

# 東北大学サイバーサイエンスセンター講習会 はじめてのスパコン

～ スーパーコンピュータAOBAの紹介と利用法入門 ～



2025年5月23日

東北大学 サイバーサイエンスセンター  
東北大学 情報部デジタルサービス支援課

- スーパーコンピュータAOBAの紹介
- スーパーコンピュータとは？
- サイバーサイエンスセンターでの応用例
- 計算機の種類
- AOBAシステムの特徴
- AOBA-S・AOBA-A・AOBA-Bのどれを使うか？
- 利用可能なアプリケーション（インストール済アプリ）
- 利用者向けウェブサイトとポータルサイト
- 負担金制度と利用申請方法
- プログラム開発の流れ
- ジョブの実行方法
- **【実習】** コンパイル・リクエスト投入・結果確認
- プログラムのコンパイル方法
- 利用者講習会・利用相談・高速化支援

# スーパーコンピュータAOBAの紹介

- サブシステム **AOBA-S**

SX-Aurora TSUBASA Type 30A

- AOBA-S用ストレージシステム

Lustre 5.6PB

- サブシステム **AOBA-A**

SX-Aurora TSUBASA Type 20B

- サブシステム **AOBA-B**

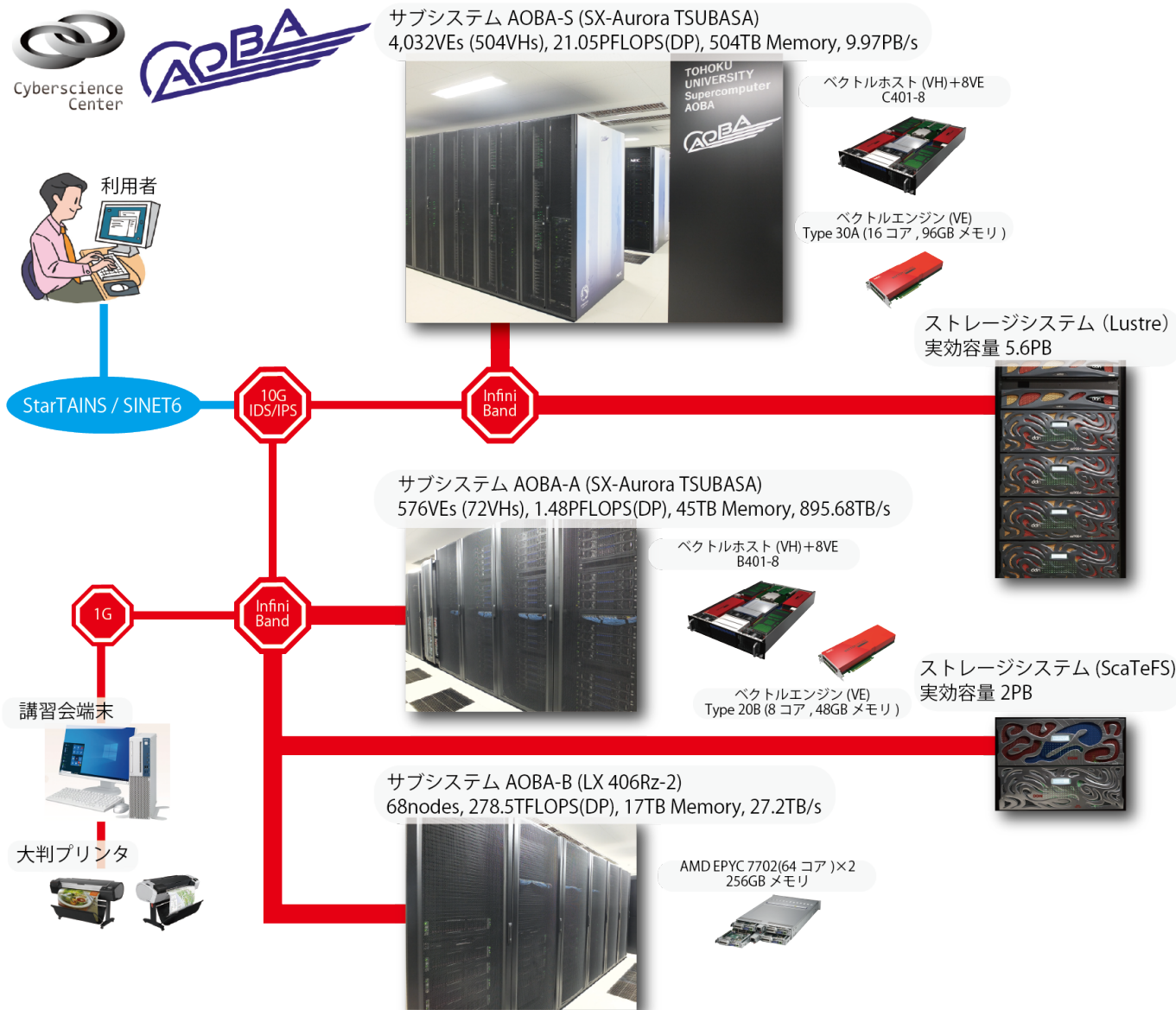
LX 406Rz-2 AMD EPYC 7702

- AOBA-A,B用ストレージシステム

ScaTeFS 2PB

- センター内施設

講習会端末、大判プリンタ



- スーパーコンピュータ（Supercomputer、略称：スパコン）とは、科学技術計算を主目的とする大規模なコンピュータのこと。
- HPC（High Performance Computer / Computing）とも呼ばれる。

# スーパーコンピュータとは？ TOP 500

- top500 (www.top500.org)
- 世界中の高性能なコンピュータのランキング
- 年2回（6月と11月）発表
- HPLベンチマークによるランキング
- 実アプリケーションとの性能乖離も指摘される

## Flop/s (Floating point number Operations Per Second)

- 1秒間に浮動小数点演算が何回できるかを表す性能指標
- CPUの性能を表す際に用いられる

## TOP500 LIST - NOVEMBER 2024

$R_{max}$  and  $R_{peak}$  values are in PFlop/s. For more details about other fields, check the TOP500 description.

$R_{peak}$  values are calculated using the advertised clock rate of the CPU. For the efficiency of the systems you should take into account the Turbo CPU clock rate where it applies.

← 1-100 101-200 201-300 301-400 401-500 →

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	<b>El Capitan</b> - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,039,616	1,742.00	2,746.38	29,581
2	<b>Frontier</b> - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607
3	<b>Aurora</b> - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
4	<b>Eagle</b> - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84	
5	<b>HPC6</b> - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, RHEL 8.9, HPE Eni S.p.A. Italy	3,143,520	477.90	606.97	8,461
6	<b>Supercomputer Fugaku</b> - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899



# スーパーコンピュータとは？ HPCGベンチマーク

- HPCGベンチマーク (<http://hpcg-benchmark.org/>)
- 実アプリケーションの挙動に近い指標
- 密行列、疎行列ベクトル計算をしており、実アプリケーションに近い多様な計算が含まれる

## HPCG LIST - NOVEMBER 2024

$R_{\max}$  and  $R_{\text{peak}}$  values are in PFlop/s. For more details about other fields, check the TOP500 description.

$R_{\text{peak}}$  values are calculated using the advertised clock rate of the CPU. For the efficiency of the systems you should take into account the Turbo CPU clock rate where it applies.

← 1-100 101-124 →

Rank	TOP500 Rank	System	Cores	Rmax (PFlop/s)	HPCG (TFlop/s)
1	6	<b>Supercomputer Fugaku</b> - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	16004.50
2	2	<b>Frontier</b> - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	14054.00
3	3	<b>Aurora</b> - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	5612.60

- Green500(<http://www.green500.org/>)
- 電力効率の良い高性能計算機のランキング
- HPLベンチマークによる演算性能をシステム電力で除した、消費電力1Wあたりの性能

## GREEN500 LIST - NOVEMBER 2024

$R_{max}$  and  $R_{peak}$  values are in PFlop/s. For more details about other fields, check the TOP500 description.

$R_{peak}$  values are calculated using the advertised clock rate of the CPU. For the efficiency of the systems you should take into account the Turbo CPU clock rate where it applies.

Green500 Data

← 1-100 101-200 201-300 301-400 401-500 →

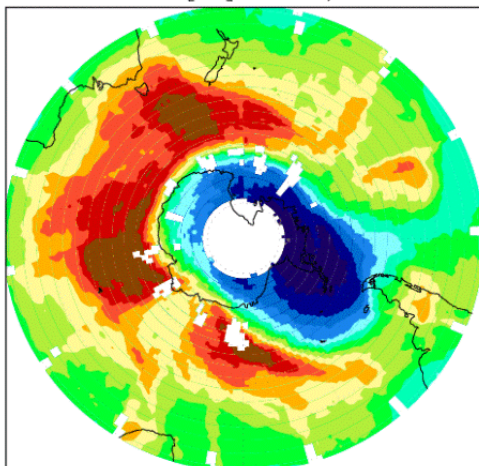
Rank	TOP500 Rank	System	Cores	Rmax (PFlop/s)	Power (kW)	Energy Efficiency (GFlops/watts)
1	222	JEDI - BullSequana XH3000, Grace Hopper Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, ParTec/EVIDEN EuroHPC/FZJ Germany	19,584	4.50	67	72.733
2	122	ROMEO-2025 - BullSequana XH3000, Grace Hopper Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, Red Hat Enterprise Linux, EVIDEN ROMEO HPC Center - Champagne-Ardenne France	47,328	9.86	160	70.912
3	440	Adastra 2 - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, RHEL, HPE Grand Equipement National de Calcul Intensif - Centre Informatique National de l'Enseignement Suprieur (GENCI-CINES) France	16,128	2.53	37	69.098

- 航空・宇宙
- 気象・地球環境
- 電磁解析
- 分子化学・分子設計
- 原子力・核融合
- ライフサイエンス
- ビッグデータ解析・AI
- 資源探索、軍事利用、金融 等

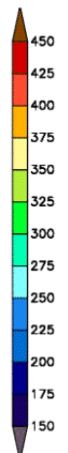
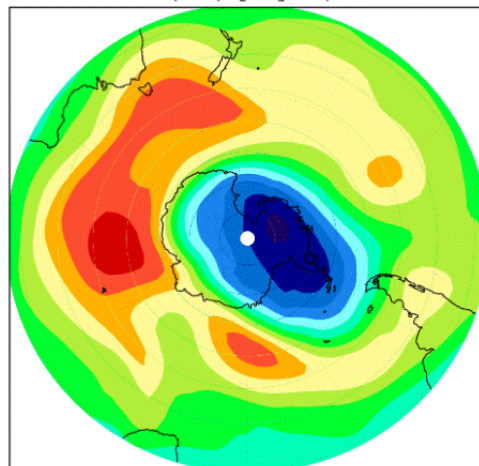
複雑な物理現象等の数値シミュレーションに用いられる

# 気象 シミュレーション

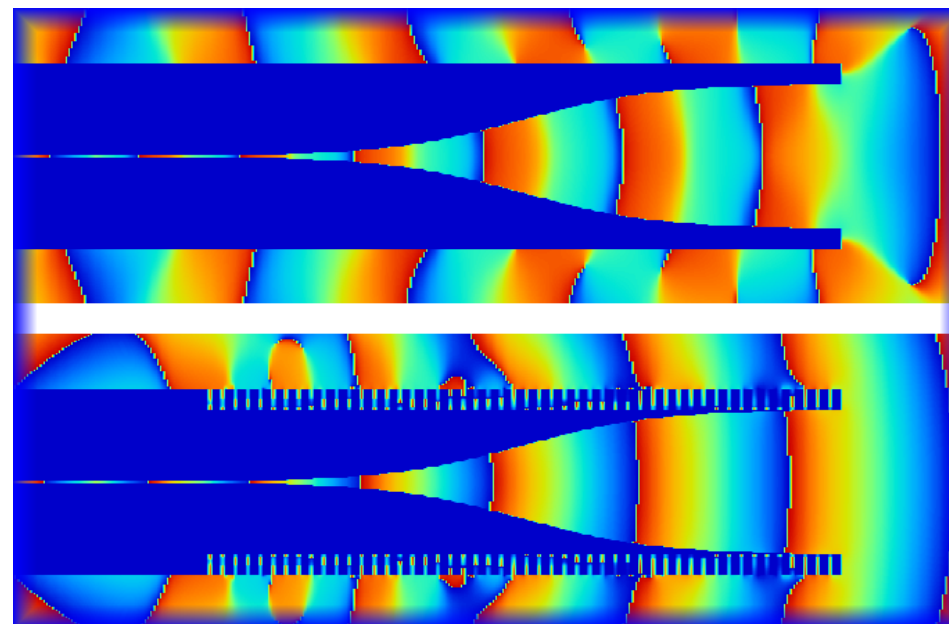
TOMS [DU] 2002 9/10



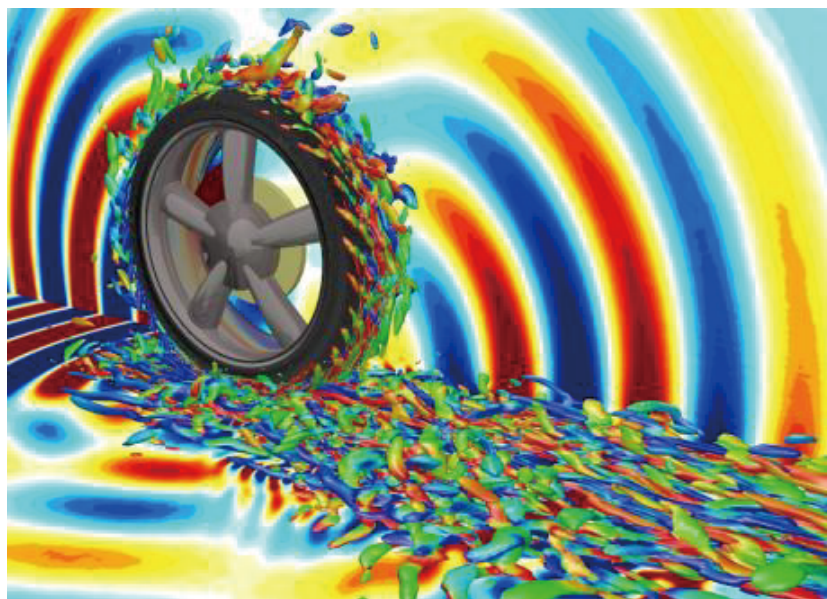
CTM(UVT) [DU] 09/10



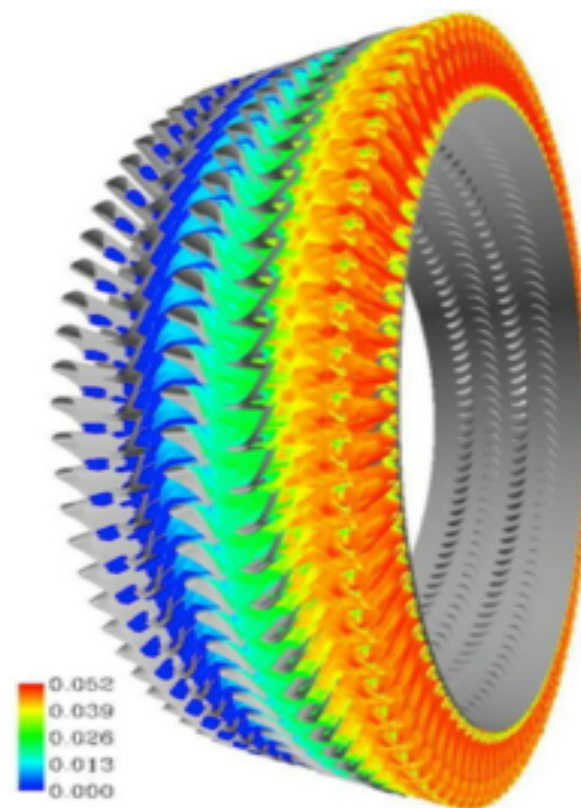
# アンテナ開発



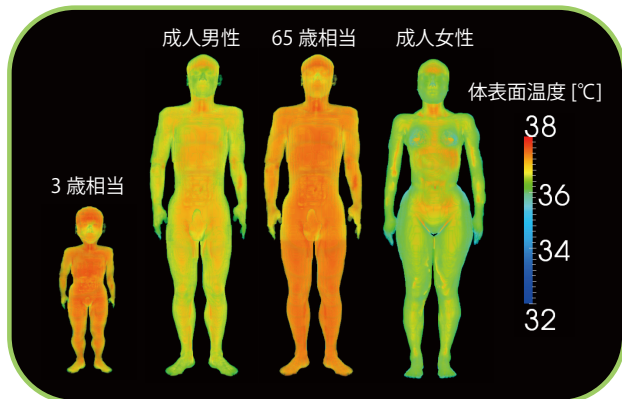
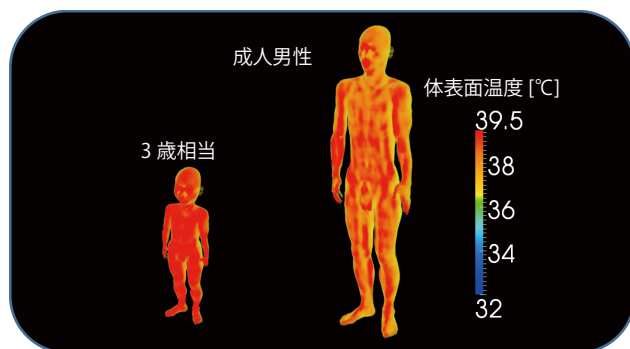
## 低騒音タイヤ開発



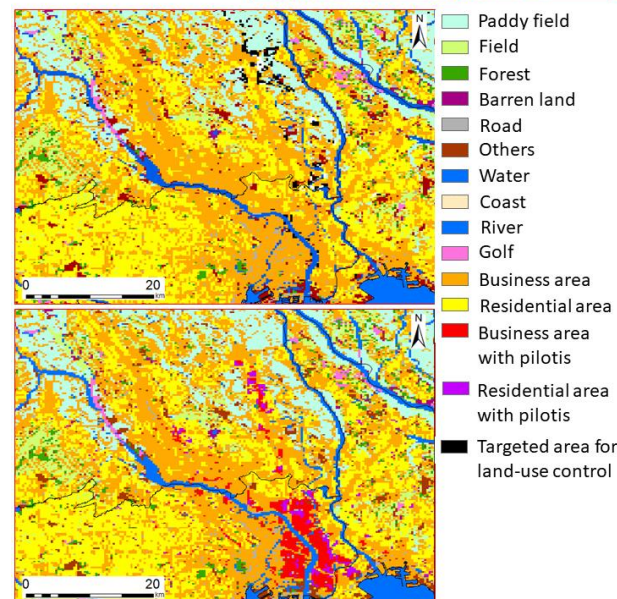
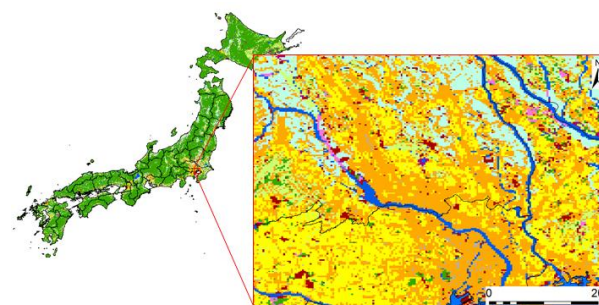
## CO2削減 タービン設計



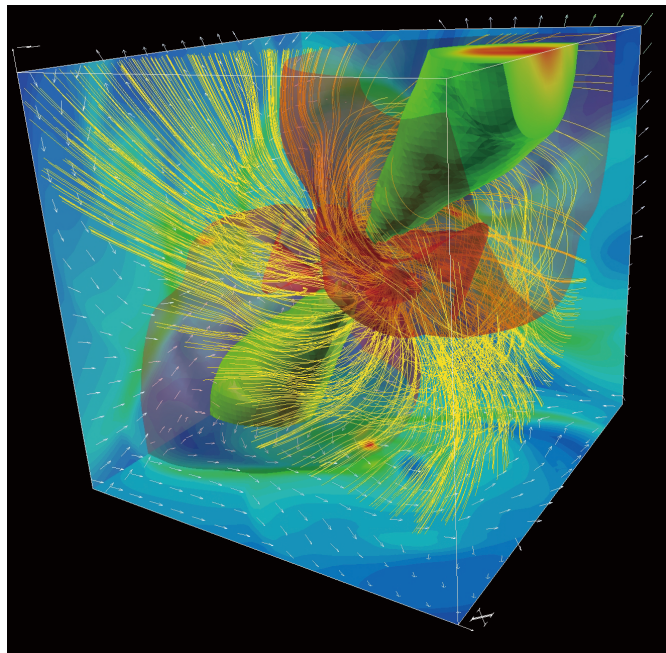
## 熱中症 シミュレーション



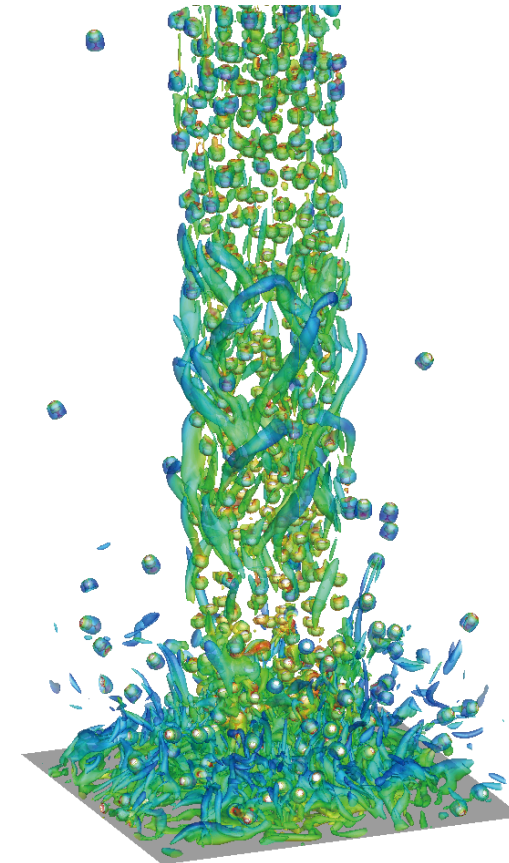
## 洪水被害予測 シミュレーション



星と惑星形成の  
大規模シミュレーション



大規模混相流解析



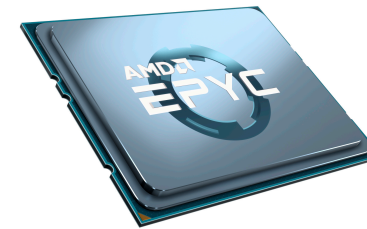
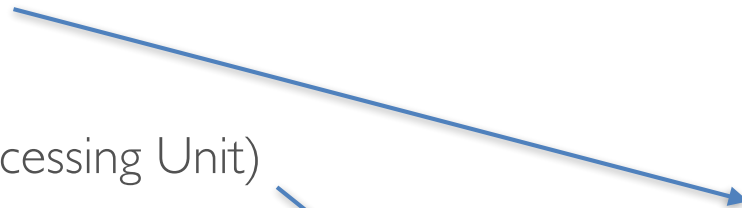
## ① プロセッサによる分類

a. ベクトル計算機



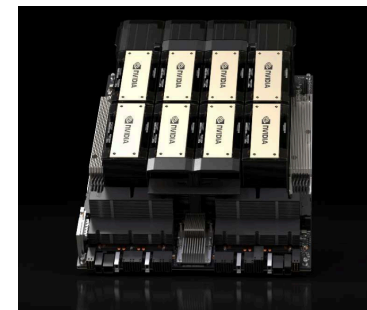
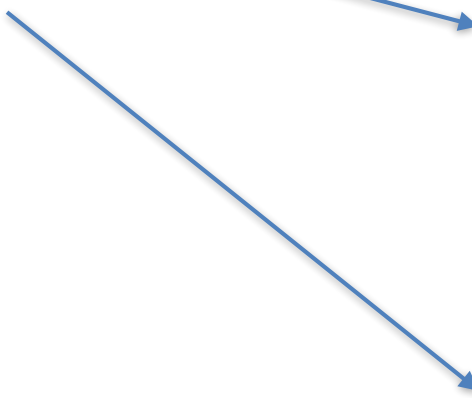
NEC SX-Aurora Type 30A, 20B  
(AOBA-S, A)

b. スカラ計算機



AMD EPYC 7720  
(AOBA-B)

c. GPU(Graphics Processing Unit)



NVIDIA H200

## ② メモリ構成による分類

a. 共有メモリ型

b. 分散メモリ型

①-a. ベクトル計算機

- ベクトル演算のための専用ハードウェアを持つプロセッサを搭載した計算機
- SXシリーズ (NEC) 、地球シミュレータ (NEC)
- 1980年代から1990年代にかけて、スパコンと言えばベクトル計算機のことだった
- SX-Aurora TSUBASAは x86 CPUとの組み合わせが必要
- センターではAOBA-SとAOBA-Aに搭載

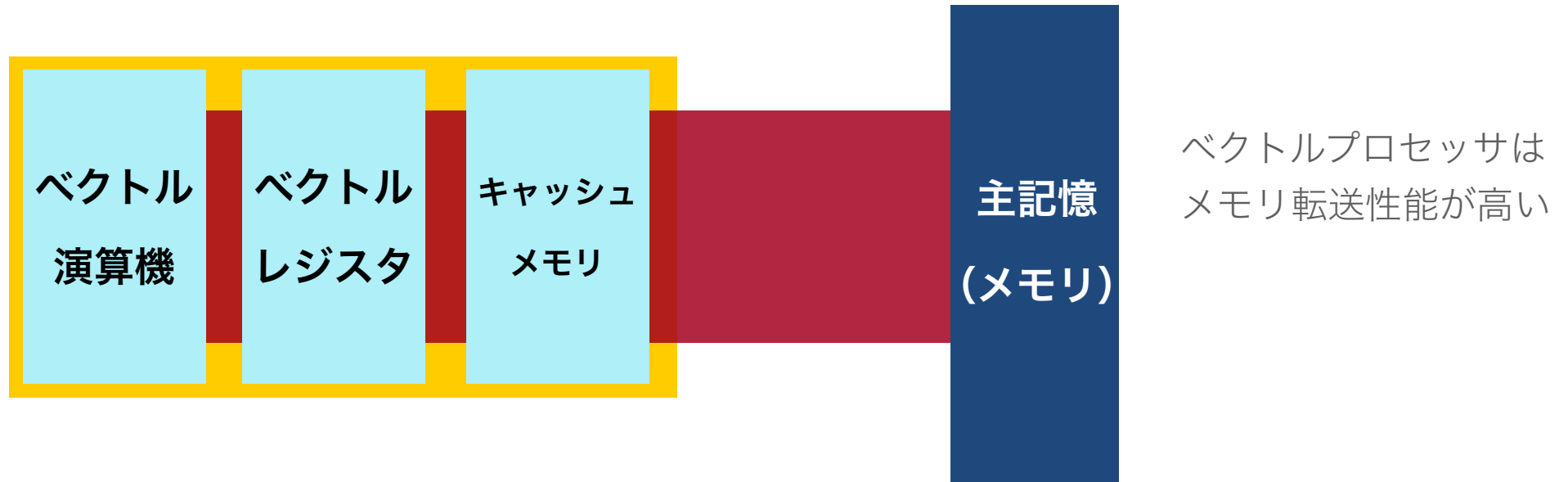
## ①-b. スカラ計算機

- 汎用プロセッサを搭載した計算機
- 汎用プロセッサは Xeon(Intel)、EPYC(AMD)、ARM(ARM)など普及品化しているプロセッサ
- 1990年代中盤以降、安価な汎用プロセッサを複数搭載した並列計算機が主流となる
- いわゆるHPCサーバ、PCクラスタ
- センターではAOBA-Bに搭載

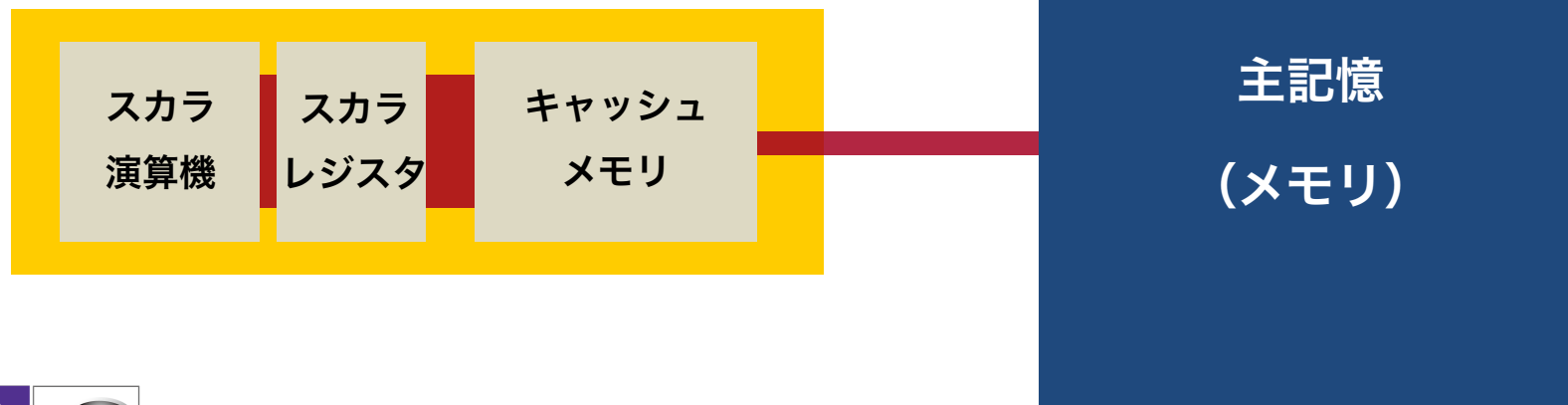
## ①-c. GPU

- 本来はコンピュータゲームに代表される、リアルタイム画像処理に特化した演算装置
- GPUのハードウェアを、より一般的な計算に活用
- AI100(NVIDIA), Instinct(AMD)
- 映像出力端子を持たない専用製品や、深層学習ベースのAI向けに特化した演算器を搭載
- 汎用CPUとの組み合わせが必要

## ■ベクトルプロセッサ

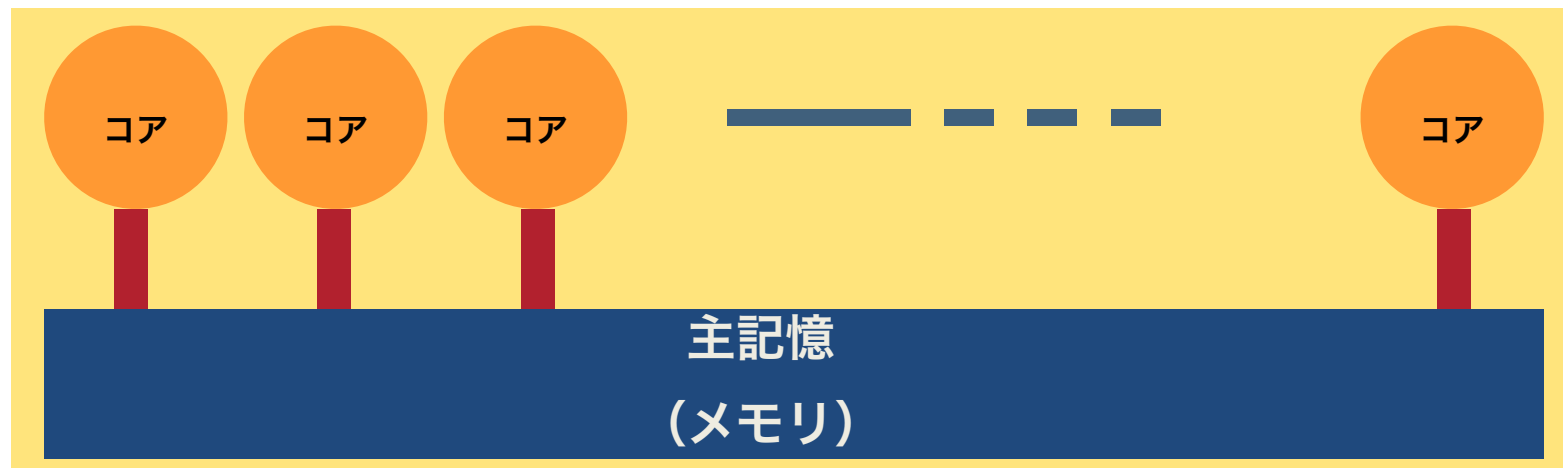


## ■スカラプロセッサ



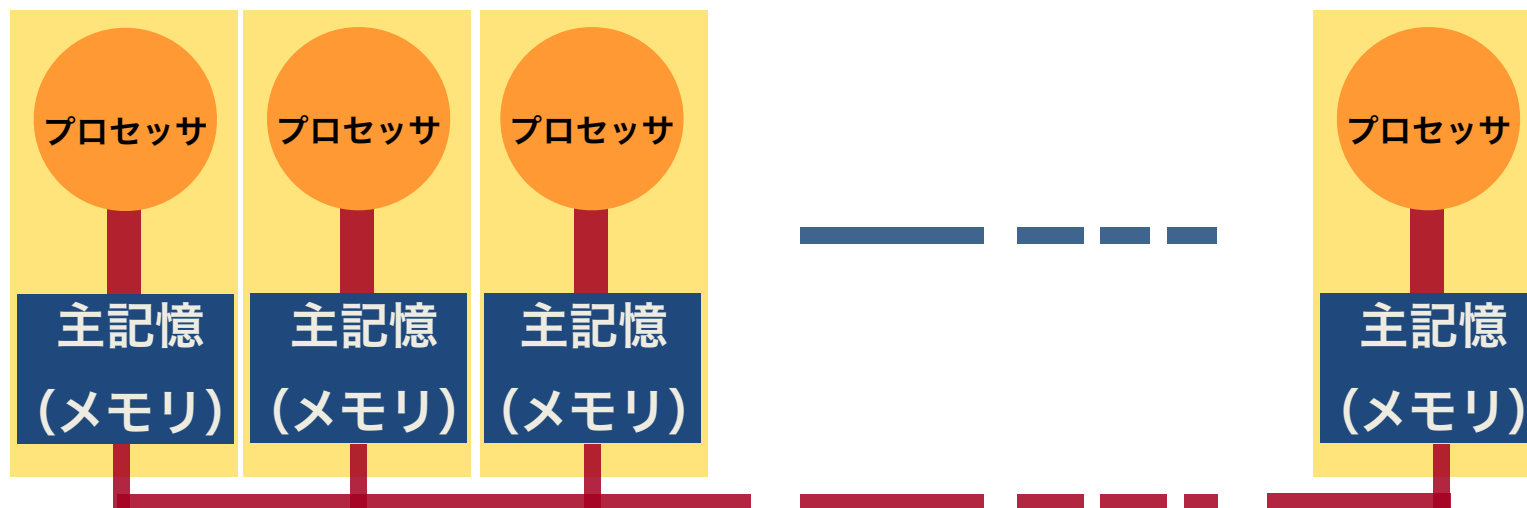
### ②-a. 共有メモリ型

- 1つの物理メモリを複数のプロセッサで共有するタイプ (SMP : Symmetric Multiple Processor)
- 自動並列化およびOpenMPによる並列化実行が可能
- MPI(Message Passing Interface)による並列化実行も可能
- AOBA-S,Aの各**VE内**並列、AOBA-Bの各**ノード内**並列



### ②-b. 分散メモリ型

- プロセッサごとにローカルな物理メモリを持ち、それらをInfiniBandなどのネットワークで複数接続するタイプ
- MPIにより並列実行する
- 自動並列化およびOpenMPによる並列化実行も同時利用が可能
- AOBA-S,Aの**複数VE**並列、AOBA-Bの**複数ノード**並列



## 並列実行に必要なこと

	既存のソースコード (逐次処理用)を 利用可能か？	並列化に要する作業	複数ノードでの実行 が可能か？	その他
自動並列	○	ない コンパイルオプション を付ける	× MPI並列化が必要	コンパイラまかせのためお手軽だが、高効率 で実行するにはユーザの一手間が必要な場合 あり。センターで最も利用されている並列化 手法。
OpenMP	指示行の挿入が必要	ある 並列化可能な箇所をユ ーザが判断し、ソース コードに指示文を追加 する	× MPI並列化が必要	ユーザの指示により並列化を行うため、意図 したとおりに動作する。演算結果の検証が必 要。自動並列の手動版的な手法。
MPI	MPI用のソースコードに 改変が必要	多い MPI用のソースコードを 作成する必要あり	○	MPI用プログラムを作成する必要があるが、大 規模並列化には不可欠。分散メモリ、共有メ モリともに利用可能。

## AOBA-S・AOBA-A・AOBA-Bのどれを使うか？

	AOBA-S (VE)	AOBA-A (VE)	AOBA-B (CPU)
利用 <b>最小</b> 単位	1VE	1VE	1ノード
コア数	16	8	<b>128</b>
理論演算性能 [TFLOPS]	<b>4.91</b>	2.45	4.09
メモリ容量 [GB]	96	48	<b>256</b>
コア性能 [GFLOPS]	<b>307</b>	<b>307</b>	32
メモリ転送性能 [TB/s]	<b>2.45</b>	1.53	0.40

	AOBA-S (VE)	AOBA-A (VE)	AOBA-B (CPU)
利用 <b>最大</b> 単位	2,048VE	512VE	16ノード
コア数	<b>32,768</b>	4,096	2,048
理論演算性能 [TFLOPS]	<b>10,055.6</b>	1,254.4	65.5
メモリ容量 [TB]	<b>192</b>	24	4

### AOBA-SまたはAOBA-Aを選択

- ・ シングルコア実行のプログラム
- ・ ベクトル演算向けのプログラム
- ・ メモリ転送性能が律速になるプログラム
- ・ MPIによる大規模並列を行う場合

### AOBA-Bを選択

- ・ ノード内並列化（OpenMP並列・自動並列）のみされているプログラム
- ・ ノード内でメモリ容量を多く使うプログラム
- ・ OSSなどVE向けのコンパイルが困難なプログラム

サブシステム AOBA-S (SX-Aurora TSUBASA Type 30A)

アプリケーション名	アプリ概要	備考
Quantum Espresso	第一原理電子状態計算パッケージ	pw.x(ver. 7.1)のみの提供
FPSEID21	近似的に電子励起が誘起する物質構造変化を計算	
Singularity	コンテナ型プラットフォーム	
ABINIT-MP	フラグメント分子軌道 (FMO) 計算を 高速に行えるソフトウェア	RISTが整備する アプリケーション
PHASE/0	密度汎関数理論に基づく擬ポテンシャル法による 平面波基底の第一原理電子状態計算プログラム	
SALMON	オープンソースの光と物質の相互作用をターゲットに した第一原理計算アプリケーションソフトウェア	
HΦ	オープンソースの数値厳密対角化法による 有効モデルソルバーパッケージ	
FrontISTR	超並列スパコンにも対応可能なオープンソースの 構造解析ソルバー	

サブシステム AOBA-A (SX-Aurora TSUBASA Type 20B)

アプリケーション名	アプリ概要	備考
Quantum Espresso	第一原理電子状態計算パッケージ	RISTが整備する アプリケーション
ABINIT-MP	フラグメント分子軌道 (FMO) 計算を 高速に行えるソフトウェア	
PHASE/0	密度汎関数理論に基づく擬ポテンシャル法による 平面波基底の第一原理電子状態計算プログラム	
HΦ	オープンソースの数値厳密対角化法による 有効モデルソルバーパッケージ	
FrontFlow/blue	非圧縮流体の非定常流動を高精度に予測可能な LESに基づいた汎用流体解析コード	
FrontISTR	超並列スパコンにも対応可能なオープンソースの 構造解析ソルバー	

サブシステムAOBA-B (LX 406-Rz2 EPYC 7702) (その1)

アプリケーション名	アプリ概要	備考
Gaussian16	非経験的分子軌道計算プログラム	
GRRM17	反応経路自動探索プログラム	学術目的での利用 (非商用利用)のみ可能
Mathematica	統合型数値計算ソフトウェア	東北大学の構成員のみ 利用可能
MATLAB	数値解析ソフトウェア	学術目的での利用 (非商用利用)のみ可能
OpenFOAM	熱流体解析ソフトウェア	
Quantum Espresso	第一原理電子状態計算パッケージ	
GROMACS	分子動力学シミュレーション	
LAMMPS	古典分子動力学計算プログラム	
Singularity	コンテナ型プラットフォーム	
Jupyter Notebook	プログラム対話型実行環境	

サブシステムAOBA-B (LX 406-Rz2 EPYC 7702) (その2)

アプリケーション名	アプリ概要	備考
ABINIT-MP	フラグメント分子軌道 (FMO) 計算を 高速に行えるソフトウェア	RISTが整備する アプリケーション
PHASE/0	密度汎関数理論に基づく擬ポテンシャル法による 平面波基底の第一原理電子状態計算プログラム	
GENESIS	超並列分子動力学計算ソフトウェア	
MODYLAS	オープンソース (ライセンス許可制) の 汎用古典分子動力学アプリケーションソフトウェア	
NTChem	Gauss型基底に基づいた量子化学計算 アプリケーションソフトウェア	
SALMON	オープンソースの光と物質の相互作用をターゲットに した第一原理計算アプリケーションソフトウェア	
HΦ	オープンソースの数値厳密対角化法による 有効モデルソルバーパッケージ	
OpenMX	オープンソースの第一原理計算 アプリケーションソフトウェア	
SMASH	オープンソースの量子化学計算 アプリケーションソフトウェア	

サブシステムAOBA-B (LX 406-Rz2 EPYC 7702) (その3)

アプリケーション名	アプリ概要	備考
mVMC	高精度な波動関数を変分モンテカルロ法によって数値的に求める有効モデルソルバーパッケージ	RISTが整備するアプリケーション
AkaiKKR	グリーン関数法(KKR法)を用いた第一原理計算プログラムパッケージ	
ALAMODE	格子振動の非調和性を露わに考慮した原子間ポテンシャルを構築するプログラムパッケージ	
Phonopy	第一原理計算ソフトウェアの計算結果から固体のフォノンに関連する物理量を計算するソフトウェア	
FrontFlow/blue	非圧縮流体の非定常流動を高精度に予測可能なLESに基づいた汎用流体解析コード	
FrontISTR	超並列スパコンにも対応可能なオープンソースの構造解析ソルバー	
FFX	「富岳」成果創出加速プログラムで開発された格子ボルツマン法による流体解析ソフトウェア	
FFVHC-ACE	完全自動、安定・高忠実な圧縮性流体解析基盤ソルバー	



- サイバーサイエンスセンター 大規模科学計算システムのウェブサイト <https://www.ss.cc.tohoku.ac.jp/>

- システムの利用マニュアル
- 運用についてのお知らせ
- 利用者支援 > 利用相談フォーム
- 利用者支援 > 利用講習会 > 講習会計画

- 利用申請からログインまでについては以下を参照

- システムの紹介 > 利用申請からログインまで

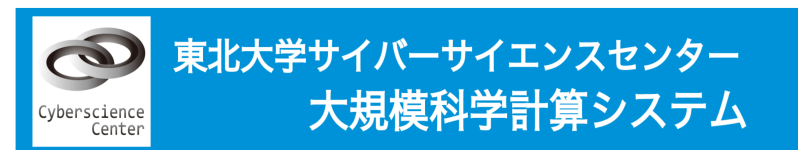
<https://www.ss.cc.tohoku.ac.jp/first-use/>



- 利用者用ポータル (LDAP認証連携)

- 公開鍵・秘密鍵ペアの作成
- 利用状況 (負担金, 合計課金対象時間, ジャーナルレコード等) の確認  
従来のフロントエンドサーバ上のコマンドでも確認可能

<https://www.ss.cc.tohoku.ac.jp/portal/>



スーパーコンピュータAOBA

2023.08.30

センターからのお知らせはこちらです。

サイバーサイエンスセンターは教育研究機関・民間研究機関のための高性能コンピューティング基盤を提供しています。

サブシステム AOBAS (SX-Aurora TSUBASA)  
4.032Ee (504Wh), 21.05PFLOPS(DP), 504TB Memory, 9.97PB/s

ペタネットワーク (10x+10E (40T))

ペタストレージシステム (Lustre)  
実効容量 4.5PB

サブシステム AOBAA (SX-Aurora TSUBASA)  
576Ee (72Wh), 1.48PFLOPS(DP), 45TB Memov, 895.68TB/s

大規模科学計算システム広報  
SENAC (Vol.56, No.3) の発行について

令和5年度サイバーサイエンスセンター講習会計画 (再掲載)

令和5年度共同研究の募集 (第2回) について

時間依存密度汎用数理論コード  
「FPSEID<sup>2</sup>」の提供開始について

計算機システムのサービス休止について

【重要】AOBA-Sの利用についての注意点

大規模並列ジョブチャレンジについて

夏季休業中の休館について

大規模科学計算システムの更新について (3)

**センターに利用申請** <https://www.ss.cc.tohoku.ac.jp/apply-for-use/> をご参照ください。

・ 大学・学術利用

負担金請求あり， 随時申請可能

・ 民間企業利用（成果公開型／成果非公開型）

負担金請求あり（2倍単価／4倍単価）， 課題審査あり， トライアルユースあり， 随時申請可能

・ センターとの共同研究（大学・学術・民間企業利用対象）

負担金請求あり， 共同研究助成あり， 応募期間あり

・ 利用者番号（アカウント）の初期登録料**なし**， 年間維持費**なし**

従量課金を基本とするため， 計算機を利用しない場合の負担金請求は**0円**

※ 利用者番号は年度を超える場合も自動継続され， ホーム領域（/uhome）のデータも保存されます。

### 各機関での課題募集

・ 学際大規模情報基盤共同利用・共同研究拠点公募型共同研究（JHPCN）

採択予算超過の場合に負担金請求あり， 応募期間あり

・ 革新的ハイパフォーマンス・コンピューティング・インフラ（HPCI）

負担金請求なし（採択資源量まで利用可能）， 応募期間あり

## 負担金制度 (1/3)

---

- ・利用者番号（アカウント）の初期登録料**なし**，年間維持費**なし**
  - 従量課金を基本とするため，計算機を利用しない場合の負担金請求は**0円**
  - ※ 利用者番号は年度を超える場合も自動継続され，ホーム領域（/uhome）のデータも保存されます。
- ・計算機利用負担金
  - 共有利用・従量
    - 課金対象時間（利用VH数または利用ノード数と，利用時間の積）に比例した課金方式
  - 共有利用・定額
    - 利用負担金の先払いにより，負担額の課金対象時間相当まで計算機を利用可能
    - 年度途中に定額負担金の追加も可能
  - 占有利用
    - 3ヶ月単位でAOBA-SおよびAOBA-A（8VE単位）またはAOBA-B（1ノード単位）を占有して利用
    - 特定利用者が計算資源を占有するため，他利用者のジョブ待ちが無い
    - （2025年度はAOBA-Bの占有利用の受付は終了）
- ・ストレージ負担経費
  - ホーム領域 5TBまで無料
  - 追加1TBにつき年額3,000円
- ・出力負担経費
  - センターの大判プリンタ1枚につき ソフトクロス紙 1,200円 光沢紙 600円
- ・民間企業利用については 成果公開型は2倍，成果非公開型は4倍の課金単価

サブシステムAOBA-S (SX-Aurora TSUBASA Type 30A)

利用形態	負担額および利用可能課金対象時間
共有 (無料)	利用 <b>VE 数 1</b> (実行数, 実行時間の制限あり) <b>無料</b>
共有 (従量)	課金対象時間 = (利用VE数 ÷ 8 を切り上げた数) × 経過時間 (秒) 課金対象時間の合計 1 時間につき <b>100 円</b> 課金対象時間は半期毎 (4~9 月および 10~3 月) に合計し, 1 時間未満を切上げて負担金を請求する
共有 (定額)	負担額 <b>10 万円</b> につき 課金対象時間の合計 <b>1,000時間</b>
占有	利用 <b>VE数8</b> , 利用期間 <b>3ヶ月</b> につき <b>216,000円</b>

サブシステムAOBA-A (SX-Aurora TSUBASA Type 20B)

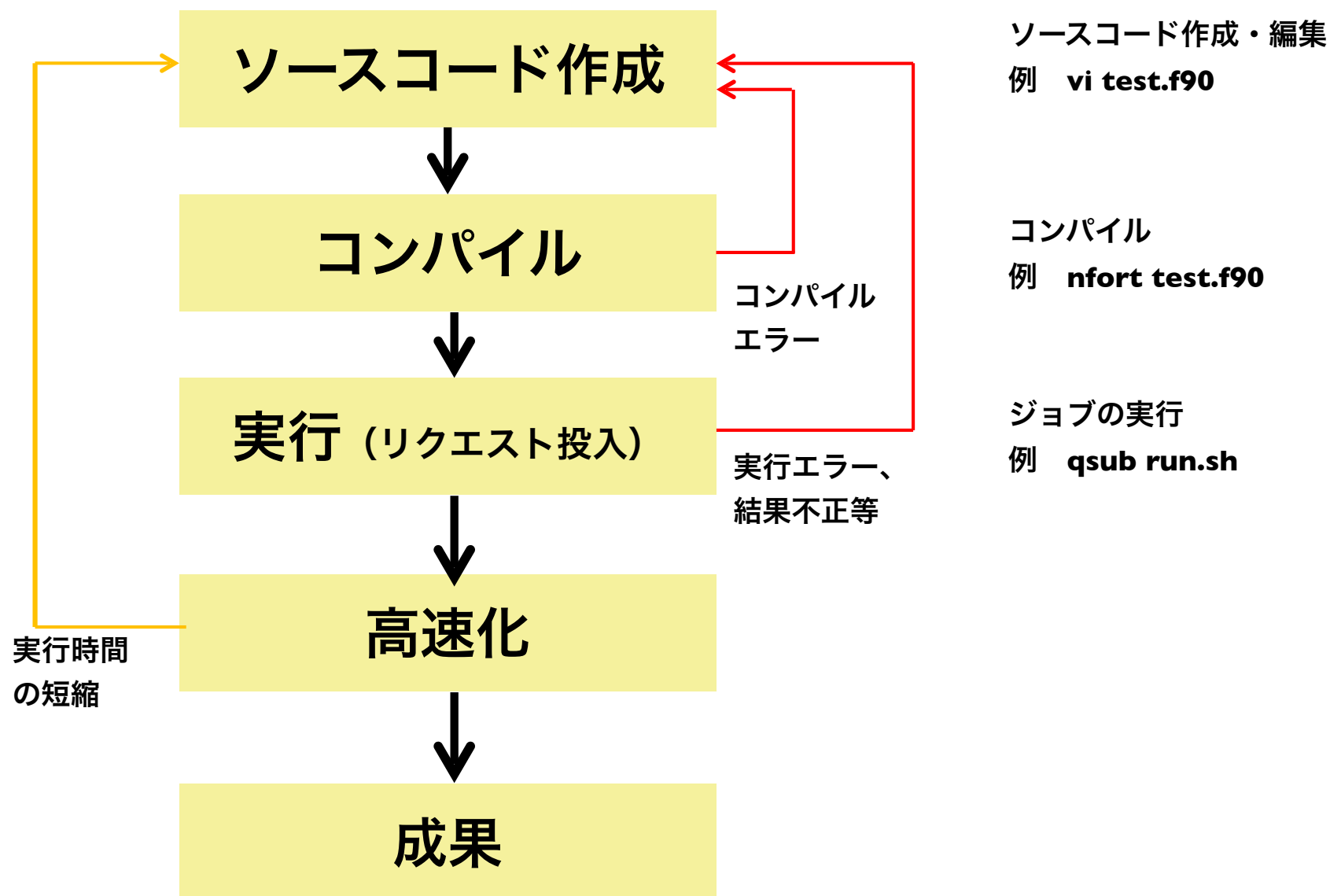
利用形態	負担額および利用可能課金対象時間
共有 (無料)	利用 <b>VE 数 1</b> (実行数, 実行時間の制限あり) <b>無料</b>
共有 (従量)	課金対象時間 = (利用VE数 ÷ 8 を切り上げた数) × 経過時間 (秒) 課金対象時間の合計 1 時間につき <b>75 円</b> 課金対象時間は半期毎 (4~9 月および 10~3 月) に合計し, 1 時間未満を切上げて負担金を請求する
共有 (定額)	負担額 <b>10 万円</b> につき 課金対象時間の合計 <b>1,400時間</b>
占有	利用 <b>VE数8</b> , 利用期間 <b>3ヶ月</b> につき <b>162,000円</b>

・ 民間企業利用については 成果公開型は**2倍**, 成果非公開型は**4倍**の課金単価

サブシステムAOBA-B (LX 406-Rz2 EPYC 7702)

利用形態	負担額および利用可能課金対象時間
共有 (従量)	<p>課金対象時間 = 利用ノード数 × 経過時間 (秒)</p> <p>課金対象時間の合計 1 時間につき <b>22 円</b></p> <p>課金対象時間は半期毎 (4~9 月および 10~3 月) に合計し, 1 時間未満を切上げて負担金を請求する</p>
共有 (定額)	<p>負担額 <b>10 万円</b>につき</p> <p>課金対象時間の合計 <b>4,600時間</b></p>
占有	<p>利用ノード数<b>1</b>, 利用期間<b>3ヶ月</b>につき <b>47,000円 (受付終了)</b></p>

- ・ 民間企業利用については 成果公開型は**2倍**, 成果非公開型は**4倍**の課金単価



## ジョブの実行方法

---

- ・ ジョブスクリプトファイルを作成し, qsub コマンドでリクエストを投入  
sfront\$ **qsub** ジョブスクリプトファイル名
- ・ バッチリクエストの実行状況, リクエストIDは, reqstat コマンドで確認  
sfront\$ **reqstat**
- ・ バッチリクエストのキャンセル, 途中終了はqdelコマンド  
sfront\$ **qdel** リクエストID

sfront\$ はAOBA-Sのフロントエンドサーバに接続した状態を示します。  
コマンドはこれ以降を入力します。

サブシステム AOBA-S (SX-Aurora TSUBASA Type 30A)

実行キュー名	利用可能VE数	実行時間制限 規定値/最大値	ジョブの実行形態
sxf	1	1時間/1時間	1VEジョブ 1時間無料 (VH を共用する)
sxs	1~2,048	72時間/720時間	8VE 単位で確保 (VH を共用しない)
sxsvh	1~16 ノード	72時間/720時間	CPU (EPYC) のみ利用 負担額は1ノード=8VE
inter	8	1時間/1時間	会話型
個別設定	契約VE数	720時間	占有利用

サブシステム AOBA-A (SX-Aurora TSUBASA Type 20B)

実行キュー名	利用可能VE数	実行時間制限 規定値/最大値	ジョブの実行形態
sxf	1	1時間/1時間	1VEジョブ 1時間無料 (VH を共用する)
sx	1	72時間/720時間	1VEジョブ (VH を共用する)
	2~256		8VE 単位で確保 (VH を共用しない)
個別設定	契約VE数	720時間	占有利用

## サブシステムAOBA-B (LX 406-Rz2 EPYC 7702)

実行キュー名	利用可能ノード数	実行時間制限 規定値/最大値	ジョブの実行形態
lx	1~16	72時間/720時間	ノードを共用しない
個別設定	契約ノード数	720時間	占有利用

## 逐次実行

**sfront\$ (プロンプト) に続くコマンドを入力します。**

【ソースコード】

ソースコードのコピー (ディレクトリごと)  
sfront\$ cd ~  
sfront\$ cp -r /mnt/lustre/ap/lecture/super/prog1 ./

【コンパイル】

AOBA-S (VE) 向けにコンパイル  
sfront\$ cd prog1  
sfront\$ nfort vec.f90 (逐次実行Fortranプログラム)  
(コンパイルメッセージが表示)  
sfront\$ ls (実行ファイル a.out が作成されていることを確認)

【リクエスト投入】

AOBA-S (VE) にバッチリクエストファイルの投入  
sfront\$ qsub run.sh  
(投入先のプロジェクトコードを確認)

【リクエストの状況確認】

バッチリクエストの実行待ち、実行中を確認  
front\$ reqstat  
  
(ジョブが終了するとなにも表示されません)

【結果の確認】

```
標準出力ファイルの確認
sfront$ ls (標準出力ファイル名の確認)
sfront$ cat run.sh.o12345
```

```
vc(1,1)= 7.6426033667289279E+05
```

【実効性能の確認】

```
標準エラー出力ファイルの確認
sfront$ ls (標準エラー出力ファイル名の確認)
sfront$ cat run.sh.e12345
```

```
***** Program Information *****
Real Time (sec) : 0.695214
User Time (sec) : 0.691426
Vector Time (sec) : 0.689939
Inst. Count : 714144395
V. Inst. Count : 348160039
V. Element Count : 89128968542
V. Load Element Count : 5368709152
FLOP Count : 65498251489
MOPS : 163599.970244
MOPS (Real) : 162657.917783
MFLOPS : 94729.228419
MFLOPS (Real) : 94183.752141
A. V. Length : 255.999996
V. Op. Ratio (%) : 99.676456
L1 Cache Miss (sec) : 0.000371
CPU Port Conf. (sec) : 0.000000
V. Arith. Exec. (sec) : 0.580189
V. Load Exec. (sec) : 0.065481
LD L3 Hit Element Ratio (%) : 0.028185
VLD LLC Hit Element Ratio (%) : 0.027911
FMA Element Count : 23622320128
Power Throttling (sec) : 0.000000
Thermal Throttling (sec) : 0.000000
Memory Size Used (MB) : 33290.000000
Non Swappable Memory Size Used (MB) : 98.000000

Start Time (date) : Fri Jun 28 12:51:55 2024 JST
End Time (date) : Fri Jun 28 12:51:56 2024 JST
```

## 自動並列実行

【ソースコード】

```
ソースコードのコピー (ディレクトリごと)
sfront$ cd ~
(sfront$ cp -r /mnt/stfs/ap/lecture/super/prog1 ./) (逐次実行プログラムと同じ)
```

【コンパイル】

```
AOBA-S (VE) 向けにコンパイル
sfront$ cd prog1
sfront$ nfort -mparallel vec.f90 (自動並列実行Fortranプログラム)
(コンパイルメッセージが表示)
sfront$ ls (実行ファイル a.out が新しく作成されていることを確認)
```

【リクエスト投入】

```
AOBA-S (VE) にバッチリクエストファイルの投入
sfront$ qsub run.sh
(投入先のプロジェクトコードを確認)
```

【リクエストの状況確認】

```
バッチリクエストの実行待ち、実行中を確認
sfront$ reqstat

(ジョブが終了するとなにも表示されません)
```

## OpenMP並列実行

【ソースコード】

ソースコードのコピー (ディレクトリごと)

```
sfront$ cd ~  
sfront$ cp -r /mnt/lustre/ap/lecture/super/prog2 ./
```

【コンパイル】

AOBA-S (VE) 向けにコンパイル

```
sfront$ cd prog2  
sfront$ nfort -fopenmp omp.f90 (OpenMP並列実行Fortranプログラム)  
(コンパイルメッセージが表示)  
ls (実行ファイル a.out が作成されていることを確認)
```

【リクエスト投入】

AOBA-S (VE) にバッチリクエストファイルの投入

```
sfront$ qsub run.sh  
(投入先のプロジェクトコードを確認)
```

【リクエストの状況確認】

バッチリクエストの実行待ち、実行中を確認

```
sfront$ reqstat  
  
(ジョブが終了するとなにも表示されません)
```

## MPI並列実行

【ソースコード】

```
ソースコードのコピー (ディレクトリごと)
sfront$ cd ~
sfront$ cp -r /mnt/lustre/ap/lecture/super/prog3 ./
```

【コンパイル】

```
AOBA-S (VE) 向けにコンパイル
sfront$ cd prog3
sfront$ mpinfort mpi.f90 (MPI並列実行Fortranプログラム)
(コンパイルメッセージが表示)
ls (実行ファイル a.out が作成されていることを確認)
```

【リクエスト投入】

```
AOBA-S (VE) にバッチリクエストファイルの投入
sfront$ qsub run.sh
(投入先のプロジェクトコードを確認)
```

【リクエストの状況確認】

```
バッチリクエストの実行待ち、実行中を確認
sfront$ reqstat

(ジョブが終了するとなにも表示されません)
```

## AOBA-SのVEとCPUでの演算時間の比較

AOBA-S (VE) 1VE 16core 4.91TFLOPS 96GB	コンパイルコマンド	全体演算時間 [sec]	演算部のみ[sec]
逐次実行	nfort vec.f90	0.695	—
自動並列実行(16core)	nfort -mparallel vec.f90	0.141	—
OpenMP並列実行(16core)	nfort -fopenmp omp.f90	0.141	—
MPI並列実行 (1VE, 16core)	mpinfort mpi.f90	0.397	0.043
MPI並列実行 (8VE, 128core)	mpinfort mpi.f90	1.379	0.013

AOBA-S (CPU) EPYC 7763 64core 2.5TFLOPS 256GB	コンパイルコマンド*	全体演算時間 [sec]	演算部のみ[sec]
逐次実行	ifx vec.f90 -mcmmodel=medium -march=znver3	10.615	—
MPI並列実行(64core)	mpiifx mpi.f90 -march=znver3	6.555	1.311

\* Intel OneAPIの利用は、bash環境 (/bin/bashの実行) で  
 module load compiler/latest mpi/latest コマンドの実行が必要  
 ジョブスクリプトファイルにも記載が必要

---

NEC Software Development kit for Vector Engine

(AOBA-A向けのコンパイルは front 上で)

- ・ 逐次実行

sfront\$ **nfort** コンパイルオプション Fortranソースファイル名

sfront\$ **ncc** コンパイルオプション Cソースファイル名

sfront\$ **nc++** コンパイルオプション C++ソースファイル名

- ・ 自動並列化

sfront\$ **nfort -mparallel** コンパイルオプション Fortranソースファイル名

sfront\$ **ncc -mparallel** コンパイルオプション Cソースファイル名

sfront\$ **nc++ -mparallel** コンパイルオプション C++ソースファイル名

- ・ OpenMP並列化

sfront\$ **nfort -fopenmp** コンパイルオプション Fortranソースファイル名

sfront\$ **ncc -fopenmp** コンパイルオプション Cソースファイル名

sfront\$ **nc++ -fopenmp** コンパイルオプション C++ソースファイル名

- ・ MPI並列化 (自動並列化, OpenMP並列化の併用も可能)

sfront\$ **mpinfort** コンパイルオプション Fortranソースファイル名

sfront\$ **mpincc** コンパイルオプション Cソースファイル名

sfront\$ **mpinc++** コンパイルオプション C++ソースファイル名

Intel OneAPI

Intelコンパイラ環境を利用するために、bash環境で以下のコマンドを実行する必要がある

```
front$ source /opt/oneapi/setvars.sh
```

- ・ 逐次実行

```
front$ ifx コンパイルオプション Fortranソースファイル名
```

```
front$ icx コンパイルオプション Cソースファイル名
```

```
front$ icpx コンパイルオプション C++ソースファイル名
```

- ・ OpenMP並列化

```
front$ ifx -qopenmp コンパイルオプション Fortranソースファイル名
```

```
front$ icx -qopenmp コンパイルオプション Cソースファイル名
```

```
front$ icpx -qopenmp コンパイルオプション C++ソースファイル名
```

- ・ MPI並列化 (OpenMPIを利用, OpenMP並列化の併用も可能)

```
front$ mpiifx コンパイルオプション Fortranソースファイル名
```

```
front$ mpiicx コンパイルオプション Cソースファイル名
```

```
front$ mpiicpx コンパイルオプション C++ソースファイル名
```

## AMD Optimizing C/C++ Compiler (AOCC)

## ・ 逐次実行

front\$ **flang** コンパイルオプション Fortranソースファイル名

front\$ **clang** コンパイルオプション Cソースファイル名

front\$ **clang++** コンパイルオプション C++ソースファイル名

## ・ OpenMP並列化

front\$ **flang -fopenmp** コンパイルオプション Fortranソースファイル名

front\$ **clang -fopenmp** コンパイルオプション Cソースファイル名

front\$ **clang++ -fopenmp** コンパイルオプション C++ソースファイル名

## ・ MPI並列化 (OpenMPIを利用, OpenMP並列化の併用も可能)

front\$ **mpifort** コンパイルオプション Fortranソースファイル名

front\$ **mpicc** コンパイルオプション Cソースファイル名

front\$ **mpic++** コンパイルオプション C++ソースファイル名

最適化のコンパイルオプション `-march=znver2` でRome向け最適化コンパイル

## Intel OneAPI ベース &amp; HPCツールキット

Intelコンパイラ環境を利用するために、bash環境で以下のコマンドを実行する必要がある

```
sfront$ module load compiler/latest mkl/latest mpi/latest
```

## ・ 逐次実行

```
sfront$ ifx コンパイルオプション Fortranソースファイル名
```

```
sfront$ icx コンパイルオプション Cソースファイル名
```

```
sfront$ icpx コンパイルオプション C++ソースファイル名
```

## ・ OpenMP並列化

```
sfront$ ifx -qopenmp コンパイルオプション Fortranソースファイル名
```

```
sfront$ icx -qopenmp コンパイルオプション Cソースファイル名
```

```
sfront$ icpx -qopenmp コンパイルオプション C++ソースファイル名
```

## ・ MPI並列化 (OpenMPIを利用, OpenMP並列化の併用も可能)

```
sfront$ mpiifx コンパイルオプション Fortranソースファイル名
```

```
sfront$ mpiicx コンパイルオプション Cソースファイル名
```

```
sfront$ mpiicpx コンパイルオプション C++ソースファイル名
```

**利用者支援** <https://www.ss.cc.tohoku.ac.jp/support/>



## ・利用者講習会

- システムの利用法, コードの高速化・並列化, ネットワーク・セキュリティ, アプリケーション利用方法について, 年間10回程度開催
- センター内端末機室および遠隔配信で実施

## ・利用相談

- 利用申請の方法, システムの利用方法, コンパイルエラー, ジョブの投入方法, コードの高速化など
- 利用相談フォームで受け付け <https://www.ss.cc.tohoku.ac.jp/consultation/>
- メールで継続的にサポート
- 年間約200件前後



## ・高速化支援

- 利用者, センター教職員, ベンダーが協力して利用者コードの高速化・並列化を実施
- コード大規模化のサポート, JHPCN課題, HPCI課題へのステップアップを支援
- 1997年から継続的な取り組み
- 年間5件程度を実施
- AOBA-A(2020年度～2023年度)ではベクトル最適化(平均2.6倍)とMPI並列化(平均約21倍)を15件実施
- AOBA-S(2023年度～2024年度)ではベクトル最適化(平均2.5倍)とMPI並列化(平均約4.9倍)を6件実施



東北大学サイバーサイエンスセンター

スーパーコンピュータAOBAのご利用をお待ちしております