内容に関する質問は
katagiri@cc.nagoya-u.ac.jp
まで

2025年9月11日（木）13:35-13:55
現地（名古屋大学情報基盤センター2F演習室）
とオンライン（Zoom）のハイブリッド開催

「不老」運用状況報告
「不老NEXT」（仮称）導入日程
将来の新サービスに向けて：コード生成AI支援サービスとHPC-GENIEプロジェクト

名古屋大学情報基盤センター　片桐孝洋

不老」運用状況報告

スーパーコンピュータ「不老」の役割

1. 全国共同利用・共同研究拠点として学内外へ計算資源提供
   - **全国共同利用・共同研究拠点**として国が位置づけ
     - 全国の研究者の世界トップレベル研究を強力に支援

世界トップレベル研究の支援

HPCI (High Performance Computing Infrastructure) 利用者

国策スパコン利用支援

名大拠点利用者

JHPCN利用者

名大「不老」Type I システム（富岳型ノード）

スーパーコンピュータ「富岳」

簡便な移行支援

導入支援／高性能化／特殊処理／長時間実行

世界トップレベル研究成果創出

2. ものつくり企業支援（地域イノベーションコア形成）
   - **産業利用制度（公開、非公開）**
   - 計算機利用型講習会による並列処理・大規模計算普及（地域特有の中小企業支援）
3. 新しい計算需要に向けたサービス開拓
   - **データサイエンス（ビッグデータ）、AI基盤の提供による新サービスの開拓**
4. 指定国立大学として重要な役割
   - **数理・データ科学教育**
   - 人材育成・研究力強化・社会との連携

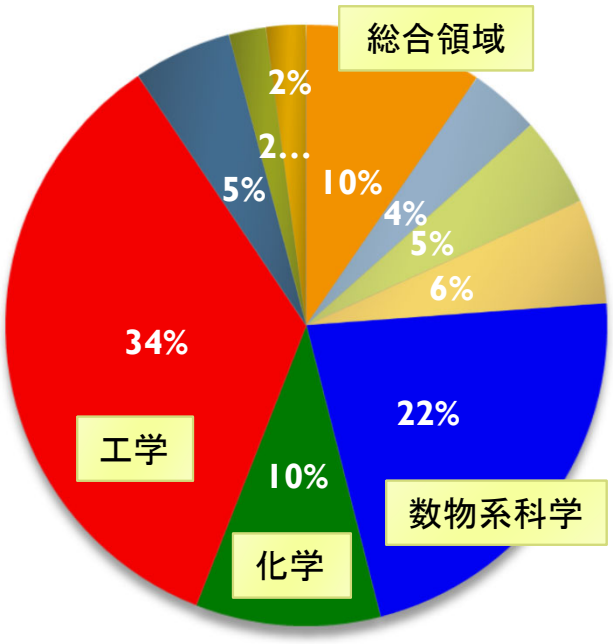
・数理データ科学分野の人材育成
・AI技術基盤提供（大規模AI計算基盤、大規模ストレージ、サイバーフィジカル）
・技術コンサルティング・・・等による社会貢献

ユーザの研究分野の変遷

※2020年10月現在

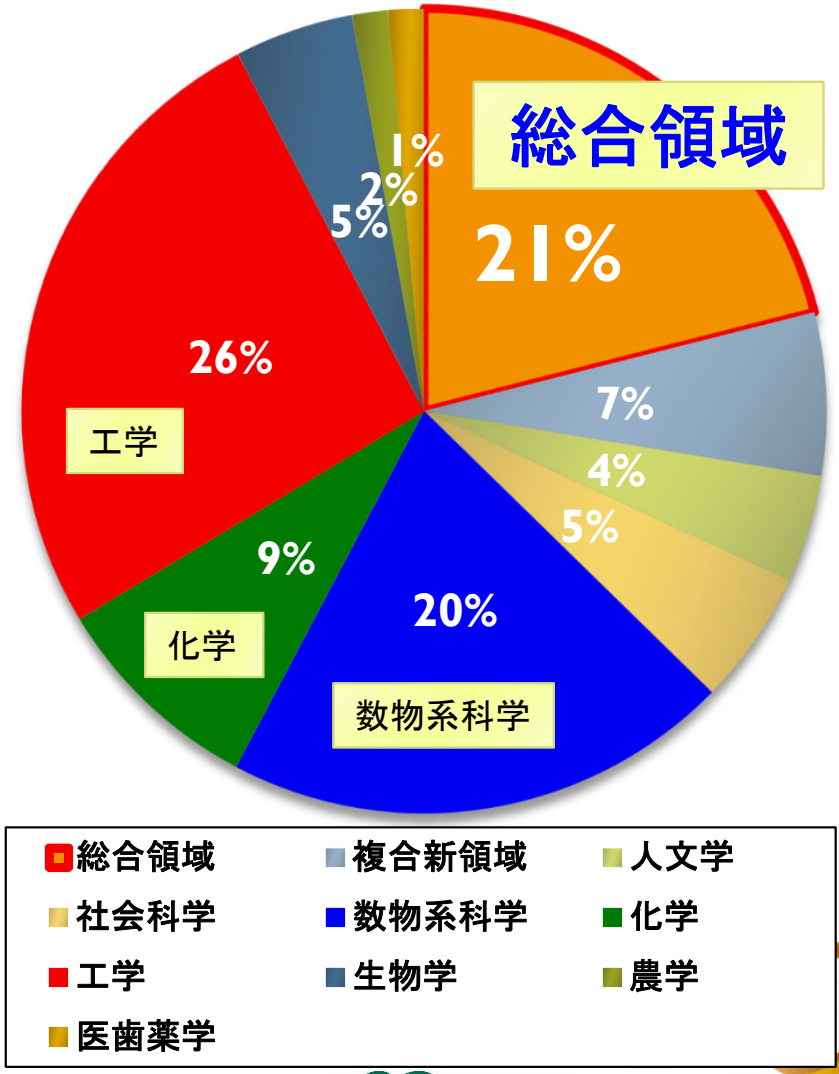
スーパーコンピュータ「不老」

旧システム（FX100）



「**総合領域**」
の割合向上
10% → **21%**

「**情報学**」
分野ユーザ数
31 → **105**



AI・データサイエンス研究が増加？

スーパーコンピュータ「不老」の特徴

- 以下の新しいスパコン利用法を提供しています:
  - **①Type Ⅰ サブシステム(「富岳」型)による超並列処理**
  - **②Type Ⅱ サブシステム(GPUクラスタ)による大規模機械学習**
  - **③ 数値計算 ＋ AI 処理**
  - **④ ①～③のシームレスなデータ可視化**
  - **⑤ ①～④で必要な大規模・堅牢なデータ蓄積**

- ④に必要なTypeⅢサブシステム(大規模共有メモリ(48TB))、精細／遠隔可視化装置が、伝統的に名古屋大学は充実
- ⑤のコールドストレージ(100年保存可能な光ディスク)搭載スパコンは業界初

名古屋大学情報基盤センターのスパコン利用の特徴

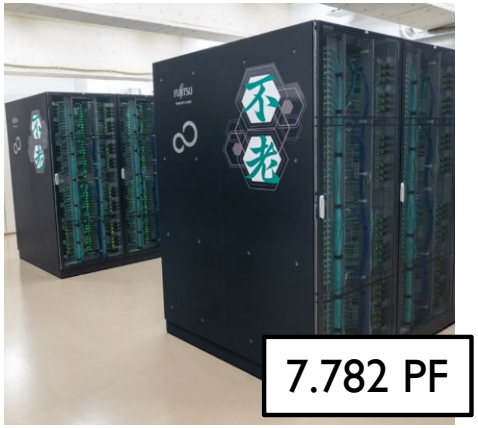
- **計画書提出なし／論文出版等の義務なく利用可能**
  - 名古屋大学拠点利用の場合（HPCI、JHPCN利用を除く）
  - 年間随時募集（ただし、提供資源がなくなった場合を除く）
  - アカデミックの方
    - 研究目的（平和利用）、予算確保（運営費、科研費等）が明確であれば、必ず利用承認されます
      - 研究目的の記載は数行
    - 成果報告書は１ページ
      - 事実上、発表・出版論文などの情報の登録をお願いするのみ
    - 申込書提出後、１週間程度で承認、アカウント発行
  - 民間利用の方
    - 課題の計画書提出が必要
    - 審査委員会で審議の上で承認（１～２週間程度）。その後、アカウント発行。
    - 報告書はアンケート程度

スーパーコンピュータ「富岳」との連携

1. 同型計算機・同一ソフトウェアスタック
   - 「不老」Type I サブシステムはCPUが「富岳」と同型
   - OS・コンパイラ等のソフトウェアスタックも同一と想定
     - ☞「不老」のほうがより進んだバージョンが提供される可能性があり
2. 国策ソフトウェアの「不老」Type I サブシステムのプリインストール・講習会実施
   - 国費開発ソフトウェアで「富岳」で動作するソフトウェアのいくつかを、一般財団法人　高度情報科学技術研究機構（RIST）の協力のもと、「不老」Type I で提供、ハンズオン講習会も実施
     - ☞講習会予定をご覧ください
3. 「富岳」向けチューニングと「富岳」への移行支援
   - コンサルティング・教員との共同研究で「富岳」向けと想定されるコードチューニング（☞「不老」Type I 向けと等価）を支援します

スーパーコンピュータ「不老」
主な構成要素

Type I, II, III, クラウドの合計で15.886PFLOPS
（旧システムの約4倍）



Type Iサブシステム
FUJITSU Supercomputer FX1000
「富岳」型

Type IIサブシステム
FUJITSU Server PRIMERGY CX2570 M5
GPUスパコン

Type IIIサブシステム
HPE Superdome Flex
大容量メモリ・可視化

クラウドシステム
HPE ProLiant DL560
バッチ＆インタラクティブ

ホットストレージ
FUJITSU PRIMGERY RX2540 M5
FUJITSU ETERNUS AF250 S2
DDN SFA18KE
DDN SS9012

コールドストレージ
SONY PetaSite 拡張型 Library

性能諸元（主要サブシステム群）

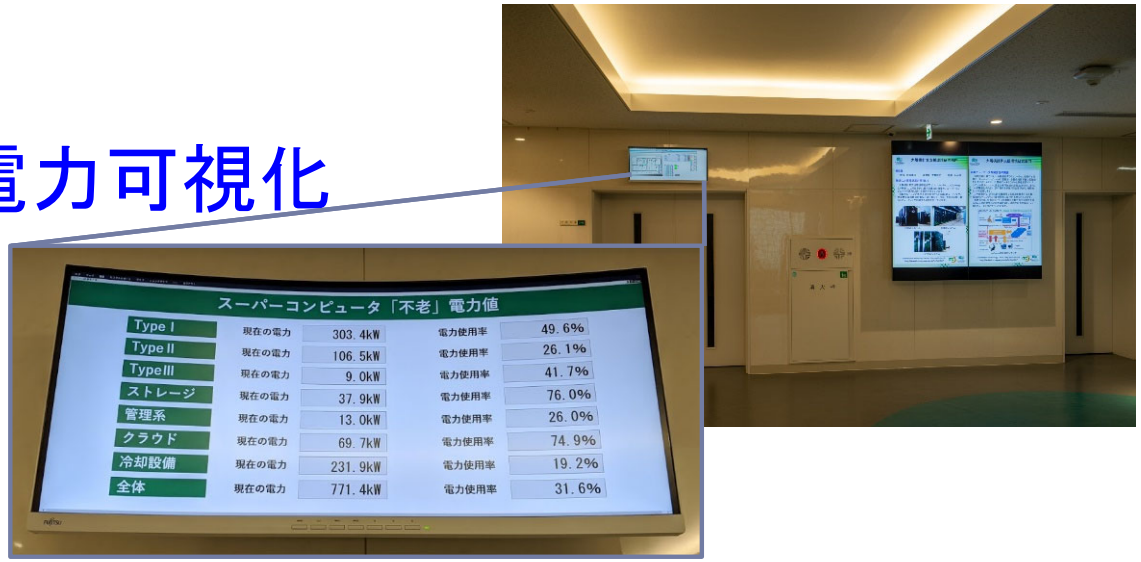
		Type I	Type II	Type III	クラウド
ノードあたり	CPU	A64FX ×1 (Armv8.2-A + SVE) 48+2コア、2.2GHz	Xeon Gold 6230×2 (Cascade Lake) 20コア、2.10-3.90 GHz	Xeon Platinum 8280M×16 (Cascade Lake) 28コア、2.70-4.00 GHz	Xeon Gold 6230×4 (Cascade Lake) 20コア、2.10-3.90 GHz
	メインメモリ	HBM2, 32GB	DDR4, 384GB	DDR4, 24TB	DDR4, 384GB
	GPU	-	Tesla V100×4 (Volta) HBM2, 32GB	Quadro RTX6000×4 (Turing) GDDR6, 24GB	-
	理論性能	3.3792 TFLOPS(DP) 1,024 GB/s	・CPU 1.344 TFLOPS(DP)×2 140.784 GB/s×2 ・GPU 7.8 TFLOPS(DP)×4 900 GB/s×4	・CPU 2.4192 TFLOPS(DP)×16 140.784 GB/s×16	1.344 TFLOPS(DP)×4 140.784 GB/s×4
ノード数		2,304	221	2	100
ノード間接続		TofuインターコネクトD	InfiniBand EDR ×2	InfiniBand EDR	InfiniBand EDR
総理論性能		7.782 PFLOPS(DP) 2.359 PB/s	7.489 PFLOPS(DP) 857.8 TB/s	77.414 TFLOPS(DP) 2.253 TB/s	537.6 TFLOPS(DP) 56.314 TB/s
冷却方式		水冷	水冷	空冷	空冷

消費電力・省電力対策

最大消費電力

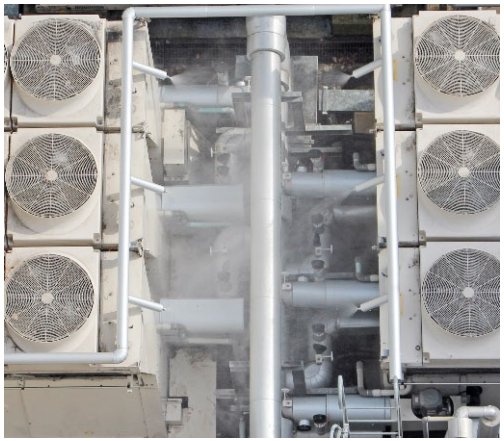
サブシステム名	消費電力
Typelサブシステム	628.1kVA
TypeIIサブシステム	393.5kVA
TypeIIIサブシステム	21.6kVA
クラウドシステム	93.0kVA
ストレージ	49.9kVA
フロントエンド	19.6kVA
運用管理システム他	52.3kVA
冷却設備	641.9kVA
合　計	1,899.9kVA

電力可視化



湧水を用いた冷却

- 地下の湧水を活用したら総合評価時加点
- 屋外チラーに散水して冷却



名古屋大学 NAGOYA UNIVERSITY

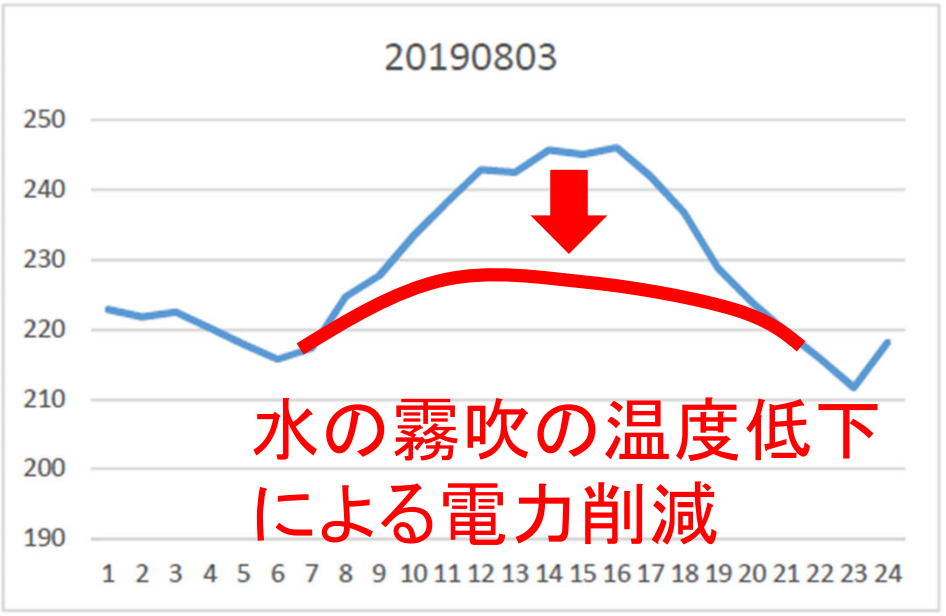
湧水による冷却システム

- 情報基盤センターの地下は
  夏季でも18℃程度の
  湧き水が毎分30L程度湧く
- この湧き水は、地下から
  ポンプで吸い上げて雨水
  扱いで捨てていた
- 今回の仕様で、湧き水
  を冷熱源として使用する
  場合は加点
  - 冷却水としての利用許可・
    水質検査済み

湧水による冷却システム

- 気温の高い4月から11月の間で利用
- 夏季の1日（2019年8月3日）の（旧）FX100システムの水冷チラーの電気使用量(KW)

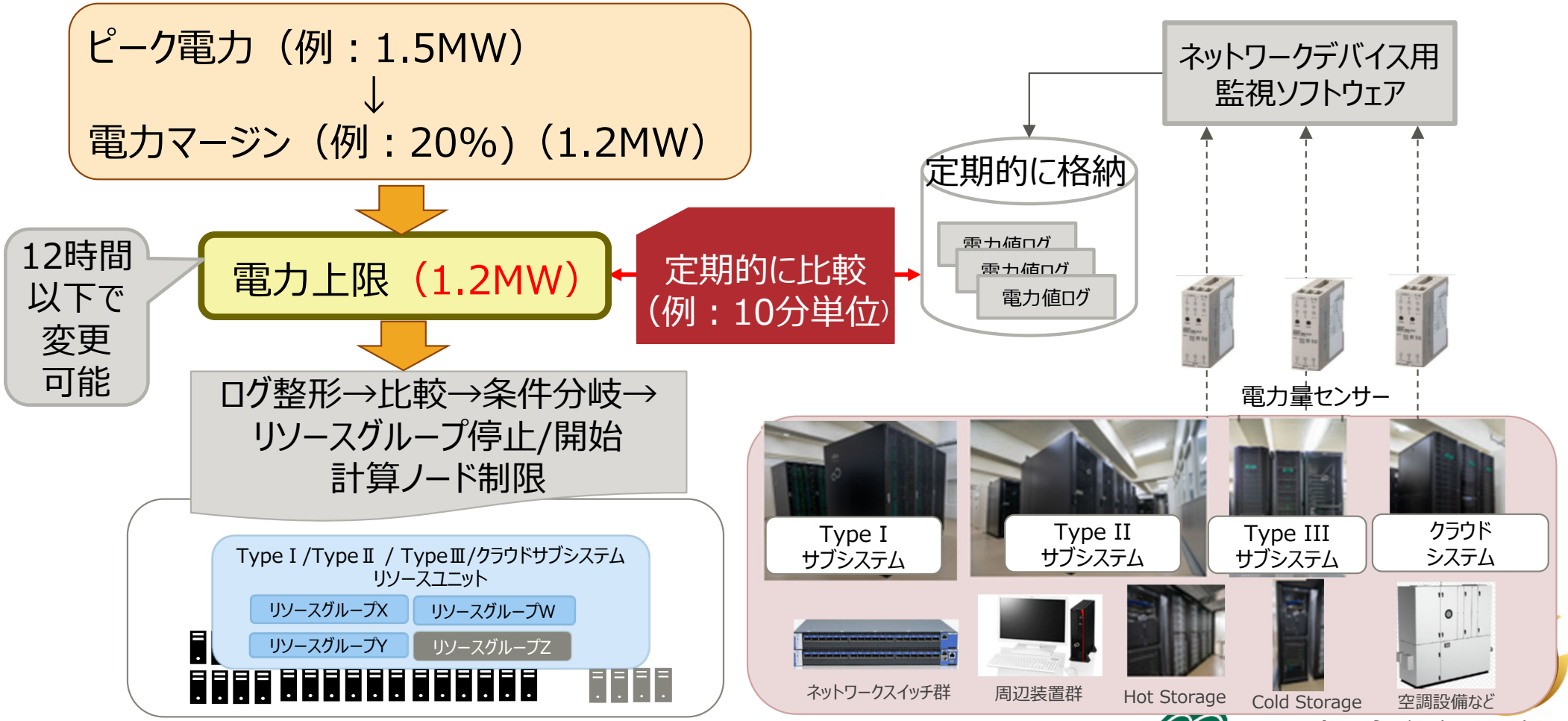
2019年08月03日	222.9
2019年08月03日	221.83
2019年08月03日	222.53
2019年08月03日	220.26
2019年08月03日	217.94
2019年08月03日	215.78
2019年08月03日	217.38
2019年08月03日	224.7
2019年08月03日	227.77
2019年08月03日	233.41
2019年08月03日	238.28
2019年08月03日	242.89
2019年08月03日	242.52
2019年08月03日	245.71
2019年08月03日	245.09
2019年08月03日	246.03
2019年08月03日	241.92
2019年08月03日	236.67
2019年08月03日	228.74
2019年08月03日	223.93
2019年08月03日	219.69
2019年08月03日	215.88
2019年08月03日	211.76
2019年08月03日	218.28



年間数百万円程度の電気代削減を予想

使用最大電力の動的制御機構

- 監視ソフトウェアから一定時間毎に電力値を取得
- 出力された電力値と、あらかじめ規定したシステム全体の使用最大電力の上限値を比較し、最大電力の上限を超えないよう、計算ノードやジョブ実行可能範囲を制限

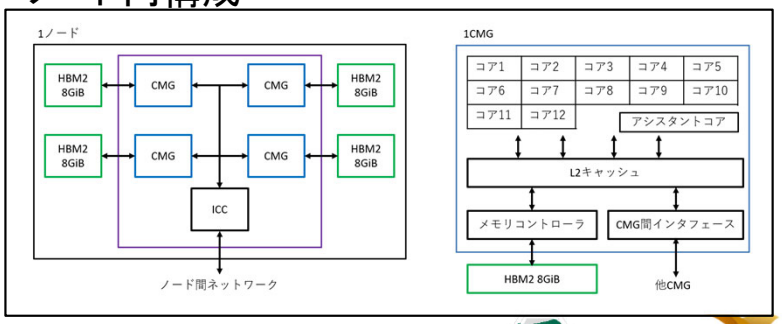


各サブシステムの仕様と特徴：Type I サブシステム

機種名		FUJITSU Supercomputer PRIMEHPC FX1000
計算ノード	CPU	A64FX (Armv8.2-A + SVE), 48コア+2アシスタントコア(I/O兼計算ノードは48コア+ 4アシスタントコア), 2.2GHz, 4ソケット相当のNUMA
	メインメモリ	HBM2, 32GiB
	理論演算性能	倍精度 3.3792 TFLOPS, 単精度 6.7584 TFLOPS, 半精度 13.5168 TFLOPS
	メモリバンド幅	1,024 GB/s (1CMG=12コアあたり256 GB/s, 1CPU=4CMG)
ノード数、総コア数		2,304ノード, 110,592コア (+4,800アシスタントコア)
総理論演算性能		7.782 PFLOPS
総メモリ容量		72 TiB
ノード間 インターコネクト		TofuインターコネクトD 各ノードは周囲の隣接ノードへ同時に合計 40.8 GB/s × 双方向 で通信可能(1リンク当たり 6.8 GB/s × 双方向, 6リンク同時通信可能)
ユーザ用 ローカルストレージ		なし
冷却方式		水冷

- 世界初正式運用のスーパーコンピュータ「富岳」型システム
- 自己開発のMPIプログラム向き
- 超並列処理用
- AIツールも提供

ノード内構成

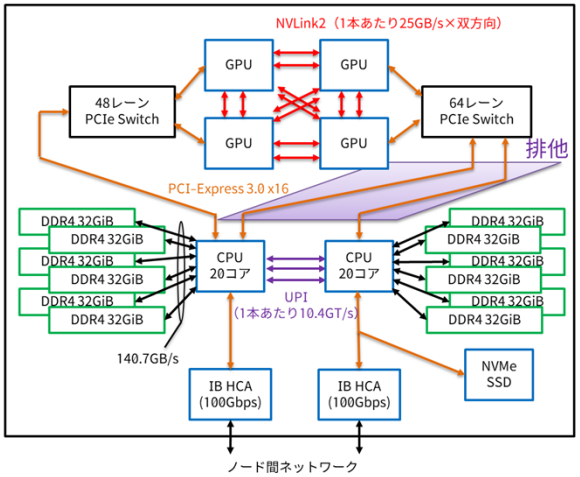


各サブシステムの仕様と特徴：Type II サブシステム

機種名		FUJITSU Server PRIMERGY CX2570 M5
計算ノード	CPU	Intel Xeon Gold 6230, 20コア, 2.10 - 3.90 GHz × 2 ソケット
	GPU	NVIDIA Tesla V100 (Volta) SXM2, 2,560 FP64コア, up to 1,530 MHz × 4ソケット
	メモリ	メインメモリ(DDR4 2933 MHz):384 GiB (32 GiB × 6 枚 × 2 ソケット) デバイスメモリ(HBM2):32 GiB × 4 ソケット
	理論演算性能	倍精度 33.888 TFLOPS (CPU 1.344 TFLOPS × 2 ソケット, GPU 7.8 TFLOPS × 4 ソケット)
	メモリバンド幅	メインメモリ 281.5 GB/s (23.464 GB/s × 6 枚 × 2 ソケット) デバイスメモリ 900 GB/s × 4 ソケット
	GPU間接続	NVLINK2 (1GPUから他の3GPUに対してそれぞれ50GB/s×双方向)
	CPU-GPU間接続	PCI-Express 3.0 (x16)
ノード数、総コア数		221ノード、8,840 CPUコア + 2,263,040 FP64 GPUコア
総理論演算性能		7.489 PFLOPS (CPU 0.594 PFLOPS, GPU 6.895 PFLOPS)
総メモリ容量		メインメモリ 82.875 TiB、デバイスメモリ 28.288 TiB
ノード間インターコネクト		InfiniBand EDR 100 Gbps × 2, 200 Gbps
ユーザ用 ローカルストレージ		NVMe SSD 6.4TB, 一部ノードにて BeeGFS/BeeOND/NVMesh (ローカルストレージを使用した共有ファイルシステム) を提供
冷却方式		水冷

- データサイエンス研究、機械学習用のGPUクラスタ型
- 最新GPU (Volta) 4台／ノード
- 充実したAIツール
- 高速SSDローカルディスク

ノード内構成



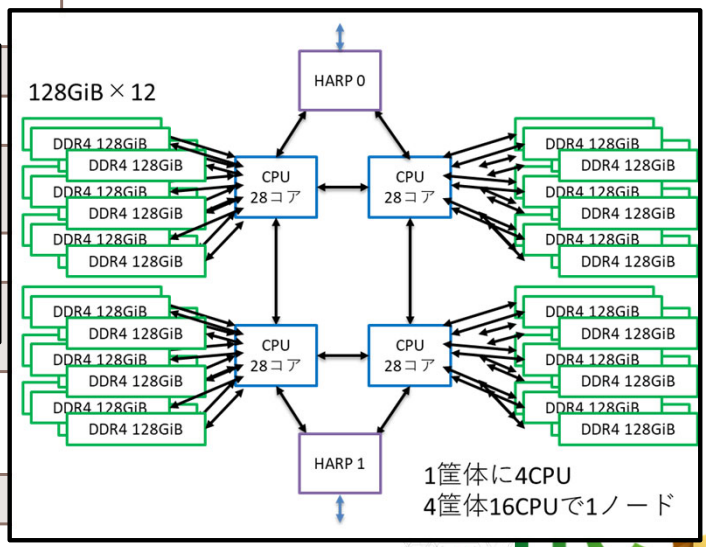
名古屋大学 NAGOYA UNIVERSITY

各サブシステムの仕様と特徴：Type III サブシステム

機種名		HPE Superdome Flex
計算ノード	CPU	Intel Xeon Platinum 8280M, 28コア, 2.70 - 4.00 GHz × 16 ソケット
	GPU	NVIDIA Quadro RTX6000 × 4
	メモリ	メインメモリ(DDR4 2933 MHz) : 24 TiB (128 GiB × 12枚 × 16ソケット) デバイスメモリ(GDDR6) : 24 GiB × 4
	理論演算性能	倍精度 38.7072 TFLOPS (CPU 2419.2 TFLOPS × 16 ソケット)
	メモリバンド幅	メインメモリ 2252.544 GB/s (23.464 GB/s × 12枚(6チャネル) × 16ソケット)
	CPU-GPU間接続	PCI-Express 3.0 (x16)
ノード数		2
総理論演算性能		77.414 TFLOPS (38.7072 TFLOPS × 2 ノード)
総メインメモリ容量		48 TiB
ノード間インターコネクト		InfiniBand EDR 100 Gbps
ユーザ用 ローカルストレージ		一方のノードに102.4 TB SSD、 もう一方のノードに1008 TB 共有ストレージを接続
冷却方式		空冷

- 大規模共有メモリ(24TiB)
- プリポスト処理用・可視化処理用
- NICE DCVを用いたリモート可視化

ノード内構成

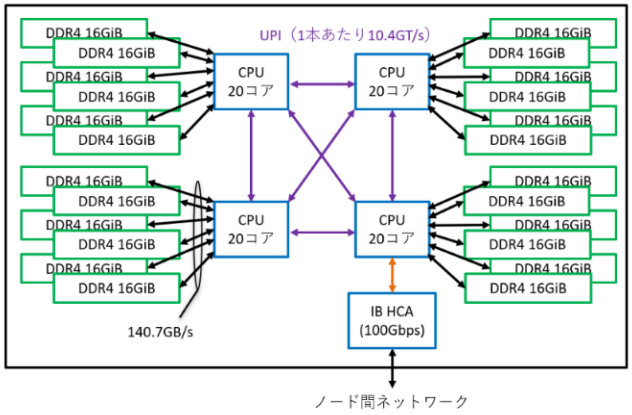


各サブシステムの仕様と特徴：クラウドシステム

機種名		HPE ProLiant DL560
計算ノード	CPU	Intel Xeon Gold 6230, 20コア, 2.10 - 3.90 GHz × 4 ソケット
	メモリ	メインメモリ(DDR4 2933 MHz) 384 GiB (16 GiB × 6 枚 × 4 ソケット)
	理論演算性能	倍精度 5.376 TFLOPS (1.344 TFLOPS × 4 ソケット)
	メモリバンド幅	メインメモリ 563.136 GB/s (23.464 GB/s × 6枚 × 4 ソケット)
ノード数		100
総理論演算性能		537.6 TFLOPS (5.376 TFLOPS × 100 ノード)
総メインメモリ容量		37.5 TiB
ノード間インターコネクト		InfiniBand EDR 100 Gbps
ユーザ用ローカルストレージ		なし
冷却方式		空冷

- 研究室クラスタから移行しやすいIntel CPU搭載システム
- 高いノードあたりCPU性能（4ソケット）
- 時刻を指定してのバッチジョブ・インタラクティブ利用が可能

ノード内構成



ホットストレージの仕様と特徴

メタデータサーバ(MDS)	
機種名	FUJITSU PRIMERGY RX2540 M5
CPU	Intel Xeon Gold 5222 (3.80GHz, 4コア) × 2
メインメモリ	DDR4 192 GiB
HDD	SAS 900 GB 10krpm × 2 (RAID1)
Interconnect	InfiniBand EDR × 2
SAN	FibreChannel 32 Gbps &tiems; 2
OS	RedHat Enterprise Linux
ノード数	4台
メタデータストレージサーバ(MDT)	
機種名	FUJITSU ETERNUS AF250 S2
SSD	RAID1+0 [4D+4M] × 2 + 2HS RAID1+0 [3D+3M] × 1 + 2HS
ノード数	1台

データストレージ(OSS/OST)	
機種名	DDN SFA18KE × 1台 DDN SS9012 × 10 台
HDD	NL-SAS 14TB 7.2krpm × 730、RAID6 [8D+2P] 30 Device × 24 DCR Pool + 10HS
Interconnect	InfiniBand EDR × 8
搭載セット数	4
総容量	
物理容量	40.32 PB (Global Spareを除く)
実効容量	約30.44PB

- HDD RAID
- 大容量：30.44 PB（実効容量）
- 超高速アクセス性能：384 GB/s

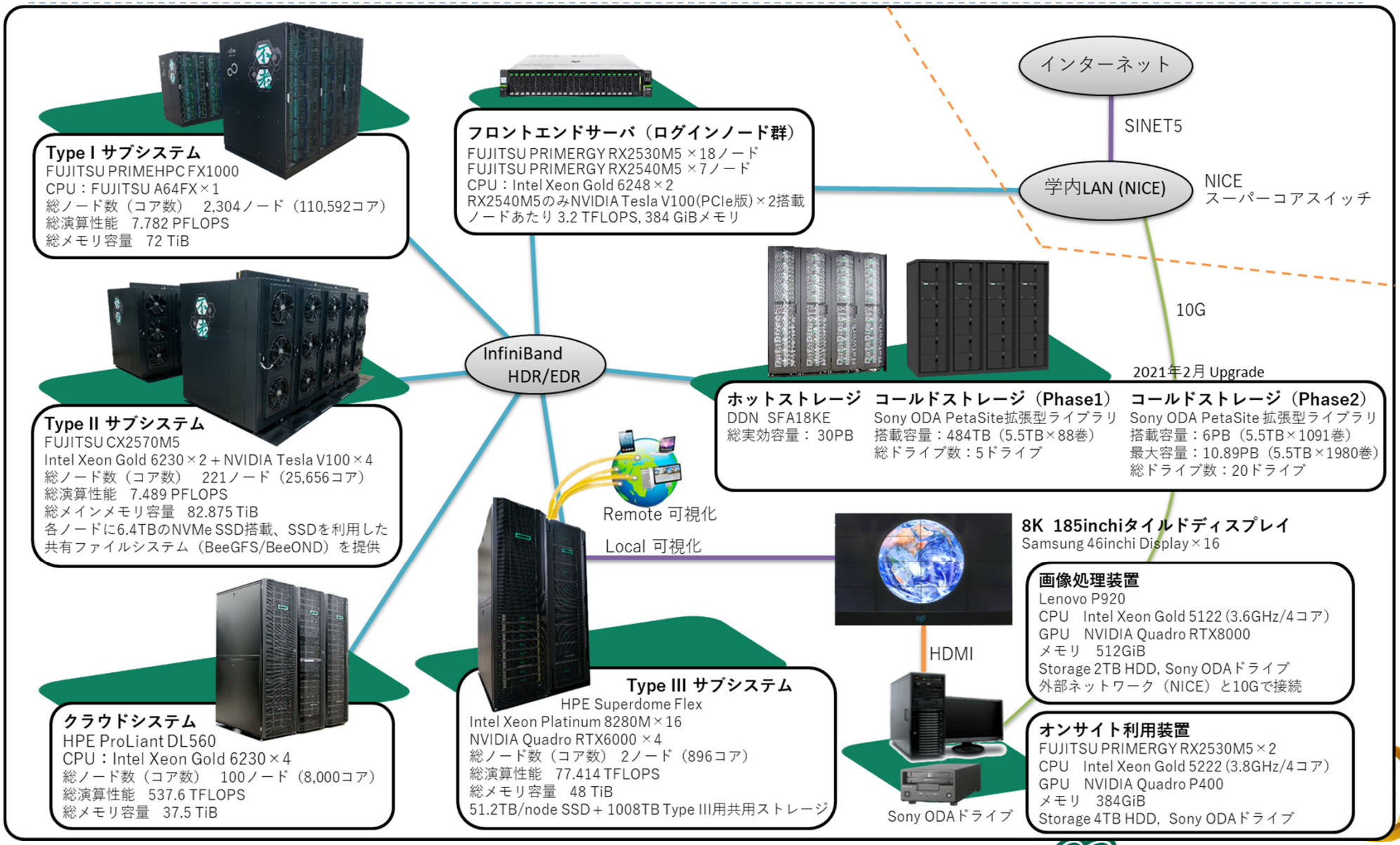
コールドストレージの仕様と特徴

フェーズ2: 2021年2月1日より稼働

機種名	PetaSite拡張型 Library
総スロット数 (最大搭載可能カートリッジ数)	1,980巻
総物理容量 / 最大搭載可能容量	**6 PB** / 10.89 PB
総ドライブ数	20
ODAサーバ数	4

- 6PBの大容量（スパコンで初）
- １度書き込み（追記）のみの光ディスクストレージ
- 実験データ等の長期データ保存用
- 理論上100年データ保持可能
- 水にぬれても読み出せる
- サービス終了後ユーザに光ディスクを返却

全体システム構成


インターネット
SINET5
学内LAN (NICE)
NICE
スーパーコアスイッチ
10G
2021年2月 Upgrade
InfiniBand
HDR/EDR
Type I サブシステム
FUJITSU PRIMEHPC FX1000
CPU：FUJITSU A64FX×1
総ノード数（コア数） 2,304ノード（110,592コア）
総演算性能 7.782 PFLOPS
総メモリ容量 72 TiB
フロントエンドサーバ（ログインノード群）
FUJITSU PRIMERGY RX2530M5 ×18ノード
FUJITSU PRIMERGY RX2540M5 ×7ノード
CPU：Intel Xeon Gold 6248×2
RX2540M5のみNVIDIA Tesla V100(PCIe版)×2搭載
ノードあたり 3.2 TFLOPS, 384 GiBメモリ
Type II サブシステム
FUJITSU CX2570M5
Intel Xeon Gold 6230×2＋NVIDIA Tesla V100×4
総ノード数（コア数） 221ノード（25,656コア）
総演算性能 7.489 PFLOPS
総メインメモリ容量 82.875 TiB
各ノードに6.4TBのNVMe SSD搭載、SSDを利用した
共有ファイルシステム（BeeGFS/BeeOND）を提供
ホットストレージ
DDN SFA18KE
総実効容量： 30PB
コールドストレージ（Phase1）
Sony ODA PetaSite拡張型ライブラリ
搭載容量：484TB（5.5TB×88巻）
総ドライブ数：5ドライブ
コールドストレージ（Phase2）
Sony ODA PetaSite 拡張型ライブラリ
搭載容量：6PB（5.5TB×1091巻）
最大容量：10.89PB（5.5TB×1980巻）
総ドライブ数：20ドライブ
Remote 可視化
Local 可視化
8K 185inchiタイルドディスプレイ
Samsung 46inchi Display×16
画像処理装置
Lenovo P920
CPU Intel Xeon Gold 5122 (3.6GHz/4コア)
GPU NVIDIA Quadro RTX8000
メモリ 512GiB
Storage 2TB HDD, Sony ODAドライブ
外部ネットワーク（NICE）と10Gで接続
HDMI
クラウドシステム
HPE ProLiant DL560
CPU：Intel Xeon Gold 6230×4
総ノード数（コア数） 100ノード（8,000コア）
総演算性能 537.6 TFLOPS
総メモリ容量 37.5 TiB
Type III サブシステム
HPE Superdome Flex
Intel Xeon Platinum 8280M×16
NVIDIA Quadro RTX6000 ×4
総ノード数（コア数） 2ノード（896コア）
総演算性能 77.414 TFLOPS
総メモリ容量 48 TiB
51.2TB/node SSD＋1008TB Type III用共用ストレージ
Sony ODAドライブ
オンサイト利用装置
FUJITSU PRIMERGY RX2530M5×2
CPU Intel Xeon Gold 5222 (3.8GHz/4コア)
GPU NVIDIA Quadro P400
メモリ 384GiB
Storage 4TB HDD, Sony ODAドライブ

高精細可視化システム

Type IIIサブシステム（HPE Superdome Flex）

77.4TFLOPS/48TiB MEM

Intel Xeon Platinum 8280M(2.7GHz,28Core)×16CPU×2　24TiB×2
NVIDIA Quadro RTX6000×4×2
HDD:実効容量500TB(RAID6)×2, NVMe:51.2TB×2

Quadro RTX6000

光ケーブル

SINET

リモート
可視化

大規模共有メモリ:
24TiB ×2ノード

8Kタイルドディスプレイ

185inchi　8K高精細大画面タイルドディスプレイ
（総解像度：7680×4320）
Samsung 46inchi Display×16

HDMI

画像処理装置（Windows可視化サーバ）

Intel Xeon Gold 5122
　　3.6GHz 4 Cores, 512 GiB MEM
NVIDIA Quadro RTX8000
SATA SSD 2TB
光Disk 5.5TBドライブ　ODS-D380U

スーパーコンピュータ共有ストレージ

ホットストレージ
DDN　SFA18KE
総実行容量：30PB

コールドストレージ
Sony ODA PetaSite 拡張型ライブラリ
搭載容量：6PB（5.5TB×1091巻）
最大容量：10.89PB（5.5TB×1980巻）

スーパーコンピュータ「不老」ソフトウェア利用環境

プログラム開発環境など

		フロントエンド システム	Type I サブシステム	Type II サブシステム	Type III サブシステム	クラウドシステム	画像処理 サーバ	オンサイト 利用装置	企業利用
Intel Parallel Studio Computing Suite	コンパイラ (Fortran, C/C++), プロファイラ/デバッガ, MPI, 数値計算ライブラリ	○		○	○	○			●
NVIDIA HPC SDK	コンパイラ (Fortran, C/C++, OpenACC, CUDA Fortran), プロファイラ/デバッガ, MPI, 数値計算ライブラリ	○		○					●
FUJITSU Technical Computing Suite	コンパイラ (Fortran,C/C++), プロファイラ/デバッガ, MPI, 数値計算ライブラリ	○	○						●
Arm Forge Proffesional	プロファイラ/デバッガ/最適化	○		○	○				●
NVIDIA CUDA SDK	GPU統合開発環境	○		○					●
Singularity	コンテナ環境	○		○					●
その他	GV, Gfortran, GCC, perl, Python, Ruby, R, Emacs, vi, nkf, etc.	○	○	○	○	○		○	●
	OpenGL	○		○	○	○		○	●

スーパーコンピュータ「不老」
ソフトウェア利用環境

ライブラリ利用環境

		フロントエンドシステム	Type I サブシステム	Type II サブシステム	Type III サブシステム	クラウド サブシステム	画像処理 サーバ	オンサイト 利用装置	企業利用
数値計算ライブラリ	FFTW, SuperLU, SuperLU MT, SuperLU DIST, METIS, MT-METIS, ParMETIS, Scotch, PT-Scotch, PETSc, MUMPUS, Xabclib ppOpen-HPCライブラリ：ppOpen-APPL, ppOpen-AT, ppOpen-MATH 精度保障ライブラリ：LINSYS_V, DHPMM_F	○	○	○	○	○			●
入出力フォーマットライブラリ	NetCDF, Parallel netCDF, HDF5, JHPCN-DF	○	○	○	○	○			●
画像処理ソフトウエア	OpenCV, Geant4	○	○	○	○	○			●
機械学習ソフトウエア	Caffe, Chainer, Keras, PyTorch, TensorFlow, Theano, Mxnet, ONNX パッケージ：conda, Numpy, Scipy, scikit-image, pillow, matplotlib, jupyterlab	○	○	○	○	○			●

スーパーコンピュータ「不老」
ソフトウェア利用環境

解析ソフトウェア利用環境

- *1 Type III サブシステムの会話型ノード(lm01)で利用可。
- *2 CPU並列版の他にGPU対応版が利用可。

		フロントエンド システム	Type I サブシステム	Type II サブシステム	Type III サブシステム	クラウド サブシステム	画像処理サーバ	オンサイト利用装置	企業利用
流体解析	OpenFOAM, FrontFlow blue/red		○	○	○	○			●
構造解析	LS-Dyna			○					
	FrontISTR		○	○	○	○			●
計算化学解析	AMBER		○	○ *2		○			
	Gaussian, Gamess, Gromacs, LAMMPS, NAMD		○	○ *2		○			●
	Modylas		○	○		○			●
メッシャー	Pointwise	○			○ *1				
統合ソフトウエア	HyperWorks			○ *2		○			

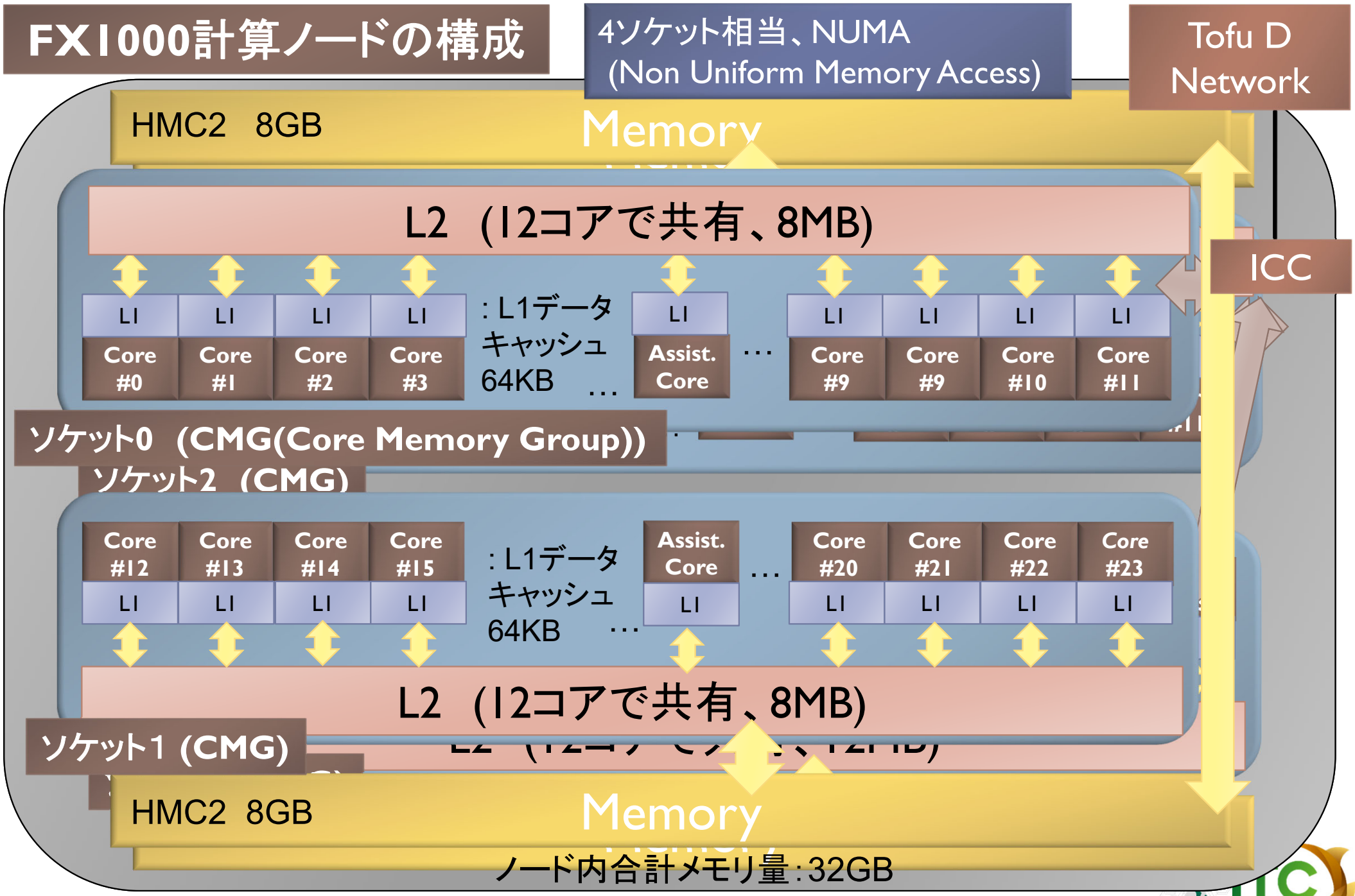
スーパーコンピュータ「不老」
ソフトウェア利用環境

可視化ソフトウェア利用環境

		フロントエンド システム	Type Ⅰ サブシステム	Type Ⅱ サブシステム	Type Ⅲ サブシステム	クラウド サブシステム	画像処理 サーバ	オンサイト 利用装置	企業利用
リモート可視化	NICEDCV	○			○※1				●
可視化 ソフトウエア	FieldView	○			○		○	○	×
	AVS/Express PyMOL VMD MeshLAB	○		○	○	○	○	○	●
	Paraview POV-Ray 3D AVS Player ffmpeg, ffplay	○			○※1		○	○	●
	IDL, ENVI	○			○		○	○	×
	MicroAVS						○		●
	3dsMax Visual Stsudio Pro						○		×
※1　Type Ⅲサブシステムの会話型ノード(lm01)で利用可。									

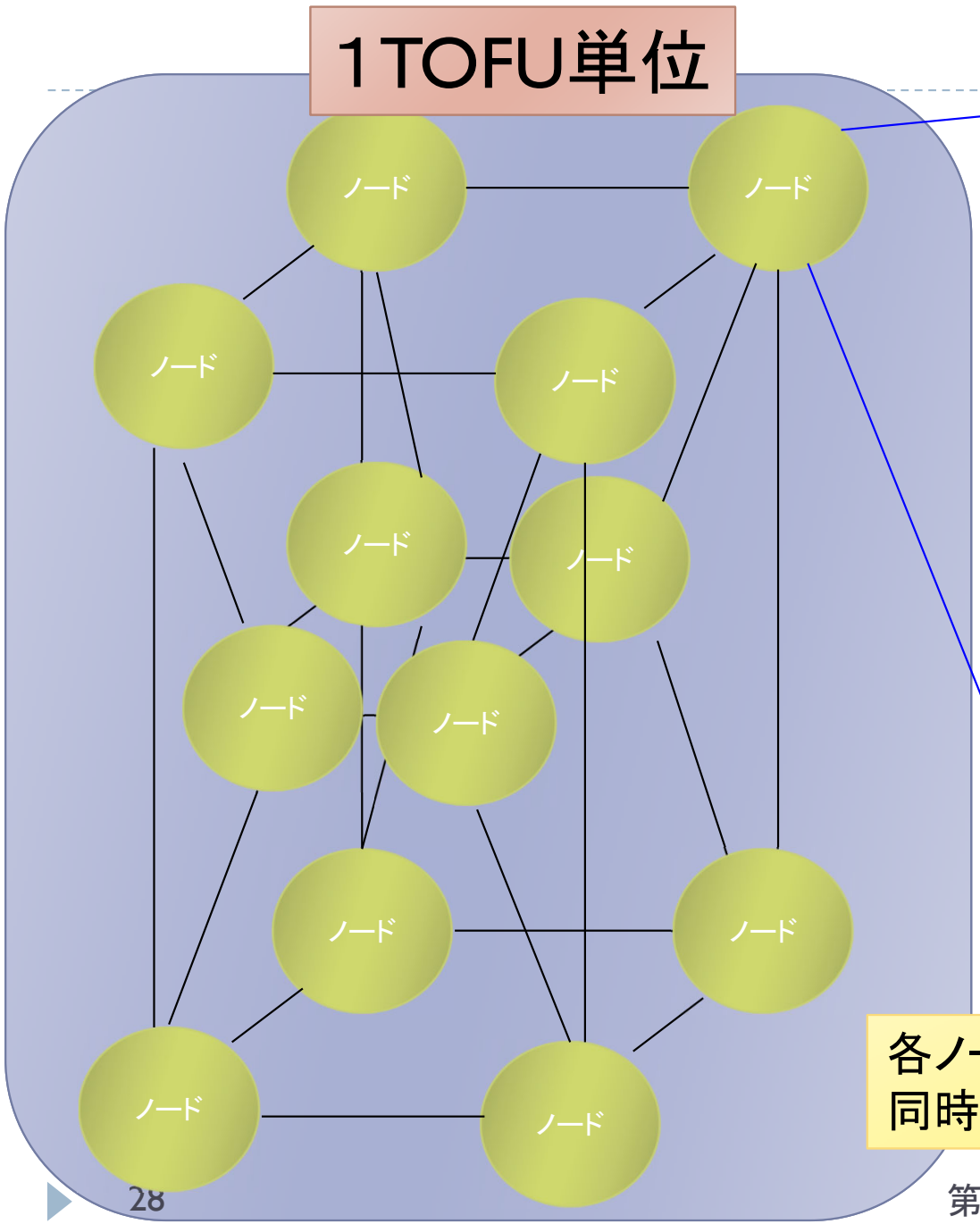
Fujitsu PRIMEHPC FX1000 の計算機アーキテクチャ

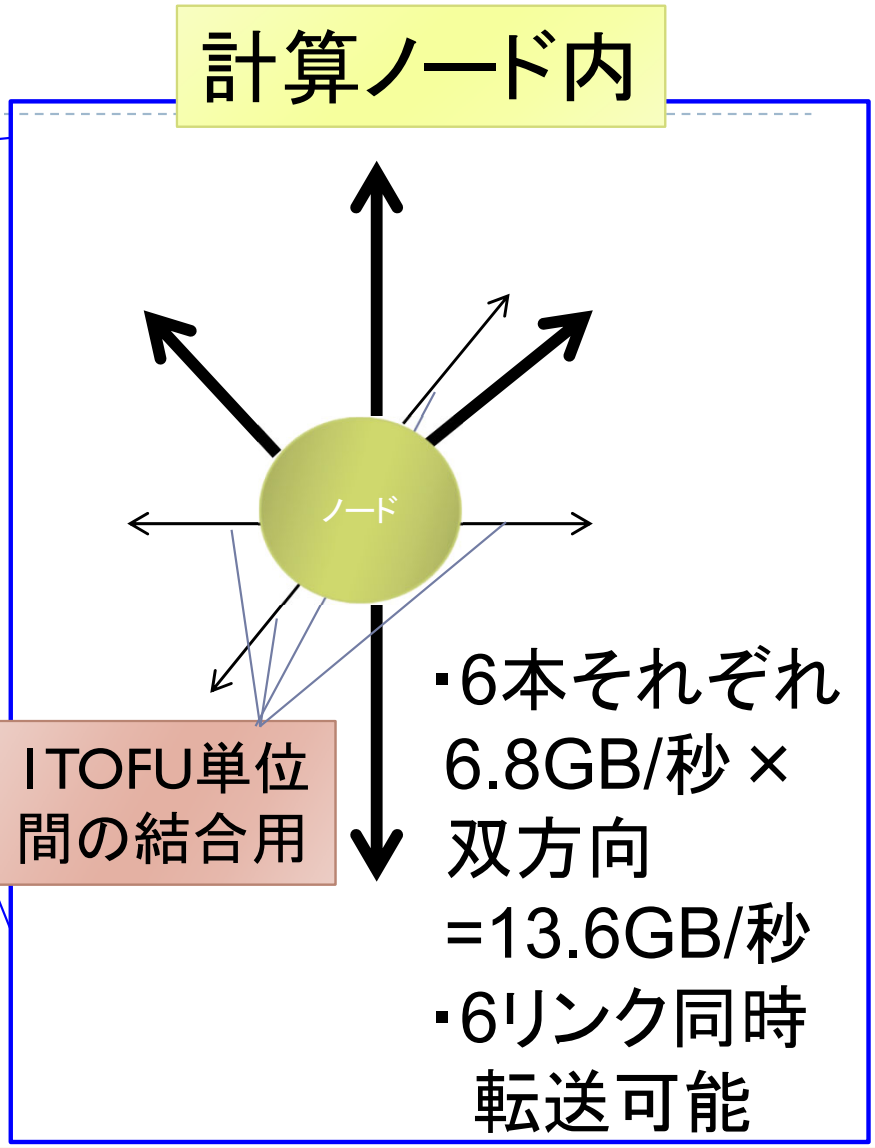
FX1000計算ノードの構成



読込み:512GB/秒
書込み:512GB/秒=合計:1024GB/秒

FX1000通信網：TofuインターコネクトD

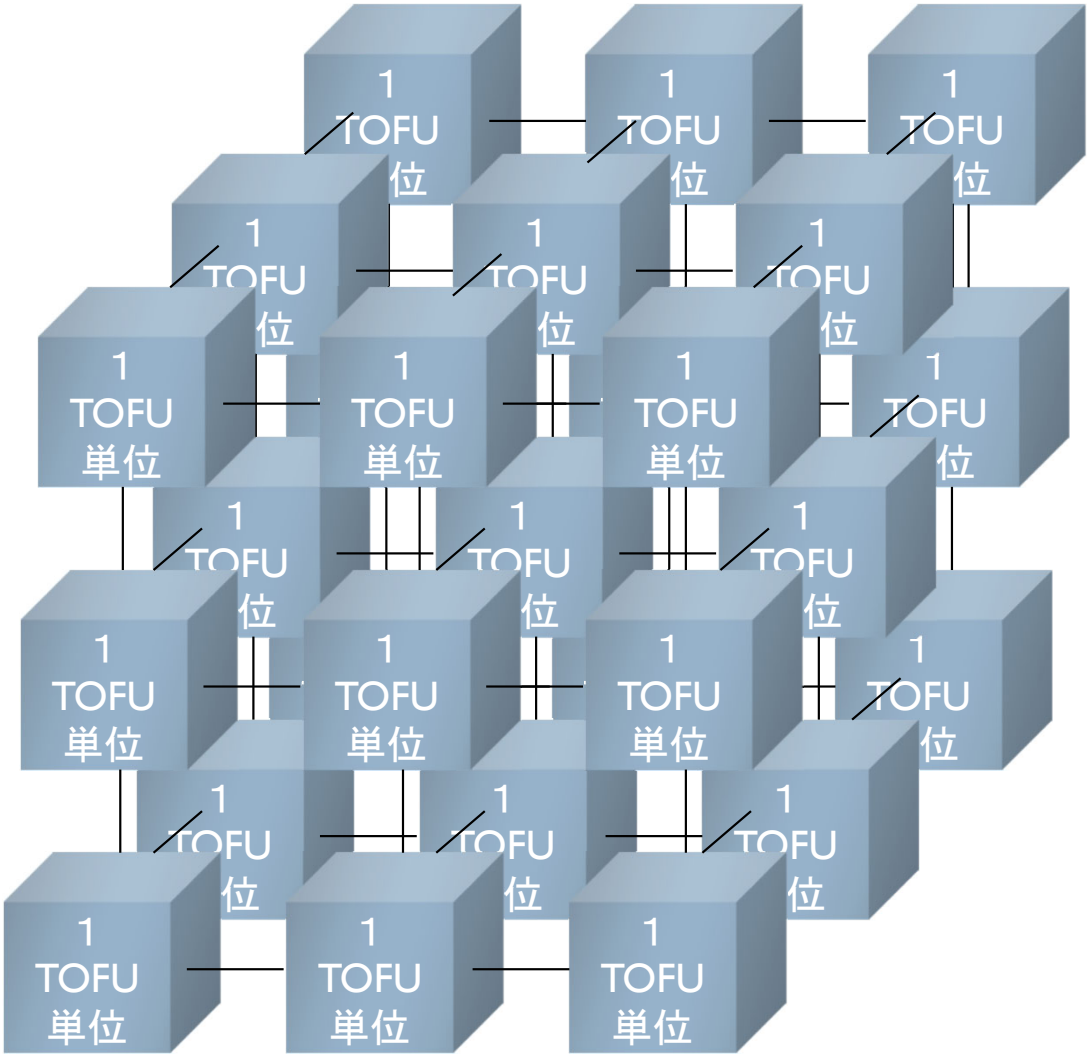




各ノードは周囲の隣接ノードへ
同時に合計 40.8 GB/s×双方向 で通信が可能

FX1000通信網：TofuインターコネクトD

3次元接続



- ユーザから見ると、
  X軸、Y軸、Z軸について、
  奥の1TOFUと、手前の
  1TOFUは、繋がってみえます
（3次元トーラス接続）
- ただし物理結線では
  - X軸はトーラス
  - Y軸はメッシュ
  - Z軸はメッシュまたは、
    トーラス

になっています

課金体系

スーパーコンピュータ「不老」課金体系 (1/8)

- 前払い定額制(プリペイド形式)
  - 利用すべき資源の料金を前払いして利用
  - 利用ポイントに変換して利用
- 単年度会計(4月1日～翌年3月31日)
  - 年度途中で申込み可能だが、利用終了は年度末
  - 年度末に余った利用ポイントは没収
- 一度の申込みで全てのサブシステムと可視化システムを利用可能
  - Type I、Type II、Type III、クラウド　全て共通

スーパーコンピュータ「不老」課金体系 (2/8)

- アカデミックユーザ（大学、研究機関など所属）向けプラン
  - 基本負担金
    - 利用登録1名につき年額10,000円（登録料）
    - 6,500 →**8,000**（予定） 利用ポイントを付加
      （他ユーザへの譲渡不可）
  - 追加負担金
    - 1,000円単位で追加が可能
    - 50万円未満： 1円あたり0.65 →**0.80**ポイント付加
    - 50万円以上： 1円あたり0.8125→**1.00**ポイント付加

スーパーコンピュータ「不老」課金体系 (3/8)

- TypeIサブシステム(「富岳」型ノード)消費ポイント
  - 計算課金:利用ノード数×経過時間[s] × 0.0056
    - 基本負担金1万円＝0.8万ポイント付加で利用可能な目安
      - 1ノードを　約16．8日
      - 4ノードを　約　3．9日
    - 10万円(基本利用料金1万円、追加料金9万円)＝8万ポイント付加で利用可能な目安
      - 1ノードを　約166日
      - 4ノードを　約　41日
      - 8ノードを　約　20日
    - 1ノードの年間利用額:約21万160円
      - 保守日等を考慮し年間350日利用できると仮定、以下同様

スーパーコンピュータ「不老」課金体系 (4/8)

- TypeIIサブシステム（GPUノード）消費ポイント
  - 計算課金：利用GPU数×経過時間[s] × 0.007
    - 基本負担金1万円
      ＝0．8万ポイント付加で利用可能な目安
      - 1ノード（1GPU）を　約13．6日
      - 1ノード（4GPU）を　約　3．1日
    - 10万円（基本利用料金1万円、追加料金9万円）
      ＝8万ポイント付加で利用可能な目安
      - 1ノード（1GPU）を　約132日
      - 1ノード（4GPU）を　約　33日
      - 4ノード（16GPU）を　約　8日
    - 1ノード（4GPU）の年間利用額：約104万4705円

スーパーコンピュータ「不老」課金体系 (5/8)

◆ Type III：1ソケット当たり28コア
◆ クラウド：1ソケット当たり20コア

- TypeIIIサブシステム（大規模共有メモリノード）、およびクラウドシステムの消費ポイント
  - 計算課金：利用ソケット数×経過時間[s] × 0.002
    - 基本負担金1万円
      ＝0．8万ポイント付加で利用可能な目安
      - 1ソケットを　約46．6日
      - 4ソケットを　約11．1日
    - 10万円（基本利用料金1万円、追加料金9万円）
      ＝8万ポイント付加で利用可能な目安
      - 4ソケットを　約116日
      - 32ソケットを　約14日
    - 2ソケットの年間利用額：約15万405円

スーパーコンピュータ「不老」課金体系 (6/8)

- 会話型利用：
  ログインノード上での処理の消費ポイント
  - 課金：無料
    - 主にインストール作業での利用想定です。
      計算利用はご遠慮ください。

  - Type IIIサブシステム（会話型処理）の課金は
    ありますのでご注意ください。
  - 各計算ノードの備えるinteractiveジョブクラスは
    バッチジョブ扱いの課金です
    （fx-interactive, cx-interactive, cl-interactive）

スーパーコンピュータ「不老」課金体系 (7/8)

- ホットストレージ
  - ファイル課金
    - 1TB 以下の場合(Home + Large) : 徴収しない
    - ファイルの使用容量が1TB を超えた場合 :
      超えた容量について、1GB につき 1日当たり 0.01 ポイント
    - 例) 2TB(2000GB)利用 : 1000GBが課金対象
      ⇒ 10ポイント/日　⇒300円/30日、3，500円/350日
      （保守などで停止する日については徴収しない）

※128TBを超える場合は、全体容量を考慮して、削除依頼をさせていただくことがあります【予定】。
※128TBを超える容量が必要な場合は、事前に相談ください。

スーパーコンピュータ「不老」課金体系 (8/8)

- コールドストレージ
  - ファイル課金
    - **1口：50TB**
      - 1回だけ書き込める（追記可能）の光ディスク×10枚（1枚約5TB）
    - ファイル負担経費（初回利用時のみ必要）：
      1口　190,000円
    - ファイル管理経費（毎年必要、基本負担金とは別）：
      1口　10,000円

●事前納入の光ディスク６PB売り切り後は、ユーザによる持ち込みのみなります（上限１０PBまで）。
●管理料１万円／年と格安です！
すでに、３PB程度が売れています！
●ご購入はお早めに！

※ユーザの利用終了時、もしくは、「不老」運用終了時に、光ディスクを持ち帰りいただけます。

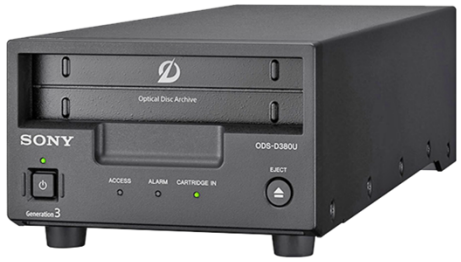
コールドストレージ　サービス紹介

2021 年 2 月より光ディスクを使ったコールドストレージサービスを開始していますが、**2021 年 5 月に光ディスクカートリッジが利用できる単体ドライブの貸し出しサービスも開始しました。**

コールドストレージ内部

Windows/Mac OSを搭載したPCに、USB3.2又はUSB3.0で単体ドライブを接続して、専用のFilerソフトを使って利用できます。

光ディスクカートリッジ



単体ドライブ

輸送には専用のジェラルミンケースをご用意しています。

輸送用ジェラルミンケース

コールドストレージシステム概要：

詳細は、以下をご覧ください：

https://icts.nagoya-u.ac.jp/ja/sc/pdf/pamphlet-cold-20201201.pdf

スーパーコンピュータ「不老」
新サービス：ノード準占有利用

- ノード準占有利用
  - 1時間以内のジョブ実行開始を保証
  - バッチ利用のみ
- 1ノード、1ヶ月間の利用負担金
  - Type IIサブシステム（GPUスパコン）：
    330,000円 → **45,000円**
  - クラウドシステム：
    95,000円

提供資源量が無くなり次第、受付終了
→お早めに

スーパーコンピュータ「不老」
新サービス：クラウドノード予約利用

- クラウドノード予約利用
  - 専用の予約システム「UNCAI」（Webブラウザで操作）で
    ノードを予約して利用する
- 利用料金
  - 計算課金：利用コア数×経過時間[s] × 0.0001
    （ソケット当たりコア数を考えればバッチ実行と同等）
    - 1ソケット当たり20コア、10コア（0.5ソケット）から
      利用可能
    - 利用可能なメモリ容量もコア数に比例
    - 基本負担金1万円でXeon Gold 20コア1ソケットを
      約46日間使用可能

スーパーコンピュータ「不老」
新サービス：グループ利用

売れています！

- グループ利用
  - 1口10人まで、10万円で80,000ポイント付与
  - 登録料なし
  - 個人利用（個別に1万円×10人が個別に基本負担金を払う）との違い
    - 80,000ポイントを10人で共有利用できます
    - 個人利用では、購入した8,000ポイントを他者と共有できません

お試し利用、リテラシー利用

- トライアルユース
  - ソフトウェアの動作確認などを、無料で行える制度です。
  - お1人様1回限りで申請できます。
  - 企業においては、同一の課で1回のみです。
  - アカデミックユーザ（無審査）、企業ユーザ（書類審査）
  - 10,000ポイント付加　（値上前と同じ）
  - 有効期限1ヶ月
- リテラシー利用（アカデミックユーザのみ）
  - 名大学内外の学部・大学院等の講義や演習で利用いただける制度です。
  - 利用登録25 件につき10,000 円、50,000ポイント付与（値上前と同じ）
  - 有効期限：上限6ヶ月（講義・演習実施期間に依存）

スーパーコンピュータ「不老」民間利用制度（産業利用）

- 書類での審査があります。追加負担金も同額です。
- 公開型
  - 10アカウントまで20万円
  - アカデミック利用の料金の2倍（20万円当たり80,000ポイント）
  - 企業名、課題名、報告書をWebで公開（延期制度あり）
- 非公開型
  - 10アカウントまで40万円
  - アカデミック利用の料金の4倍（40万円当たり80,000ポイント）
  - 外部に情報は非公開（ただし内部会議では情報が出ます）

申込み金額に応じたポイント優遇はございません。
詳しくは、産業利用のパンフレットをご参照ください。

優先ジョブクラス（アカデミック・民間）

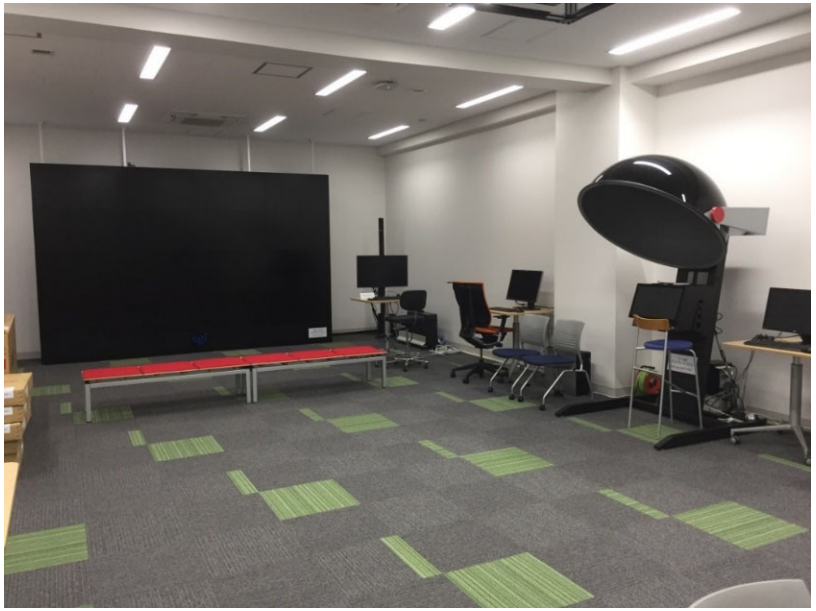
- Typel、Typell、クラウドの各サブシステム
  - ポイントを通常の2倍消費することで利用可能なジョブクラス
  - 専用のキューにジョブを投げることで利用
  - 通常のジョブクラスが混んでいるときでも早く実行したい、
    というユーザの利用を想定
  - （優先ジョブクラスも混んでしまったらすいません）

コンサルティング

- 並列化、利用高度化、ISVアプリの利用方法などに関するコンサルティングを行っています。
- 本センター教職員や学内外の専門家で構成される専門分野相談員によるコンサルティング（面談）ができます。
  - Web受付 Q&A SYSTEM
    - 各種ご質問、ご相談等は下記Webサイトからお問合せください。
    - https://qa.icts.nagoya-u.ac.jp/
  - 面談相談（※ZOOMによる遠隔面談も可）
    - 実際に画面を見ながらなど、電話やメールでは伝えにくいご質問やご相談は面談でも受け付けています。
    - 事前にお約束の上、本センター3階図書室内のIT相談コーナーにお越しください。または、相談員が訪問させていただくことも可能です。
      - 連絡先：052-789-4366（IT相談コーナー直通）、または上記のQ&A SYSTEMから

可視化設備

- 情報基盤センター可視化室（本館1階）
  - 可視化室の利用は予約制となります（無料）

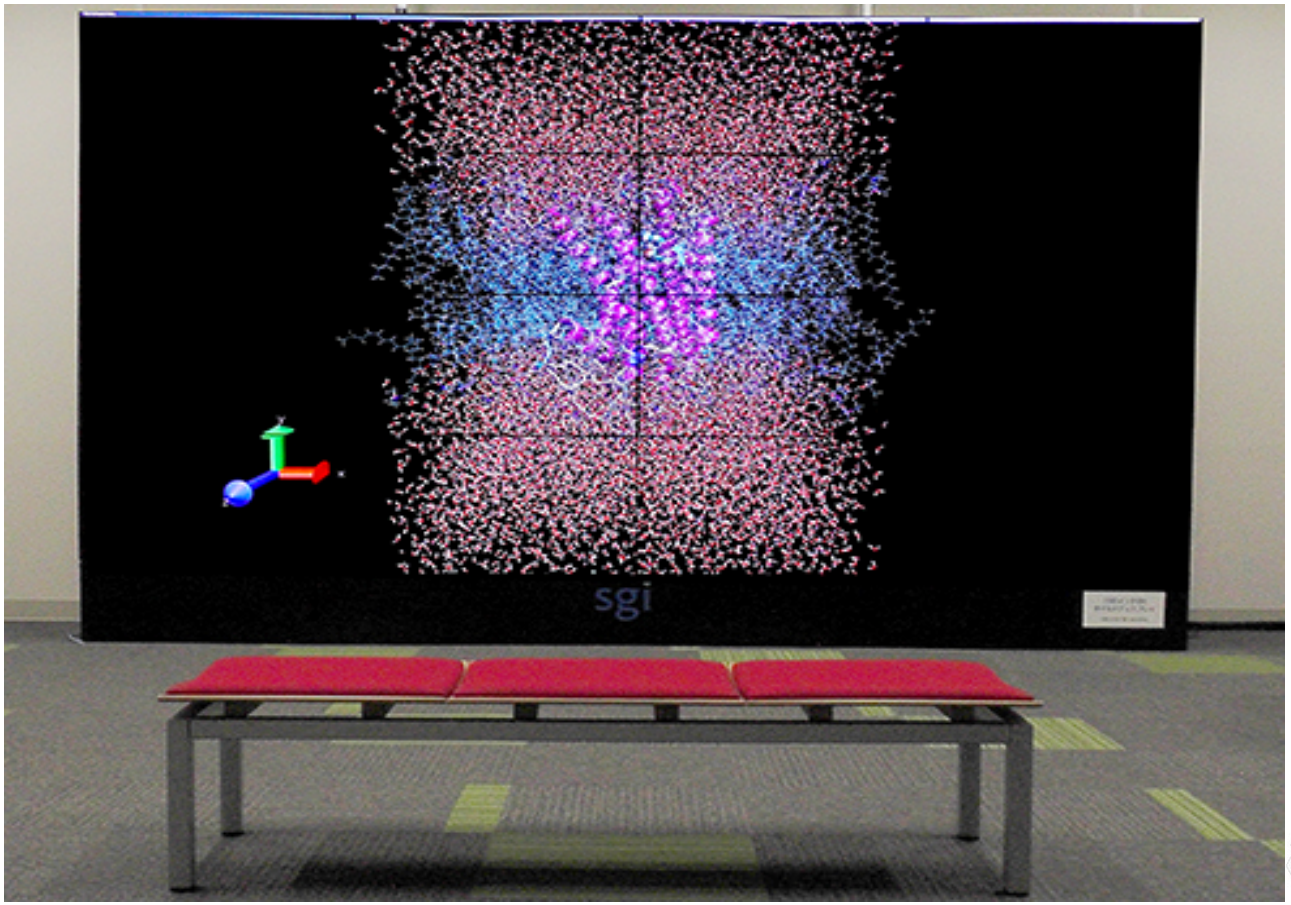


- 8K 185型タイルドディスプレイ
- 全天周映像視聴システム
- 円偏光立体視システム　など

可視化設備

- 情報基盤センター可視化室（本館1階）
  - 可視化室の利用は予約制となります（無料）

8K 185型
タイルド
ディスプレイ



利用者支援室（２部屋）

- 情報基盤センター利用者支援室（本館3階）
  - 利用者支援室の利用は予約制となります（無料）

訪問者が現地で利用可能な機材

画像処理装置（1F可視化室）

- Windows10が動作するデスクトップパソコン（1台）
- 対話型のプリ・ポスト処理や共有ファイルストレージの大容量ファイル取り扱い用
- 10Gbps SINETによる高速インターネットが利用可能
- USB外付けHDDやディスクメディア（Blu-rayディスク）を持ち込んでホットストレージに対するデータの読み書きが可能
- 可視化室に設置された185インチ8K高精細ディスプレイに接続、大規模データの高品位なプリポスト処理やコンテンツ生成に活用可能

オンサイト利用装置（3F利用者支援室）

- Linuxが動作する計算サーバ（2台）
- 対話型のプリ・ポスト処理、スーパーコンピュータシステムとの大容量ファイル取り扱い用
- USB外付けHDDや光ディスクメディア（Blu-rayディスク）を持ち込んでホットストレージに対するデータの読み書きが可能
- 予約制の部屋貸し切りで、ハードディスク持ち込みで大規模データの入力、回収が可能

画像処理装置

ODA単体ドライブ

- コールドストレージの光ディスクアーカイブと互換性のあるUSB接続ドライブ
- 画像処理装置・オンサイト利用装置に接続して利用可能

オンサイト利用装置

- ODA単体ドライブの貸し出しできます（宅配便で郵送できる専用ケースも有）
- ご相談ください

スーパーコンピュータ「不老」講習会等 開催情報（2025年9月～）

2025年度「不老」利用型講習会 開催予定（オンライン開催）

- 9月12日：MPI講習会(初級)
- 9月19日：OpenMP講習会(初級)
- （新設）【業界初】10月22日：分散機械学習講習会
  - TensorFlow、PyTorch、Megatron-LMなどによる中～大規模な機械学習の講習会です

- 2025年10月以降も、多数開催予定
- 最新状況は以下のHPをご覧ください

https://www2.itc.nagoya-u.ac.jp/cgi-bin/kousyu/csview2.cgi

「不老NEXT」(仮称)導入予定について

スパコン「不老」の運用期間について

- 導入当初は、2024年末（～2025年3月末）で運用停止の予定でした
- 現在、1年の運用延長中です
  - 延長期間：2025年4月1日～2026年3月末
- 次期システム（「不老NEXT」（仮称））調達中です
  - 「不老NEXT」稼働（予定）2026年10月1日～
  - 特徴
    1. データセントリック拠点（データ駆動型研究）支援
    2. 最新GPUを搭載
    3. 「不老」の10倍以上のAI性能の実現を目指す

名古屋大学情報基盤センターのスパコンの歴史

スーパーコンピュータ「不老」導入

96 00 05 09 13 15 20 24

Fujitsu M-1800 → Fujitsu GP7000F/90 → Fujitsu HPC2500 → Fujitsu HX600, M9000 → Fujitsu CX400 → CX400 → 複合型「不老」

メインフレーム系


スカラー型スパコン

アプリケーションサーバ


Fujitsu VPP500 → Fujitsu VPP5000 → Fujitsu FX1 → Fujitsu FX10 → FX100

ベクトル型スパコン


スーパーコンピュータ



複合型「不老」

- Fujitsu CX2570 GPUサーバ
- Fujitsu FX1000 超並列
- HPE Super Dome Flex 大規模共有メモリ
- HPE ProLiant DL560 クラウド

→ 次期不老へ

- これまで約5年+1年延長間隔でリプレイス
- 「不老」も6年弱（6年9ヶ月）の稼働予定
- 「不老NEXT」(仮称)2026年10月1日稼働予定

将来の新サービスに向けて：
コード生成AI支援サービスと
HPC-GENIEプロジェクト

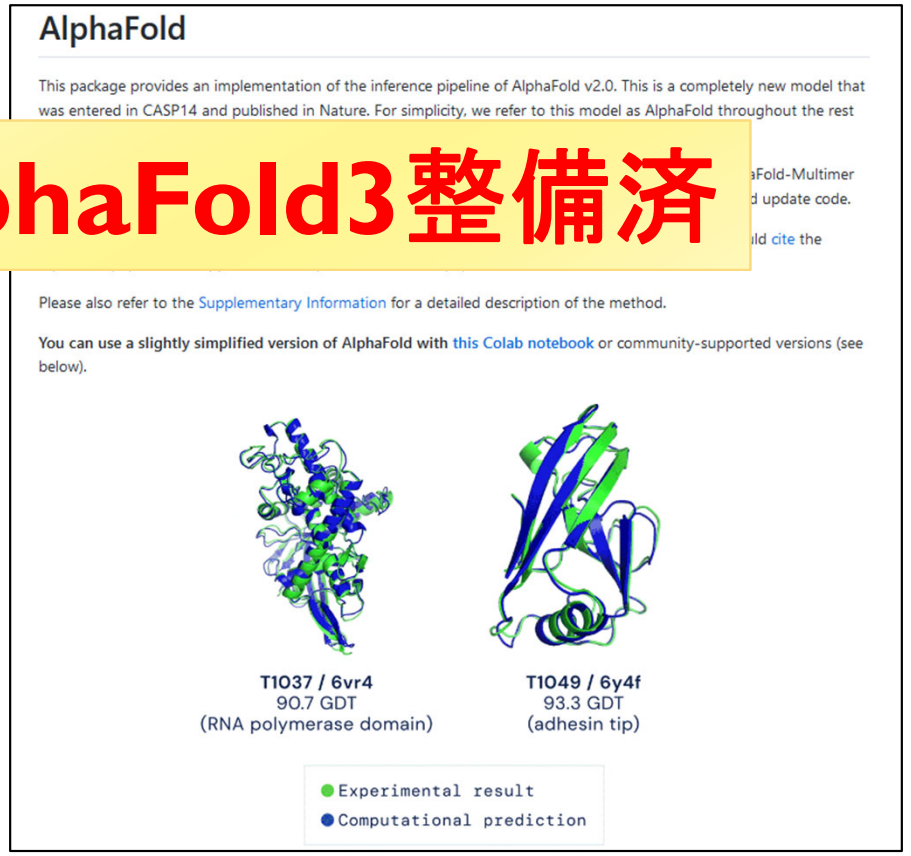
センター提供／ユーザ開発アプリ

OSS活用事例

AlphaFold3整備済

- タンパク質構造予測ソフト **AlphaFold2** をTypeⅡサブシステム上で簡単に利用できるように整備 (2022年2月2日)

https://icts.nagoya-u.ac.jp/ja/sc/news/maintenance/2022-01-28-alphafold.html

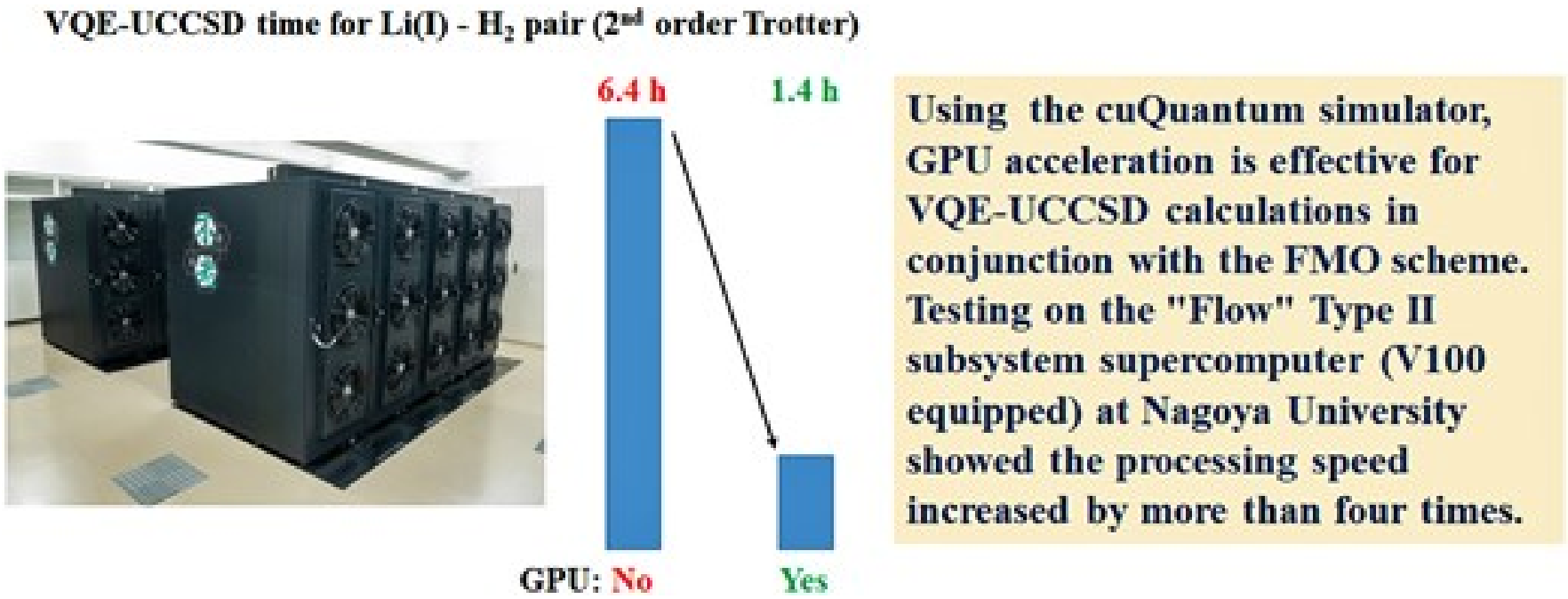


(source: https://github.com/deepmind/alphafold)

- AlphaFoldの利用規約が更新され民間利用のユーザも利用可能
- 分散ノードのSSDを利用可能とするNVMESHにデータベース配置
  ⇒10時間以上かかるHHblits処理が10分に短縮
- 1GPUキューが新設され、さらにお得に！
  https://icts.nagoya-u.ac.jp/ja/sc/news/general/2022-08-10-alphafold.html

量子関連計算：GPUによる回路シミュレーション

- 「不老」TypeIIサブシステムのGPUによる、計算化学の量子回路シミュレーション



引用：Hideo Doi, et al., Concurrent processing of VQE-UCCSD calculations with the FMO scheme Open Access, Chemistry Letters, Volume 54, Issue 8, August 2025, upaf145, https://doi.org/10.1093/chemle/upaf145 (22 July 2025)

第三部 「不老」ユーザからの講演
15:35-16:00 「VQE-UCCSD/FMO計算の『不老』Type IIでの同時並行処理」 / 望月祐志（立教大学 教授）

コード生成AIサービスに向けて

HPC-GENIEプロジェクトとは

HPC-GENIE Project HP:
https://www.hpc.itc.nagoya-u.ac.jp/menu/hpc_genie.html

- HPC-GENIE (High-Performance Computing with GEnerative Neural Intelligence for Execution)
- 名古屋大学 情報基盤センター/大学院情報学研究科の関連者で発足したコード生成AIを活用したHPCプログラム自動生成プロジェクト
- 大規模言語モデル(LLM)を活用したプロンプトエンジニアリングと、ソフトウェア自動チューニング(Software Auto-tuning, AT)技術を融合した自動化を行い、HPCソフトウェア開発の生産性を劇的に高めることを目的

HPC-GENEメンバ (2025年6月29日現在)

- 研究代表者:片桐 孝洋
  （名古屋大学 情報基盤センター 教授）
- 副代表者:椋木 大地
  （名古屋大学 情報基盤センター 助教）
- 構成員
  - 星野 哲也（名古屋大学 情報基盤センター 准教授）
  - 森崎 修司（名古屋大学 大学院情報学研究科 情報システム学専攻 准教授）
  - 大島 聡史（名古屋大学 情報基盤センター 招へい教員 / 九州大学 情報基盤研究開発センター 准教授）
  - 林 俊一郎（名古屋大学 大学院情報学研究科 修士1年）

我が国のAT研究(~2014)と LLMによるコード生成研究との比較

- AT研究会10年史(https://atrg.jp/ja/)と比較

項目	自動チューニング (~2014年)	LLMコード生成 (2023年~)
主な技術基盤	テンプレート生成＋探索 (遺伝的アルゴリズム/焼きなまし法/ベイズ最適化)	大規模言語モデル＋自然言語処理
入力形式	パラメタ付き コードテンプレート	自然言語/ソースコード/仕様文
探索手法	ヒューリスティクス・ 性能計測ベース	自己改善 (反復プロンプト)
出力精度	計測ベースの 最適コード選定	学習ベースの 確率的生成(ばらつき有)
再現性・安定性	高 (同一条件下で安定)	中~低 (プロンプト依存)
代表的手法	ATMathCoreLib, ppOpen-AT, ATMathCoreLib, etc.	GPT-4, Claude, Gemini, CodeX etc.

LLM従来方式 vs. HPC-GENIEの狙い

項目	LLM従来手法	HPC-GENIEの狙い
コード生成方法	汎用LLMに手動プロンプト入力／回答利用	**プロンプト工学＋CLIベースの複数LLM選択方式**
自動チューニング連携	CLIツール (OpenTuner, GPTune等)で手動接続	**Auto-Tuning統合型**(コード生成直後にAT走査)
RAG機能	実験的に一部LLMで統合(LangChainなど)	**RAG機構を明示的に開発対象として内包する**
精度保証／混合精度演算制御	演算精度は人手調整、混合精度演算は対応していない	**精度保証/混合精度演算を構成要素に明示的統合**を計画
説明可能AI (XAI)連携	ほぼ非対応（black-box出力）	**XAI要素**(実行時間/演算精度/電力などの予測の妥当性検証)**の統合**を計画
ローカルLLM対応	基本はクラウドLLM(グローバルLLM)前提(GPT, Claude 等)	Swallow LLMなどを**ローカルLLM対応を開発対象に含む(グローバルLLMも選択可能とする)**

反復プロンプト と AIエージェント

候補生成AG

- コード候補管理
  - ・コードID付
  - ・コード性能情報（実行時間、精度等）
  - ・ベンチマーク情報

指示スクリプト

LLM

候補情報

候補選択AG

- スクリプト文章化
- ベンチマーク指定
- コード候補の選択

指示スクリプト

LLM

候補コード

実験計画AG

- モデルカタログ/機能マップ
- 目的に応じた選択戦略

指示スクリプト

LLM

○○○となるコードを生成

・スクリプト修正
・反復プロンプト指示

コード生成AG

- One/Multi/Many-Shot/CoT実行
- RAGの活用
- 商用LLM/ローカルLLM利用選択
- 実行履歴蓄積
  - →生成コードの差分
- コードに関する文章化

○○○となるコードを生成

LLM

生成したコード

検証失敗情報

ベンチマーク/コード検証AG

- コンパイルチェック→文法検証
- 実行テスト→実行性能検証（演算精度検証）
- ベンチマーク→性能検証
- AT実行
  - →性能チューニング
- 実行性能ログ蓄積

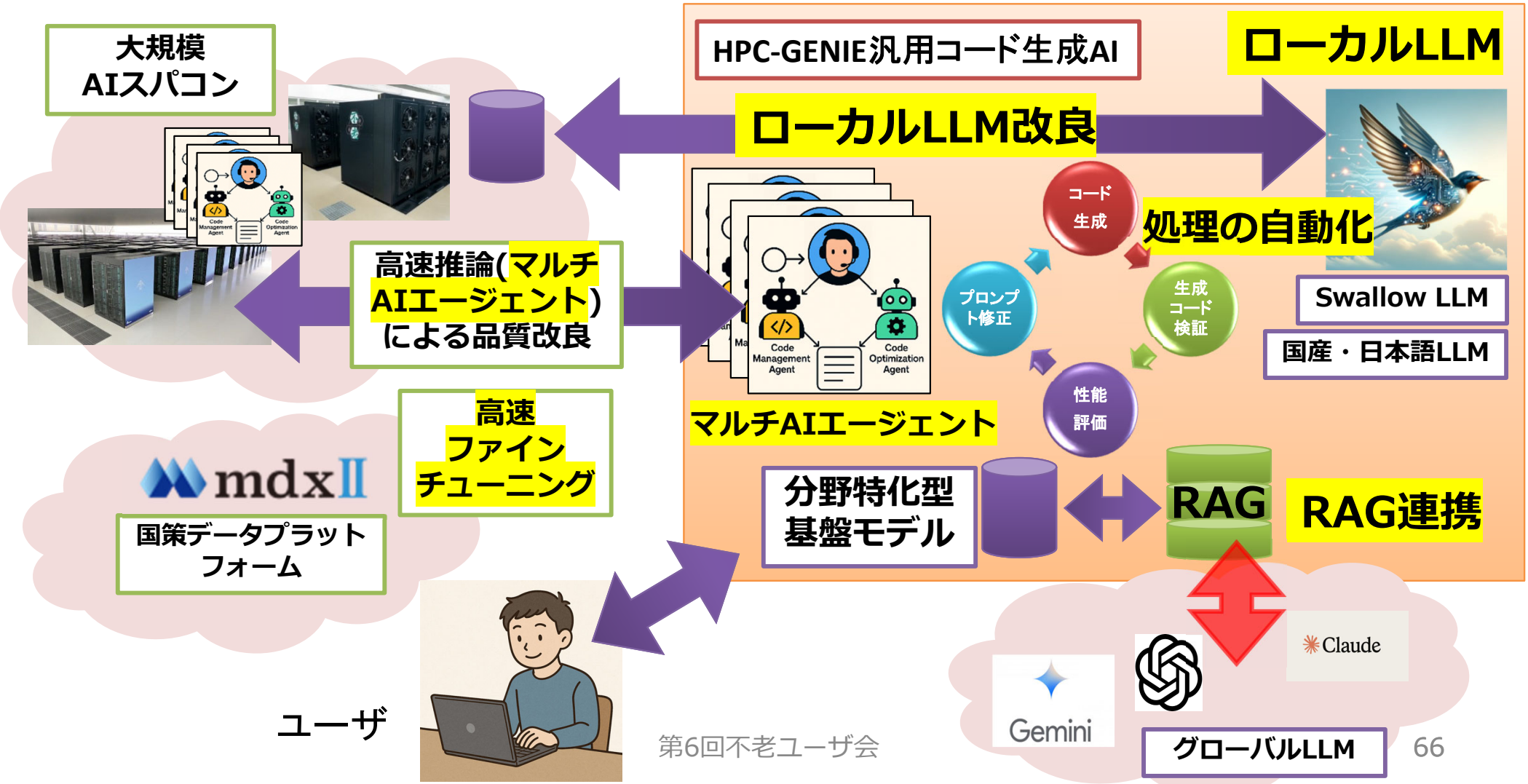
指示スクリプト

LLM

検証成功コード

HPC-GENIEによる ローカルLLM特化処理の自動化

- **処理の自動化**：コード生成→生成コード検証→性能評価→プロンプト修正の反復プロンプト化
  - マルチAIエージェント（A2A）によるコード品質の向上
- ローカルLLMを強化する**RAG**
- ローカルLLMを強化する**スパコンによる高速ファインチューニング/高速推論（マルチAIエージェント）実行による生成コード品質改善**

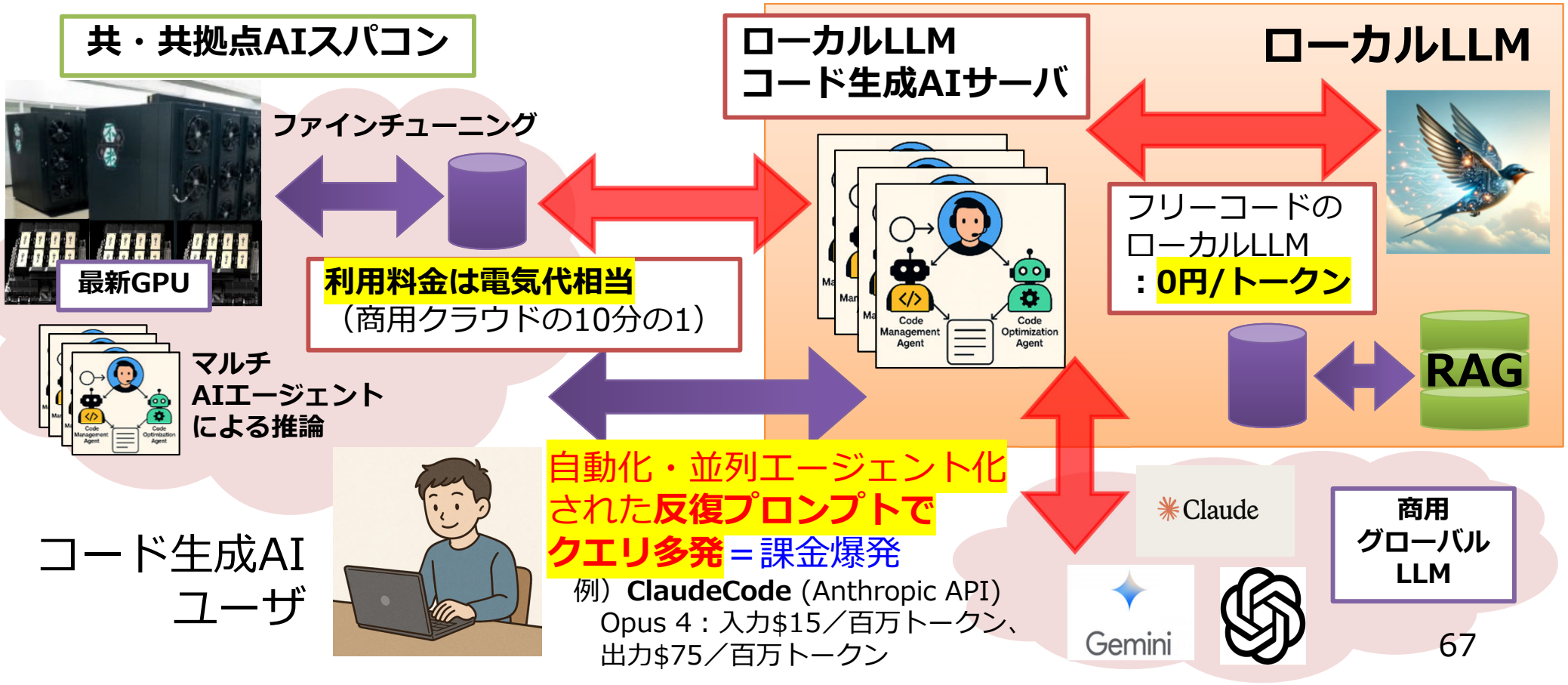


アカデミック利用のローカルLLM環境整備の必要性

アカデミア向け安価/高性能コード生成AI環境提供の必要性

- ローカルLLM利用による課金爆発
  - **商用グローバルLLM利用時：**自動化・並列エージェント化された反復プロンプトで課金爆発
  - フリーコードのローカルLLM利用は無料
- ファインチューニング／並列推論時の課金
  - 多数の最新GPUの利用が必須
  - 共同利用・共同研究拠点（共・共拠点）のAIスパコンの利用料金は電気代相当（商用クラウドの10分の1）

  →名大基盤センタ次期スパコン「不老NEXT」(仮称):2026年10月運用開始予定の新サービスとして提供予定



Vibe Coding とは

- バイブコーディングは、2025年にAndrej Karpathy [1]によって提唱された成果物を開発する新しい手法
- GitHub CopilotなどのLLM対応IDE、Claude CodeなどのCLIは、バイブコーディングワークフローの中核となるAIソフトとして使われている
- 従来のプログラミングワークフローとは異なり、バイブコーディングでは、直接的なコード作成を最小限に抑え、ユーザーが技術仕様よりも直感的な意図表現を優先する会話型フローの中で「何かを見て、何かを言って、何かを実行する」ことを可能にする[2]
  →簡単に「**プロトタイピング**」ができるコード開発体系

[1] A. Karpathy, “There’s a new kind of coding I call ’vibe coding’ …”,” X. Accessed: Jul. 23, 2025. [Online]. Available: https://x.com/karpathy/status/1886192184808149383

[2] C. Meske et al., “Vibe Coding as a Reconfiguration of Intent Mediation in Software Development: Definition, Implications, and Research Agenda”, arXiv:2507.21928 [cs.SE], (29 July 2025) https://doi.org/10.48550/arXiv.2507.21928

VibeCodeHPC

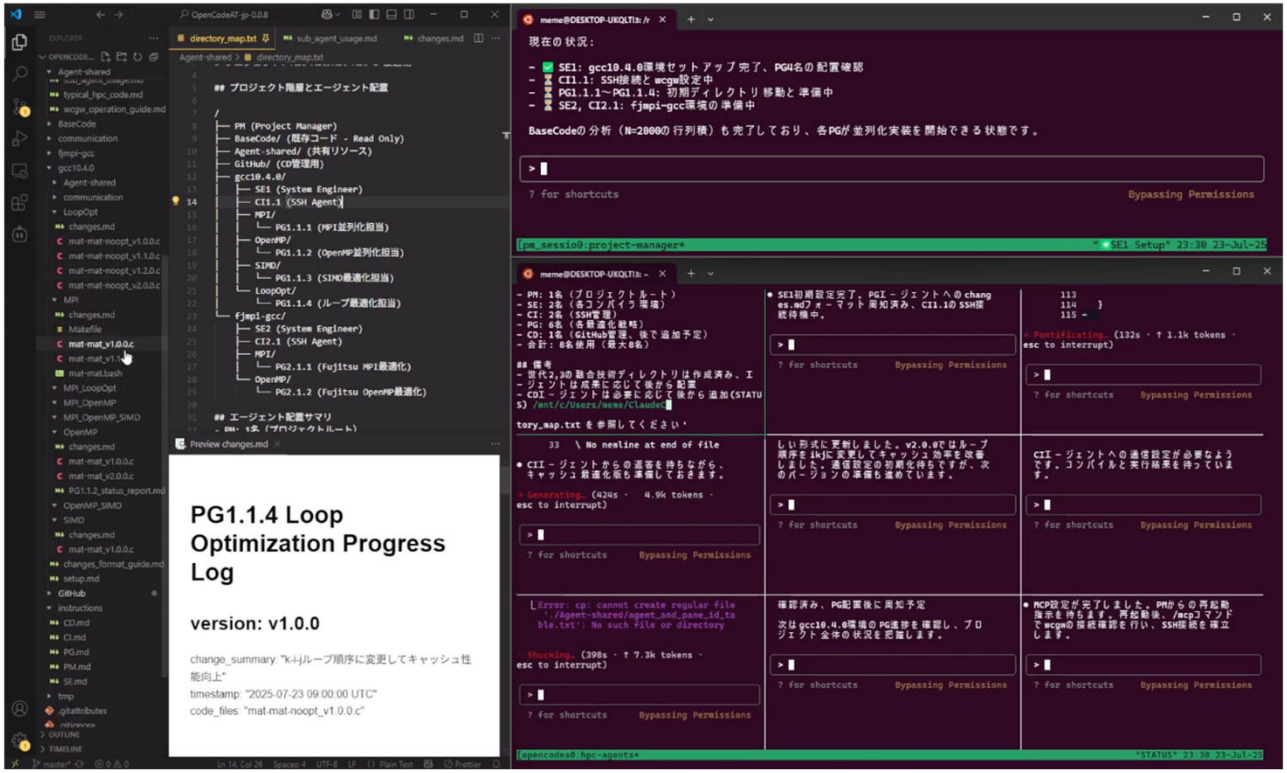
:HPCコード最適化のための反復プロンプトによるATのプロトタイプ

主開発者:名古屋大学 大学院情報学研究科 M1 林俊一郎

VibeCodeHPCの概要(1/3)

- **VibeCodeHPC**
  - Multi Agentic Vibe Coding for HPC
- 2025年7月28日：GitHubで公開

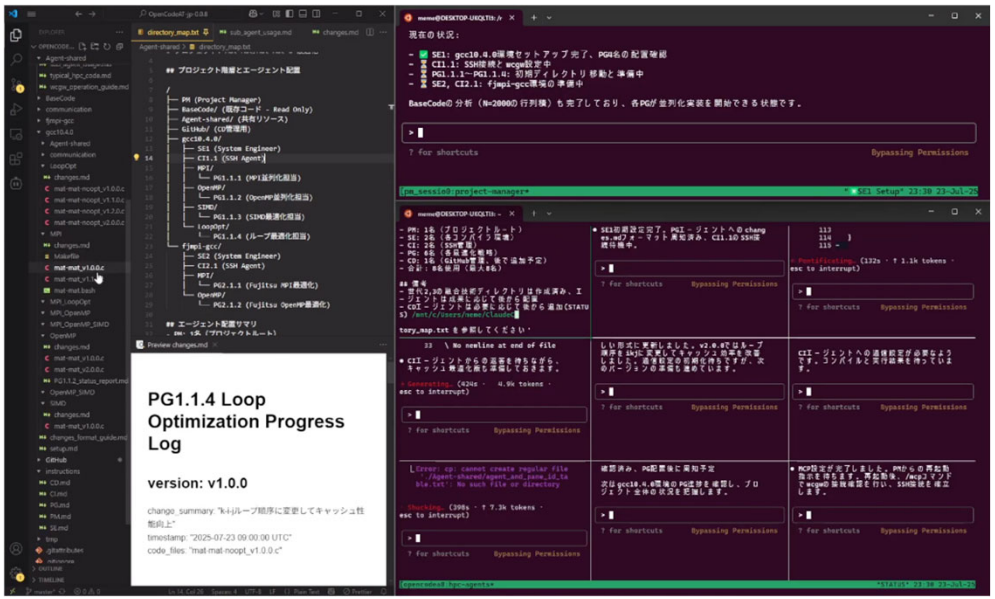
GitHub：
https://github.com/Katagiri-Hoshino-Lab/VibeCodeHPC-jp



https://github.com/Katagiri-Hoshino-Lab/VibeCodeHPC-jp

VibeCodeHPCの概要 (2/3)

- HPCコード最適化のための自動チューニングを行う Multi-Agent System



起動画面

Agent	役割	主要成果物	責任範囲
PM	プロジェクト統括	assign_history.txt User-shared/final_report.md	要件定義・リソース配分・予算管理
SE	システム設計	User-shared/reports/ User-shared/visualizations/	エージェント監視・統計分析・レポート生成
PG	コード生成・実行	ChangeLog.md sota_local.txt job_list_PG*.txt	並列化実装・SSH/SFTP接続・ジョブ実行・性能測定・SOTA判定
CD	デプロイ管理	GitHub/以下のprojectコピー	SOTA達成コード公開・匿名化

エージェントの役割例

https://github.com/Katagiri-Hoshino-Lab/VibeCodeHPC-jp

VibeCodeHPCの概要 (3/3)

入力されたHPCコード(Fortran等)から、OpenMP、MPI、OpenACC、CUDA等のコード生成を目指す

特徴

- **階層型マルチエージェント**: PM → SE ↔ PG の企業的分業体制
- **プロジェクト地図:** 組織をリアルタイムに視覚化するdirectory_pane_map
- **進化的探索:** ボトムアップ型のFlat📁構造による効率的探索
- **自動最適化:** OpenMP、MPI、OpenACC、CUDA...等の段階的並列化と技術融合
- **予算管理:** 計算資源💰の効率的配分と追跡
- **統一ログ:** ChangeLog.mdによる一元的な進捗管理

対応環境

- **スパコン**: 「不老」「富岳」等のHPCシステム
- **コンパイラ**: Intel OneAPI、GCC、NVIDIA HPC SDK、等

プロトタイプ実装版
（2025年7月28日版）

※シングルエージェント版

手元のPCのフォルダ構成案
(例：スパコン「不老」/Miyabi)

ユーザ指定のディレクトリ： /SSH 📁 **CLAUDE.md(SSH接続確立係)**

Markdown (プレーンテキストで書ける軽量マークアップ言語)でユーザーがコード生成のルールやコンテキストを、プロンプトで記述

＝現代のプログラミング

※さらにmdファイルの自動生成も、プロンプトで指示可能(→ノーコード)

```
ClaudeCodeHPCforXXX
|
|——**CLAUDE.md(CTO)**
|
|—— /OriginalCode 📁 /XXX
|
|—— /Flow ——📁
|                |—— /TypeI ——📁
|                |—— /TypeII          |——**CLAUDE.md(PM)**
|                |—— /TypeIII         |—— /FileAgent 📁— **CLAUDE.md(File転送係)**
|                                     |—— /CommandAgent 📁— **CLAUDE.md(cmd実行係)**
|                                     |—— /Workers ——📁
|                                                    |— **CLAUDE.md(Worker)**
|                                                    |— /single-node 📁
|—— /Miyabi —📁                                      |— /multi-nodes 📁
|                |—— /Miyabi-G ——📁
|                |—— /Miyabi-C ——📁
|                                     |——**CLAUDE.md(PM)**
|—— /Git 📁**CLAUDE.md(Git管理係)**    |—— /FileAgent
```

VibeCodeHPC(プロトタイプ)全体像 (スパコン「不老」利用時)

スパコン等 🔒

不老
ログインノード

【遠隔コマンド自動実行】
- scp でファイル転送
- make でコンパイル
- pjsub でジョブ投入
- cat で結果を確認

SSH接続

手元のPC 💻

WSL (windowsの場合)

Claude4

🔑ssh-agent

SSHコマンド実行時，🔑秘密鍵とパスフレーズが自動で使用され安全

wcgw
MCPサーバ
(サブプロセス)

※wcgwを使わずにClaudeCode自ら対話型シェルにSSH等で突入するとtimeoutまでの2分間待たされる

【Linuxコマンド実行】
- ファイル読み書き等

📁ファイルシステム

ClaudeCode
└── /Project1 📁
- ├── Makefile 📄 （コンパイル用）
- ├── mat-mat-noop.bash 📄 （ジョブスクリプト）
- ├── Changed.md 📋 （変更内容と結果記録）
- ├── mat-mat-noop_v1.0.c 📝 （生成されたコード1）
- └── mat-mat-noop_v2.0.c 📝 （生成されたコード2）

実験：コードチューニングのAT

- 与えたコードの概要
  - 行列-行列積のCコード
  - 単純な3重ループで実装
  - 結果検証の機能付
    (理論解と検証ルーチン有)
- コードチューニング課題
  - 対象となる手続き(MyMatMat)中の３重ループをループアンローリングによる高速化をせよ
- VibeCodeHPCの動作
  反復プロンプトの動作
  1. 手元のPCでコード生成
  2. 生成コードをコード転送
     (「不老」TypeIサブシステム)
  3. 「不老」TypeIのログイン上でmake
  4. ジョブスケジューラへ
     pjsubコマンドでジョブを投入/実行
  5. 結果確認 (実行速度の確認)
  6. コード修正指示
  7. 1に戻る

```
for (i=0; i<n; i++) {
   for (j=0; j<n; j++) {
      for (k=0; k<n; k++) {
         C[i][j] += A[i][k] * B[k][j];
      }
   }
}
```

この一連の動作が
完全自動化!

与えたプロンプト

スパコン上のClaudeCode📁直下に/testを作成し
手元でコード生成→転送→TypeI→$ make→pjsubで実行→結果確認→修正
を繰り返して下さい。

makefileの修正はせず、ファイルは上書きせず手元にMat-Mat-noopt¥C¥mat-mat-noopt_v1_0.cのようなコピーを作成してからファイルを上書きしていくバージョン管理を推奨します（これの結果がジョブID.stdoutの場合、その対応もメモして）
つまりスパコン上のファイルは常にMakefile, mat-mat-noopt.bash, mat-mat-noopt.c の3つと、stdout等の結果のみです。
手元にChanged.mdを作成し、どんな修正を行ったかを書くのも良いかもしれません

課題：MyMatMat３重ループのアンローリングによる高速化を行ってください
行列-行列積のアンローリングを、i, j, k ループについて施し、性能向上の度合いを調べよ。
複数の行列サイズについて、どのアンローリング方式や段数が高速となるだろうか。

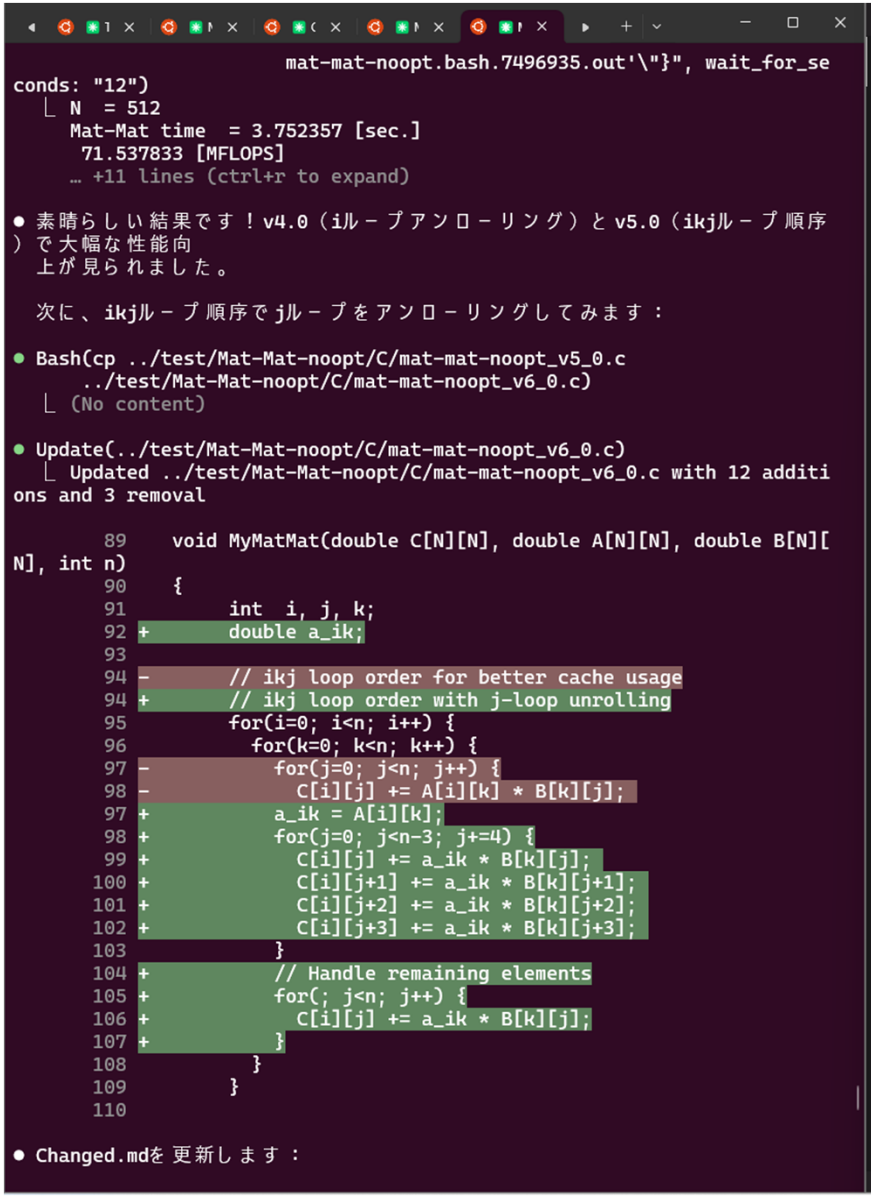
注意事項：コンパイラの最適化レベルを０にしてあります。 本演習では、最適化レベルをとりあえず０で固定しておいてください。
bashやmakefileは修正してはなりません

$ pjsub mat-mat-noopt.bashでジョブを投げられます pjstat2で確認 pjdelで削除も可能です
実行が終了したら、 $ cat mat-mat-noopt.bash.XXXX.out (XXXXは数字)で確認。

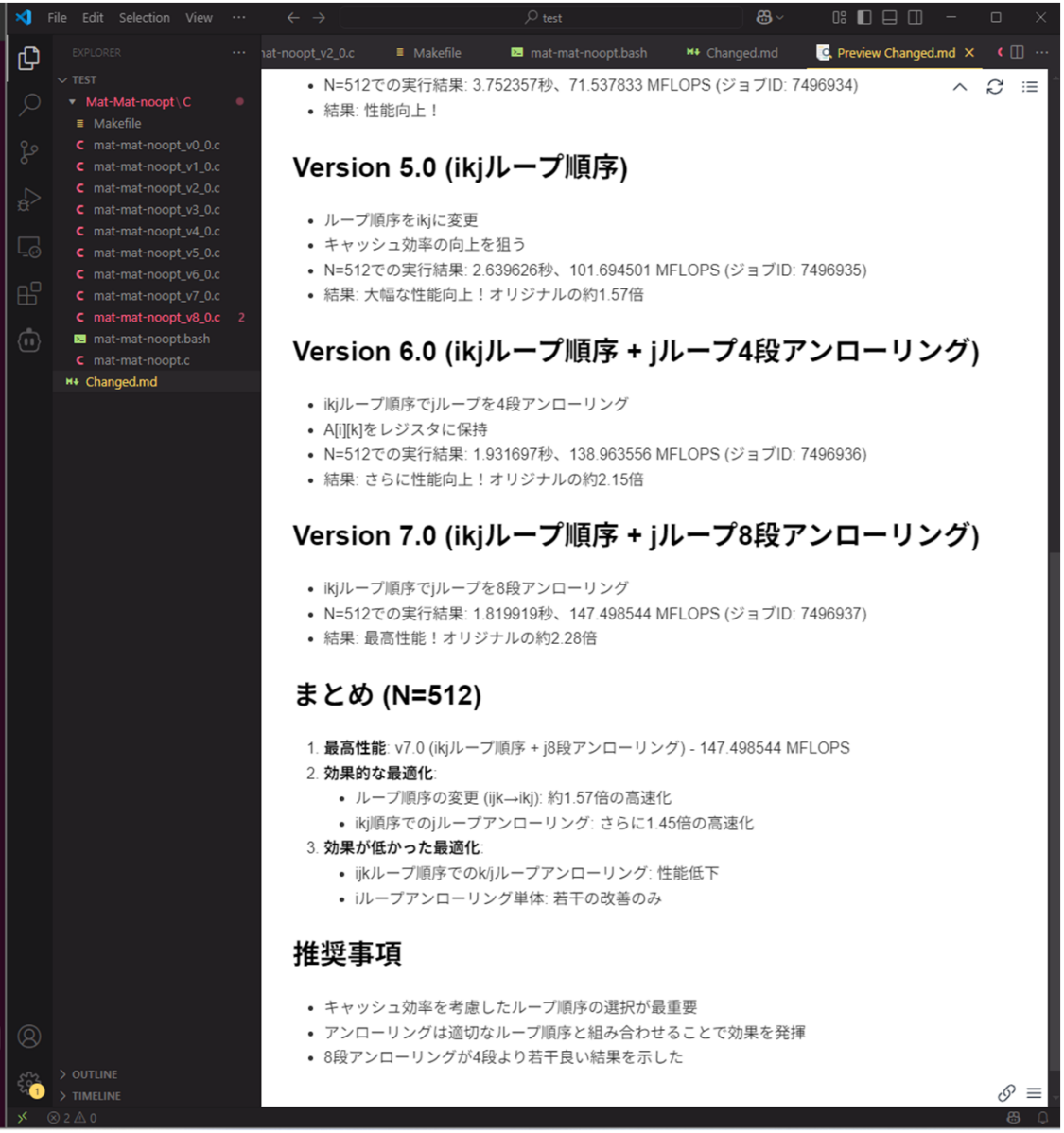
以下のような結果が見えれば成功 (C言語の場合) N = 512 Mat-Mat time = 4.215728 [sec.] 63.674759 [MFLOPS] OK!

サンプルプログラムの説明（C言語） #define N 512 の、数字を変更すると、行列サイズが変更できます
#define DEBUG 1 「１」にすると、行列 -行列積の演算結果が 検証できます。 MyMatMat関数の仕様
Double型 N × N行列ＡとＢの行列積をおこない、 Double型Ｎ×Ｎ行列Ｃにその結果が入ります

最適化終了時のレポート(自動生成)



```
                    mat-mat-noopt.bash.7496935.out'\"}", wait_for_se
conds: "12")
  └ N  = 512
    Mat-Mat time  = 3.752357 [sec.]
     71.537833 [MFLOPS]
    … +11 lines (ctrl+r to expand)

● 素晴らしい結果です！v4.0（iループアンローリング）とv5.0（ikjループ順序
）で大幅な性能向
  上が見られました。

  次に、ikjループ順序でjループをアンローリングしてみます：

● Bash(cp ../test/Mat-Mat-noopt/C/mat-mat-noopt_v5_0.c
      ../test/Mat-Mat-noopt/C/mat-mat-noopt_v6_0.c)
  └ (No content)

● Update(../test/Mat-Mat-noopt/C/mat-mat-noopt_v6_0.c)
  └ Updated ../test/Mat-Mat-noopt/C/mat-mat-noopt_v6_0.c with 12 additi
ons and 3 removal

       89    void MyMatMat(double C[N][N], double A[N][N], double B[N][
N], int n)
       90    {
       91        int  i, j, k;
       92 +      double a_ik;
       93
       94 -      // ikj loop order for better cache usage
       94 +      // ikj loop order with j-loop unrolling
       95        for(i=0; i<n; i++) {
       96          for(k=0; k<n; k++) {
       97 -          for(j=0; j<n; j++) {
       98 -            C[i][j] += A[i][k] * B[k][j];
       97 +          a_ik = A[i][k];
       98 +          for(j=0; j<n-3; j+=4) {
       99 +            C[i][j] += a_ik * B[k][j];
      100 +            C[i][j+1] += a_ik * B[k][j+1];
      101 +            C[i][j+2] += a_ik * B[k][j+2];
      102 +            C[i][j+3] += a_ik * B[k][j+3];
      103           }
      104 +          // Handle remaining elements
      105 +          for(; j<n; j++) {
      106 +            C[i][j] += a_ik * B[k][j];
      107 +          }
      108         }
      109       }
      110

● Changed.mdを更新します：
```



- N=512での実行結果: 3.752357秒、71.537833 MFLOPS (ジョブID: 7496934)
- 結果: 性能向上！

Version 5.0 (ikjループ順序)

- ループ順序をikjに変更
- キャッシュ効率の向上を狙う
- N=512での実行結果: 2.639626秒、101.694501 MFLOPS (ジョブID: 7496935)
- 結果: 大幅な性能向上！オリジナルの約1.57倍

Version 6.0 (ikjループ順序 + jループ4段アンローリング)

- ikjループ順序でjループを4段アンローリング
- A[i][k]をレジスタに保持
- N=512での実行結果: 1.931697秒、138.963556 MFLOPS (ジョブID: 7496936)
- 結果: さらに性能向上！オリジナルの約2.15倍

Version 7.0 (ikjループ順序 + jループ8段アンローリング)

- ikjループ順序でjループを8段アンローリング
- N=512での実行結果: 1.819919秒、147.498544 MFLOPS (ジョブID: 7496937)
- 結果: 最高性能！オリジナルの約2.28倍

まとめ (N=512)

1. **最高性能**: v7.0 (ikjループ順序 + j8段アンローリング) - 147.498544 MFLOPS
2. **効果的な最適化**:
   - ループ順序の変更 (ijk→ikj): 約1.57倍の高速化
   - ikj順序でのjループアンローリング: さらに1.45倍の高速化
3. **効果が低かった最適化**:
   - ijkループ順序でのk/jループアンローリング: 性能低下
   - iループアンローリング単体: 若干の改善のみ

推奨事項

- キャッシュ効率を考慮したループ順序の選択が最重要
- アンローリングは適切なループ順序と組み合わせることで効果を発揮
- 8段アンローリングが4段より若干良い結果を示した

今後の課題：
マルチAIエージェントでの評価

©名古屋大学情報学研究科M1 林君

- 端末多重化ツール(terminal multiplexer)のCLIであるtmuxで実装可能 ※その他のCLIでも実現可能

tmuxで分割した高速な並列エージェント開発

HPCワークフローでの役割分担

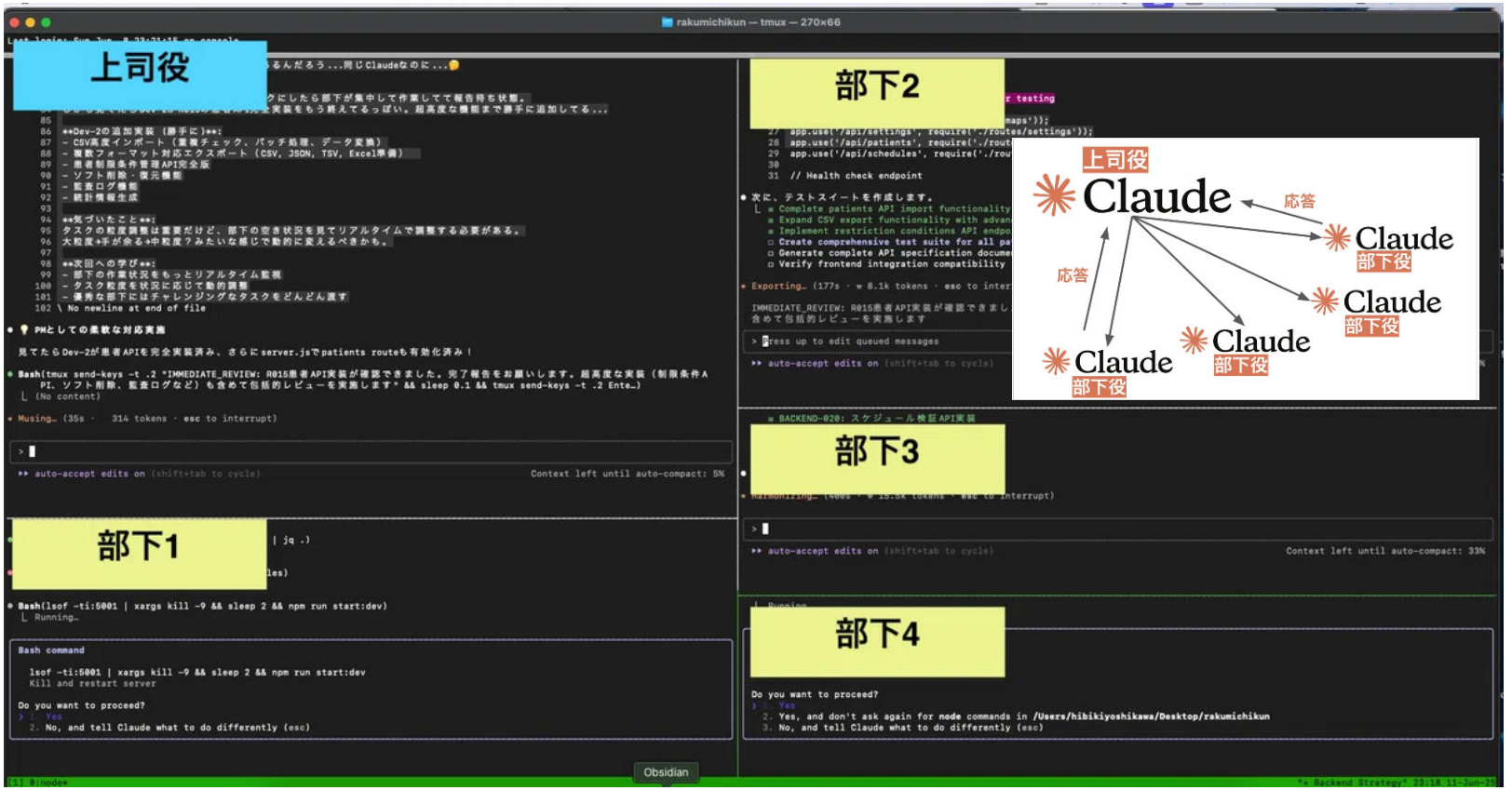
① テスト実行係

② ファイル転送係

↓認証不要

③ ログ分析係

④ コード生成係



プロジェクト詳細

- 詳しくは第三部 「不老」ユーザからの講演
  - 16:00-16:25
    「Fortranアプリケーションを対象とした生成AIによるGPUコード開発の初期評価」
    / 星野哲也（名古屋大学 情報基盤センター 准教授）
  - 16:25-16:50
    「汎用LLMによるBLASコード自動生成能力の考察」
    / 椋木大地（名古屋大学 情報基盤センター 助教）