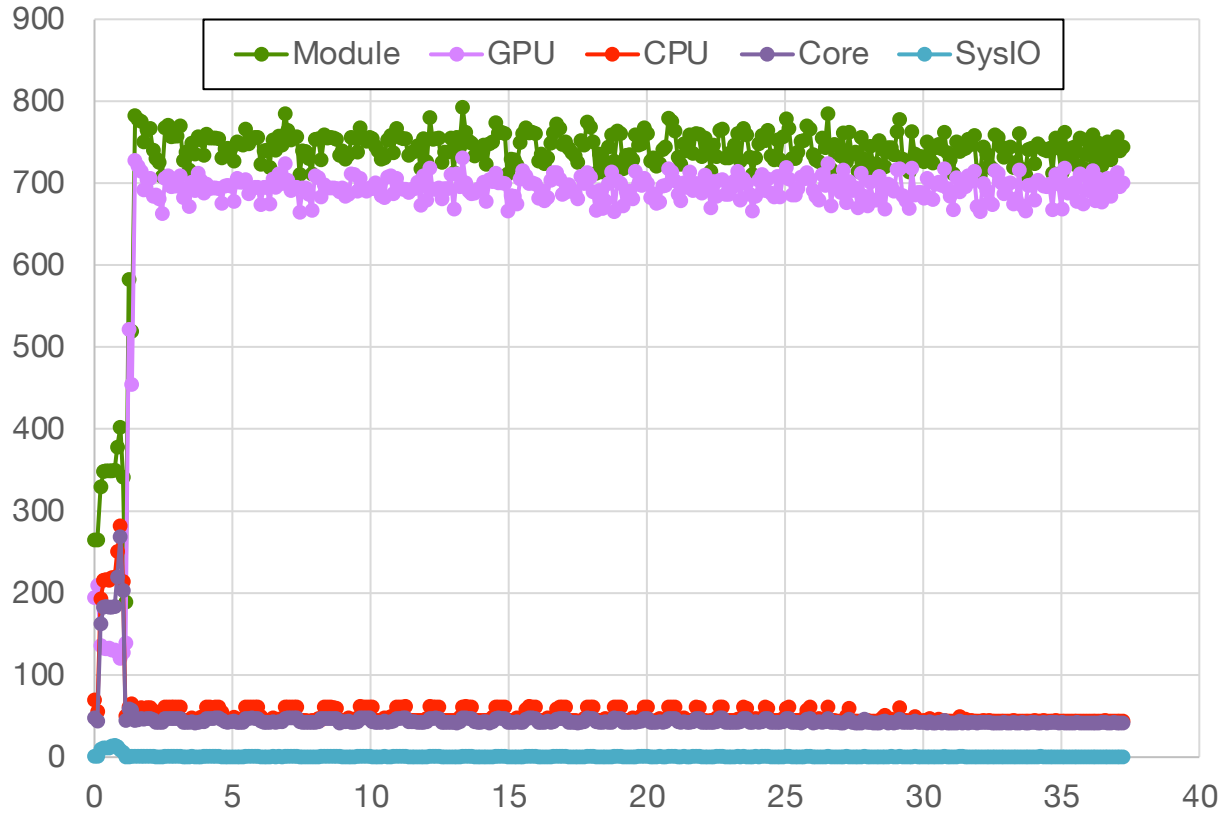
➔ 67.89 GF/W

HPL @ GH200, ondemand

- HPL_OOC_MODE=1, N=141312
- 50.57 TFLOPS
- 27.21 kJ

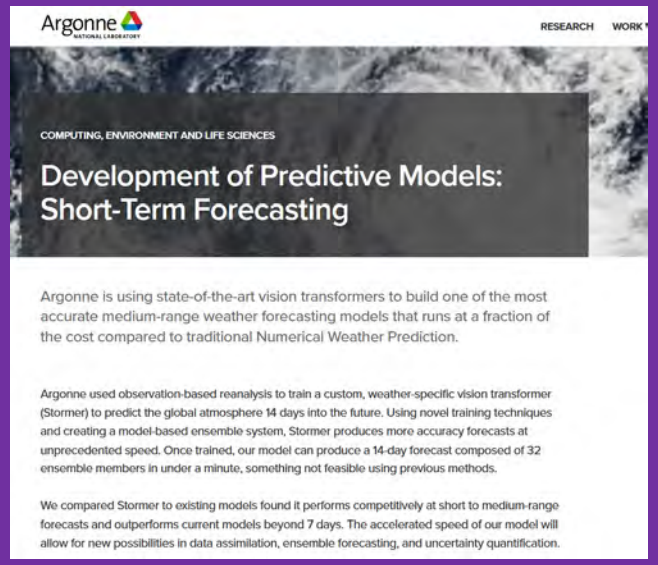
➔ 平均電力 731.1 W

➔ 69.17 GF/W



AI for Science・Miyabiな計算科学

- **生成AIによる科学の革新（Innovation in Scientific Research）**
  - **従来のAI for Science（AIの計算科学への応用）：データ駆動型アプローチ⇒既に実施**
- 米国アルゴンヌ国立研究所の “Stormer”：全球天気予報
  - 「粗いメッシュ，短期予報」という制限のもと，大規模スパコンをGPU数枚で代替可能
  - https://www.anl.gov/cels/development-of-predictive-models-shortterm-forecasting
- 利用者によるこの種の「AI＋CSE」の試みをサポートして行きたい



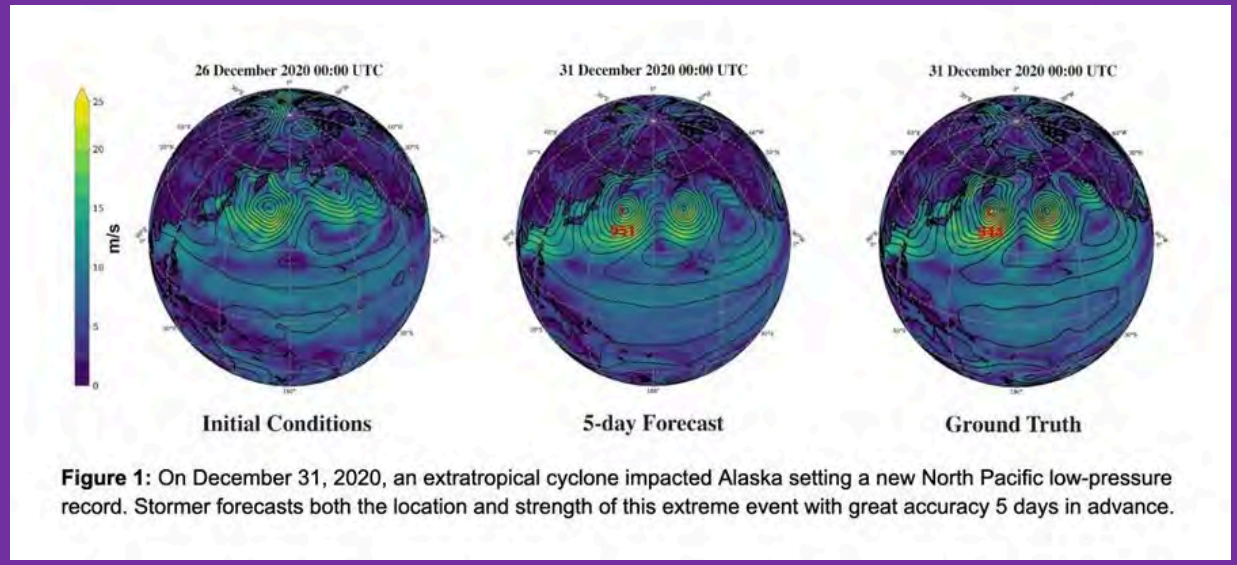
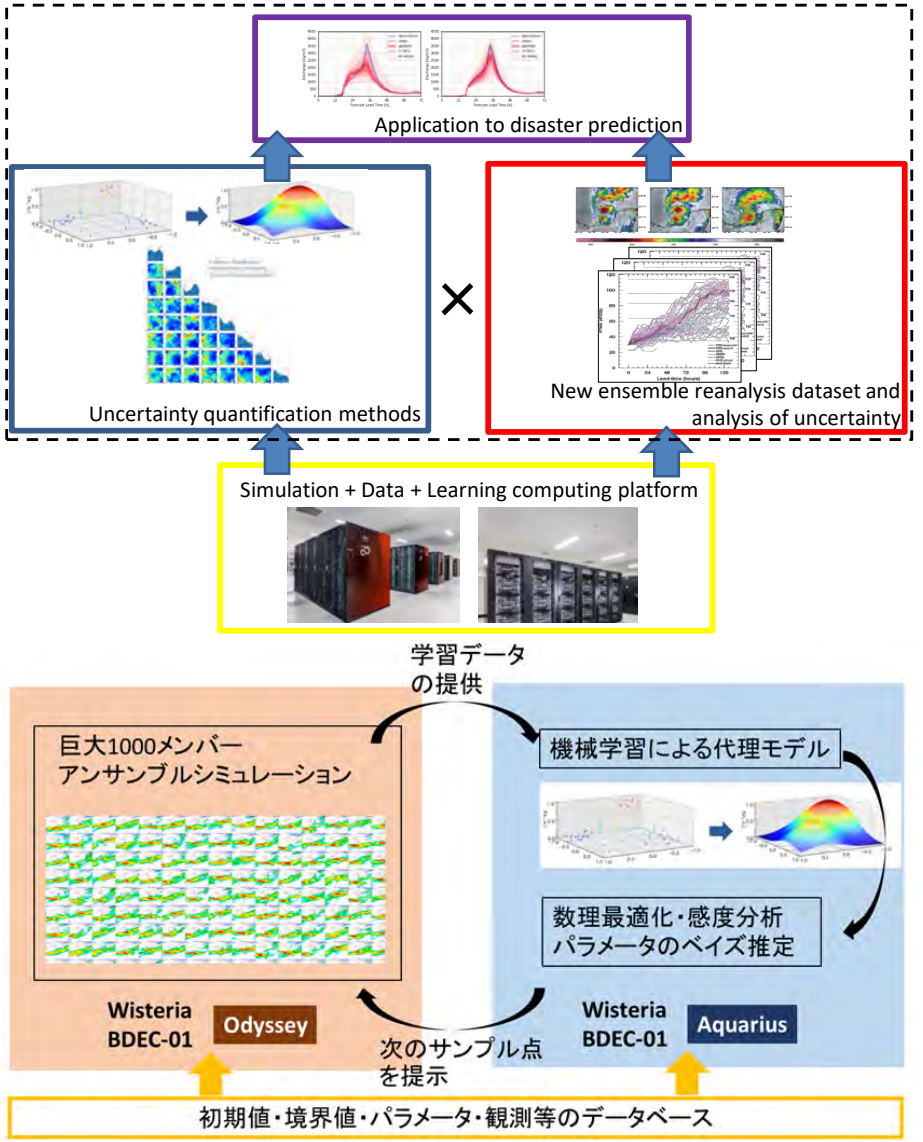


Figure 1: On December 31, 2020, an extratropical cyclone impacted Alaska setting a new North Pacific low-pressure record. Stormer forecasts both the location and strength of this extreme event with great accuracy 5 days in advance.

極端気象現象予測における不確実性の起源の解明

澤田洋平准教授（東大・工学系）

- 東大AI-for-Science（2020-2021年度）
- JHPCN共同研究課題（2022-，2025も応募）
- 極端（激甚）気象シミュレーションの不確実性定量化手法の確立
  - Uncertainty Quantification of Extreme Weather Prediction
- 不確実性情報を付加した極端気象データセットの整備と解析
  - 現在はWisteria/BDEC-01上で「データ駆動型アプローチ」を適用
  - 将来的には生成AIを適用
    - 基盤モデル生成フレームワーク向けベンチマーク（アプリケーション）



Solomon: OpenACCとOpenMPを自由自在に切り替え

資料は三木洋平准教授より

指示文ベースでGPU化したい場合の選択肢

- OpenACC
  - GPU向けのメジャーな指示文
  - PGIがNVIDIAに買収された結果, NVIDIA色が強くなってしまった
  - AMD, Intelは(きっと)サポートしない
    - HPE Crayコンパイラであれば, AMD GPU向けのOpenACCもサポート
    - IntelはOpenACCからOpenMP targetへの変換ツールを開発中
    - ➔GPUベンダー(AMD, Intel)による直接支援が受けられない
- OpenMPのtarget指示文
  - OpenMP 4.0以降でアクセラレータへのオフロードがサポート
  - OpenMP 5.0で loop 指示節が追加, OpenACC的実装も可能に
  - NVIDIA, AMD, Intel 全てのGPU向けにサポートされる
  - 現時点ではOpenACCの全ての機能に対応できていない
    - 非同期実行の(細やかな)制御など
- 実装時には, 使う指示文についても選択する必要がある

マクロを用いた指示文のブラックボックス化

- Solomon(Simple Off-LOading Macros Orchestrating multiple Notations)を実装
  - Miki & Hanawa (2024, IEEE Access)
- バックエンドで OpenACC or OpenMP target に展開
  - Fallback mode (マルチコアCPU向けのOpenMPに展開)も実装済み
- ユーザ的にはオーバーオールのフラグ制御だけで OpenACC or OpenMP target の切り替えが可能
  - NVIDIA GPU 上では OpenACC で,
    AMD/Intel GPU 上では OpenMP target で動かす, ということが可能になる
  - 最適化レベルを揃えた上で OpenACC と OpenMP target の性能比較
- コンパイラではないのでベンダー製のコンパイラ性能をそのまま利用できる
  - (GPU向けプログラミングに詳しい人は)HIP の指示文版とイメージすると良い
  - 自作コンパイラの場合には, 最新機能への追随のためのコストが継続的に生じる
- 開発者が更新をさぼっても, 自分でマクロを付け足すことも簡単
  - 新コンパイラ/新指示文の実装であれば, 一般ユーザはほぼ手出しできない

Solomonを用いた実装例

- OpenACCとOpenMP両方に対応した指示文の追加例(右側)
  - 場合分け(ACC for GPU, OMP for CPUなど)がかなり煩雑
  - `$ nvc++ -acc=multicore -mp=gpu …` も(見かけないが)実は可能
- 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
  - https://github.com/ymiki-repo/solomon で公開
  - Miki & Hanawa (2024, IEEE Access)
  - 手品の種: `_Pragma()`形式で指示文を記述
  - 対応しているバックエンド:
    - OpenACC, OpenMP target, OpenMP
- どちらの手法でコードを実装しますか?
  - 通常の(煩雑な)実装方法➡️
  - ⬇️ Solomon を用いて簡易化した手法

```
OFFLOAD(AS_INDEPENDENT, NUM_THREADS(NTHREADS))
for (int32_t i = 0; i < N; i++) {
```

```
#ifdef OFFLOAD_BY_OPENACC
#pragma acc kernels vector_length(NTHREADS)
#pragma acc loop independent
#endif  // OFFLOAD_BY_OPENACC
#ifdef OFFLOAD_BY_OPENMP_TARGET
#ifdef OFFLOAD_BY_OPENMP_TARGET_LOOP
#pragma omp target teams loop
thread_limit(NTHREADS)
#else  // OFFLOAD_BY_OPENMP_TARGET_LOOP
#pragma omp target teams distribute parallel
for simd thread_limit(NTHREADS)
#endif  // OFFLOAD_BY_OPENMP_TARGET_LOOP
#endif  // OFFLOAD_BY_OPENMP_TARGET
for (int32_t i = 0; i < N; i++) {
```

小まとめ

- GPU向け指示文は複数あり, ユーザーはどれか1つを選択する必要がある（諸悪の根源は各社の政治的な思惑？）
  - OpenACC: 機能・資料が充実しているが, ほぼNVIDIA GPU向け
  - OpenMP target: NVIDIA/AMD/Intel 全社に対応, 機能はキャッチアップ中
- GPU向け指示文統合マクロSolomonを開発（Miki & Hanawa 2024）
  - Simple Off-LOading Macros Orchestrating multiple Notations
  - プリプロセッサマクロ経由で指示文を記載するためのマクロ集
  - NVIDIA GPU上ではOpenACCを, AMD/Intel GPU上ではOpenMP target を選択, ということができる
  - 簡易記法, OpenACC的記法, OpenMP的記法があるため, 学習コストを低減
  - GPU提供ベンダー製のコンパイラをそのまま使える
  - OpenACC と OpenMP target の性能比較も簡単にできる
- GitHub で公開: https://github.com/ymiki-repo/solomon
  - 今は C/C++ のみ対応. Fortran向けインタフェースを作ってもらう計画

まとめ

- Miyabi (OFP-II) 運用開始: 2025年1月、導入、運用：富士通
  - 設置場所：東京大学 柏キャンパス（OFPと同じ部屋、ずっと小さい）
  - Miyabiの合計性能 **80.1 PFLOPS**：OFP (25 PFLOPS)の**約3.2倍**の性能
    - Miyabi-G: 演算加速ノード, NVIDIA GH200 Superchip
  - GH200を採用する国内初のオープンシステム、世界的に見ても特徴あるシステム：CG1+InfiniBand NDR
- 電力最適化：CG1ならではの最適設定を確認中
  - CPU Frequency governorによる制御、ondemand + パラメータ変更が良さそう
  - 複数ノードでの検証
  - Green500のリベンジ
    - 特殊な条件設定をせずに良い性能が得られるように
- AI for Scienceの推進
- GPU移行のサポート、Solomonの開発